Ilia Bider   Terry Halpin
John Krogstie   Selmin Nurcan
Erik Proper   Rainer Schmidt
Pnina Soffer   Stanisław Wrycza (Eds.)

LNBIP 113

Enterprise, Business-Process and Information Systems Modeling

13th International Conference, BPMDS 2012
17th International Conference, EMMSAD 2012, and 5th EuroSymposium
held at CAiSE 2012, Gdańsk, Poland, June 2012, Proceedings

Springer

Lecture Notes
in Business Information Processing 113

Series Editors

Ilia Bider Terry Halpin
John Krogstie Selmin Nurcan
Erik Proper Rainer Schmidt
Pnina Soffer Stanisław Wrycza (Eds.)

Enterprise, Business-Process and Information Systems Modeling

13th International Conference, BPMDS 2012
17th International Conference, EMMSAD 2012
and 5th EuroSymposium
held at CAiSE 2012, Gdańsk, Poland, June 25-26, 2012
Proceedings

Springer

Volume Editors

Ilia Bider
IbisSoft AB, Sweden
E-mail: ilia@ibissoft.se

Terry Halpin
LogicBlox, Australia / INTI International University, Malaysia
E-mail: terry.halpin@logicblox.com

John Krogstie
Norwegian University of Science and Technology, Norway
E-mail: krogstie@idi.ntnu.no

Selmin Nurcan
University Paris 1 Pantheon-Sorbonne, France
E-mail: nurcan@univ-paris1.fr

Erik Proper
CRP Henri Tudor, Luxembourg / Radboud University Nijmegen, The Netherlands
E-mail: erik.proper@tudor.lu

Rainer Schmidt
Aalen University, Germany
E-mail: Rainer.Schmidt@htw-aalen.de

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

Stanisław Wrycza
University of Gdańsk, Poland
E-mail: swrycza@ug.edu.pl

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-31071-3 e-ISBN 978-3-642-31072-0
DOI 10.1007/978-3-642-31072-0
Springer Heidelberg Dordrecht London New York
Library of Congress Control Number: 2012939337
ACM Computing Classification (1998): J.1, H.4.1, H.3.5, D.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

This book contains the proceedings of two long-running events held along with the CAiSE conferences relating to the areas of enterprise, business-process and information systems modeling: the 13th International Conference on Business Process Modeling, Development and Support (BPMDS 2012), the 17th International Conference on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2012). EMMSAD 2012 was arranged jointly with the AIS SIGSAND Europe EuroSymposium 2012. The two working conferences are introduced below.

BPMDS 2012

BPMDS has been held as a series of workshops devoted to business process modeling, development and support since 1998. During this period, business process analysis and design has been recognized as a central issue in the area of information systems (IS) engineering. The continued interest in these topics on behalf of the IS community is reflected by the success of the last BPMDS events and the recent emergence of new conferences and workshops devoted to the theme. In 2011 BPMDS became a two-day working conference attached to CAiSE (Conference on Advanced Information Systems Engineering). The basic principles of the BPMDS series are:

1. BPMDS serves as a meeting place for researchers and practitioners in the areas of business development and business applications (software) development.
2. The aim of the event is mainly discussions, rather than presentations.
3. Each event has a theme that is mandatory for idea papers.
4. Each event's results are, usually, published in a special issue of an international journal.

The goals, format, and history of BPMDS can be found on the website: www.bpmds.org

The intention of BPMDS is to solicit papers related to business process modeling, development and support (BPMDS) in general, using quality as a main selection criterion. As a working conference, we aim to attract papers describing mature research, but we still give place to industrial reports and visionary idea papers. To encourage new and emerging challenges and research directions in the area of business process modeling, development and support, we have a unique focus theme every year. Papers submitted as idea papers are required to be of relevance to the focus theme, thus providing a mass of new ideas around a relatively narrow but emerging research area. Full research papers and experience reports do not necessarily need to be directly connected to this theme (they still needed to be explicitly relevant to BPMDS though). The focus theme for BPMDS 2012 idea papers was "Business Processes and Business Process Management in the Cloud." Many enterprises use cloud computing for implementing their business processes. They intend to reduce cost, improve their agility or concentrate on their core competencies. Important areas of research are the mapping of business processes to cloud services and the use of cloud-enabled capabilities for business process design and redesign. Cloud computing also provides new means for collaboration in business process management. Stakeholders can thus be integrated more intensively and more frequently. Therefore, one important research question is, how and to which extent might the cloud influence the design, the operation and the management of business process lifecycles. BPMDS 2012 received a number of 48 submissions from 26 countries (Australia, Austria, Brazil, Cameroon, Colombia, Estonia, France, Germany, Greece, India, Israel, Italy, Latvia, Lebanon, The Netherlands, Norway, Poland, Portugal, Spain, Sweden, Switzerland, Tunisia, Turkey, Uganda, USA). The management of paper submission and reviews was supported by the EasyChair conference system. Selecting the papers to be accepted was a worthwhile effort. Each paper received at least three reviews. Eventually, 17 high-quality papers were selected; among them one experience reports and two idea papers. The accepted papers cover a wide spectrum of issues related to business process development, modeling, and support.

They are organized under the following section headings:

- Business Process in the Cloud;
- Advanced BPM in an Organizational Context;
- Similarity, Variations, Configuration;
- BPM and Requirements Engineering;
- Humans and Business Process Models;
- BPM Technologies Using Computational Methods.

We wish to thank all the people who submitted papers to BPMDS 2012 for having shared their work with us, as well as the members of the BPMDS 2012 Program Committee, who made a remarkable effort reviewing the large number of submissions. We also thank the organizers of CAiSE 2012 for their help with the organization of the event, and IFIP WG8.1 for the support.

April 2012

Ilia Bider
Selmin Nurcan
Rainer Schmidt
Pnina Soffer

EMMSAD 2012/EuroSymposium 2012

The field of information systems analysis and design includes numerous information modeling methods and notations (e.g., ER, ORM, UML, Archimate, EPC, DEMO, DFDs, BPMN) that are typically evolving. Even with some attempts toward standardization (e.g., UML for object-oriented design), new modeling methods are constantly being introduced, many of which differ only marginally from existing approaches. These ongoing changes significantly impact the way information systems are being analyzed and designed in practice. EMMSAD focuses on exploring, evaluating, and enhancing current information modeling methods and methodologies. Though the need for such studies is well recognized, there is a paucity of such research in the literature. The objective of the conference is to provide a forum for researchers and practitioners interested in modeling methods in systems analysis and design to meet and exchange research ideas and results. It also gives the participants an opportunity to present their research papers and experience reports, and to take part in open discussions. EMMSAD 2012 was the 17th in a series of events, previously held in Heraklion, Barcelona, Pisa, Heidelberg, Stockholm, Interlaken, Toronto, Velden, Riga, Porto, Luxembourg, Trondheim, Montpellier, Amsterdam, Hammamet and London. This year we had 28 papers submitted with authors from 19 countries and five continents (Australia, Belgium, Brazil, Canada, Germany, Israel, Italy, Luxembourg, The Netherlands, New Zealand, Norway, Poland, Portugal, Russia, Saudi Arabia, Spain, Sweden, UK, USA). After an extensive review process by a distinguished international Program Committee, with each paper receiving at least three reviews, we accepted the 13 papers that appear in these proceedings. Congratulations to the successful authors! Apart from the contribution by paper authors, the quality of EMMSAD 2012 depended in no small way on the generous contribution of time and effort by the Program Committee and the additional reviewers. Their work is greatly appreciated. We also express our sincere thanks to the CAiSE Organizing Committee. Continuing with our very successful collaboration with IFIP WG 8.1 (https://research.idi.ntnu.no/ifip-wg81/) that started in 1997, this year's event was again a joint activity between CAiSE and WG 8.1 and this year also with AIS-SIGSAND (http://sigsand.aisnet.org/). Another co-organizer this year was the Polish chapter of AIS, European Research Centre for Information Systems, Enterprise Architecture Network, the ORM Foundation, and University of Gdansk, Poland.

For more information on the EMMSAD-series, see our website: www.emmsad.org

The Association for Information Systems (AIS) is undertaking an initiative toward Systems Analysis and Design (SAND) international development, including the EuroSymposium on Systems Analysis and Design. Its objective is to promote and develop high-quality research on all issues related to SAND. It provides a forum for SAND researchers and practitioners in Europe and beyond to interact, collaborate and develop their field. The EuroSymposia were initiated by Keng Siau as the SIGSAND – Europe Initiative. Previous EuroSymposia were held at:

- University of Galway, Ireland – 2006
- University of Gdansk, Poland – 2007
- University of Marburg, Germany – 2008
- University of Gdansk – 2011

The accepted submissions of EuroSymposium 2007 were published in: A.Bajaj, S.Wrycza (eds), Systems Analysis and Design for Advanced Modeling Methods: Best Practices Information Science Reference, IGI Global, Hershey, New York, 2009. After a three-year break, one of the former organizers, the Department of Business Informatics of the University of Gdansk, decided to re-start the EuroSymposium as a joint undertaking of two AIS units – SIGSAND and PLAIS. Therefore, the three organizers of the 4th EuroSymposium on Systems Analysis and Design were:

- SIGSAND – AIS Special Interest Group on Systems Analysis and Design
- PLAIS – The Polish Chapter of AIS
- The Department of Business Informatics of University of Gdansk, Poland

SIGSAND is one of the most active AIS SIGs with a substantial record of contributions to AIS. It provides services such as the annual North American and European SAND Symposia, research and teaching tracks at major IS conferences, a listserv and special issues in journals.

The Polish Chapter of the Association for Information Systems (PLAIS) was established in 2006 as the joint initiative of Claudia Loebbecke, former President of AIS, and Stanislaw Wrycza, University of Gdansk, Poland. PLAIS co-organizes international and domestic IS conferences.

The Department of Business Informatics of the University of Gdansk conducts intensive teaching and research activities. Some of its academic manuals are bestsellers in Poland. The department is also active internationally, organizing various conferences including the 10th European Conference on Information Systems (ECIS 2002) and the 7th International Conference on Perspectives in Business Informatics Research (BIR 2008). The department is a partner of the European Research Center for Information systems consortium.

April 2012

John Krogstie
Stanislaw Wrycza
Erik Proper
Terry Halpin

Organization

BPMDS 2012 Organizing Committee

Selmin Nurcan	University Paris 1 Panthéon Sorbonne, France
Pnina Soffer	University of Haifa, Israel
Rainer Schmidt	Aalen University, Germany
Ilia Bider	IbisSoft, Stockholm, Sweden

BPMDS 2012 Industrial Advisory Board

Ilia Bider	IbisSoft, Stockholm, Sweden
Gil Regev	EPFL and Itecor, Switzerland
Lars Taxén	Linköping University, Sweden

BPMDS 2012 Program Committee

Sebastian Adam	Fraunhofer IESE, Germany
Antonia Albani	Delft University of Technology, The Netherlands
Judith Barrios Albornoz	Universidad de Los Andes, Venezuela
Eric Andonoff	Université Toulouse 1, France
Ilia Bider	IbisSoft, Stockholm, Sweden
Pere Botella	UPC Barcelona Tech, Spain
Johny Ghattas	University of Haifa, Israel
Claude Godart	Nancy-Université, France
Giancarlo Guizzardi	Federal University of Espírito Santo, Brazil
Marte Indulska	University of Queensland, Australia
Paul Johannesson	Royal University of Technology, Sweden
Marite Kirikova	Riga Technical University, Latvia
Christian Koot	Aalen University, Germany
Agnes Koschmider	Karlsruhe Institute of Technology, Germany
Renata Mendes de Araujo	Federal University of the State of Rio de Janeiro, Brazil
Jan Mendling	Vienna University of Economics and Business, Austria
Jens Nimis	Karlsruhe University of Applied Sciences, Germany
Selmin Nurcan	University Paris 1 Panthéon Sorbonne, France
Andreas Oberweis	Karlsruhe Institute of Technology, Germany
Oscar Pastor	Universitat Polytechnica de Valencia, Spain
Louis-Francois Pau	Erasmus University, The Netherlands

Jan Recker	Queensland University of Technology, Australia
Gil Regev	EPFL and Itecor, Switzerland
Manfred Reichert	University of Ulm, Germany
Hajo Reijers	Eindhoven University of Technology, The Netherlands
Iris Reinhartz-Berger	University of Haifa, Israel
Stephanie Rinderle-Ma	University of Vienna, Austria
Michael Rosemann	Queensland University of Technology, Australia
Shazia Sadiq	University of Queensland, Australia
Rainer Schmidt	Aalen University, Germany
Pnina Soffer	University of Haifa, Israel
Lars Taxén	Linköping University, Sweden
Roland Ukor	FirstLinq Limited, UK
Barbara Weber	University of Innsbruck, Austria
Petia Wohed	Stockholm University, Sweden
Jelena Zdravkovic	Royal University of Technology, Sweden
Michael zur Muehlen	Stevens Institute of Technology, USA

Additional Reviewers for BPMDS 2012

Eike Bernhard	Queensland University of Technology, Australia
Anne Groß	Fraunhofer IESE, Germany
Jens Kolb	University of Ulm, Germany
Marcello La Rosa	Queensland University of Technology, Australia
Maria Leitner	University of Vienna, Austria
Jürgen Mangler	University of Vienna, Austria
Erik Perjons	University of Stockholm, Sweden
Norman Riegel	Fraunhofer IESE, Germany
Niz Safrudin	Queensland University of Technology, Australia
Özgür Ünalan	Fraunhofer IESE, Germany

EMMSAD 2012/EuroSymposium 2012 Organizing Committee

John Krogstie	Norwegian Institute of Science and Technology, Norway
Stanislaw Wrycza	University of Gdansk, Poland
Erik Proper	Public Research Centre Henri Tudor, Luxembourg, and Radboud University Nijmegen, The Netherlands
Terry Halpin	LogicBlox, Australia, and INTI International University, Malaysia

EMMSAD Steering Committee

Terry Halpin	LogicBlox, Australia, and INTI International University, Malaysia
John Krogstie	Norwegian Institute of Science and Technology, Norway
Keng Siau	University of Nebraska-Lincoln, USA

EMMSAD 2012/EuroSymposium 2012 Program Committee

Stephan Aier	University of St. Gallen, Switzerland
Antonia Albani	Delft University of Technology, The Netherlands
Raian Ali	Lero, University of Limerick, Ireland
Eduard Babkin	Higher School of Economics, Moscow, Russia
Herman Balsters	University of Groningen, The Netherlands
Annie Becker	Florida Institute of Technology, USA
Giuseppe Berio	University of Turin, Italy
Nacer Boudjlida	Loria, France
Andy Carver	INTI International University, Malaysia
Olga De Troyer	Vrije Universiteit Brussel, Belgium
John Erickson	University of Nebraska-Omaha, USA
Peter Fettke	Institute for Information Systems (IWi) at the DFKI, Germany
Remigijus Gustas	Karlstad University, Sweden
Wolfgang Hesse	University of Marburg, Germany
Stijn Hoppenbrouwers	Radboud University Nijmegen, The Netherlands
Jon Iden	Norges Handelshøyskole, Bergen, Norway
Marite Kirikova	Riga Technical University, Latvia
Bogdan Lent	University of Applied Sciences, Zurich, Switzerland
Pericles Loucopoulos	Loughborough University, UK
Kalle Lyytinen	Case Western Reserve University, USA
Leszek Maciaszek	Macquarie University Sydney, Australia
Florian Matthes	Technical University of Munich, Germany
Raimundas Matulevičius	University of Tartu, Estonia
Graham McLeod	University of Cape Town, South Africa
Jan Mendling	Vienna University of Economics and Business, Austria
Wolfgang Molnar	Public Research Centre Tudor, Luxembourg
Tony Morgan	INTI International University, Malaysia
Haralambos Mouratidis	University of East London, UK
Bjorn Erik Munkvold	University of Agder, Norway

Andreas L. Opdahl	University of Bergen, Norway
Sietse Overbeek	Delft University of Technology, The Netherlands
Hervé Panetto	University Henri Poincaré Nancy I, France
Barbara Pernici	Politecnico di Milano, Italy
Anne Persson	University of Skövde, Sweden
Michaël Petit	University of Namur, Belgium
Nava Pliskin	Ben-Gurion University of the Negev, Israel
Paul Ralph	Lancaster University, UK
Jolita Ralyté	University of Geneva, Switzerland
Sudha Ram	University of Arizona, USA
Jan Recker	Queensland University of Technology, Australia
Colette Rolland	University of Paris 1, France
Michael Rosemann	Queensland University of Technology, Australia
Matti Rossi	Helsinki School of Economics, Finland
Kurt Sandkuhl	Jönköping University, Sweden
Peretz Shoval	Ben-Gurion University of the Negev, Israel
Piotr Soja	Cracow Economic University, Poland
Guttorm Sindre	University of Trondheim, Norway
Il Yeol Song	Drexel University, USA
Janis Stirna	Royal Institute of Technology, Sweden
Inge van de Weerd	Utrecht University, The Netherlands
Wil van der Aalst	Eindhoven University, The Netherlands
Dirk van der Linden	Public Research Centre Henri Tudor, Luxembourg
Johan Versendaal	University of Utrecht, The Netherlands
Carson Woo	University of British Columbia, USA
Jelena Zdravkovic	Stockholm University
Martin Zelm	CIMOSA, Germany
Iryna Zolotaryova	Kharkiv National University of Economics, Ukraine
Pär Ågerfalk	Uppsala University, Sweden

Additional Reviewers for EMMSAD 2012/EuroSymposium 2012

Lamia Abo Zaid
Ralf Abraham
Nils Labusch
Eduardo Loures
Christian Neubert
Ivan Sidorenko
Alexander Steinhoff
Jürgen Walter
Iyad Zikra

Table of Contents

Humans and Business Process Models

BPM Technologies Using Computational Methods

Business Process in the Cloud (2)

EMMSAD

Modeling of Enterprise Architecture

Modeling Facts and Rules

Business Process Modeling

Modeling of Non-functional Requirements

Safety Modeling and Analysis

Quality of Models and Modeling Languages

Supporting the Learning of Conceptual Modeling

Scalable Business Process Enactment in Cloud Environments

Rainer Schmidt

Aalen University,
Beethovenstraße 1,
73430 Aalen,
Germany
Rainer.Schmidt@htw-aalen.de

Abstract. Business processes have to be enacted with increasing numbers of instances and under a growing pressure to reduce the cost of process enactment. Cloud computing provides scalable computing resources as utility at low cost. Therefore, this paper introduces service systems as new, cloud-based architecture for the scalable and flexible enactment of business processes.

1 Introduction

Business processes have to be enacted in a scalable and flexible manner. The have to be enacted with increasing numbers of instances and under a growing pressure to reduce the cost of process enactment. Business processes have to be adapted to changed requirements, and new business processes have to be implemented in very short time. Thus, business process management solutions have to be flexible [1].

Cloud computing [2], [3] is able to provide computing services economically and flexibly. There are economies of scale in three areas [4]: supply-side saving by large scale datacenters, demand-side aggregation and multi-tenancy efficiency. Cloud computing offers cloud services with very low upfront investments. E.g. a start-up enterprise is able to minimize its expenses, by beginning with a low-volume cloud-service consumption and scale up on a pay-for-use only basis. Cloud services are scalable. It is possible to increase or decrease their usage according to market requirements without the need to create an infrastructure capable to fulfil the maximum demand.

Due to these advantages, the idea is obvious to use cloud computing for business process management. First approaches [5][6] in this direction simply map the elements of a business process management systems [7] to cloud services either as Software as a Service (SaaS), Platform as a Service (PaaS) or Infrastructure as a Service (IaaS) [2], [3]. In their core these approaches can be traced back to the workflow reference model of the workflow management coalition. [8]. A (semi-)formal model of the business process, using e.g. BPMN [9] is either directly or indirectly executed using an engine. Often, BPMN is transformed into an executable language such as BPEL [6].

However, to use cloud services is not enough to provide a scalable and flexible enactment of business processes. Many approaches for business process enactment

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 1–15, 2012.

use a centralized engine for interpreting and executing business processes. Such a centralized engine creates an architectural bottleneck that cannot be circumvented by using cloud services. All instances have to pass the central engine. In this way, the load distribution and resource sharing mechanisms in clouds are canceled, as already analyzed in in [10–12]. Also, a single point of failure is created.

The contribution of this paper is to introduce the first sketch of a scalable process enactment architecture based on services systems, sets of configured services and resources provided by a cloud environment.

The paper starts with an analysis of cloud based business process management and creates a framework for the classification of different approaches. The framework allows also to compare the properties such as scalability and adaptability. Then a description and formalization of cloud-environments is given. Using it, the new approach for scalable process enactment using service systems is described. Related work is discussed in the following section. Finally, a conclusion and outlook is given.

2 A Framework for Cloud-Based Business Process Management

The advantages of cloud computing initiated a number of approaches to move business process management systems into the cloud. In [5] a first architectural sketch for cloud-based business process management systems [7] is presented. The basic architectural elements of a business process management system are associated with the abstraction layers of cloud-computing. Different cloud delivery models for BPEL-based business process management systems are analyzed in [6]. In [13] a business process centric model for Platform-as-a-Service is introduced to support industry-specific application requirements. This approach is limited to Platform-as-a-Service.

To compare the existing approaches, a framework shall be created. It uses four architectural layers: business process management, applications, platforms and resources. The top-most layer represents business process management as defined in [7]. Below is a layer representing application functionality. These applications are based on (software) platforms. Software platforms are provided using resources described in the fourth layer.

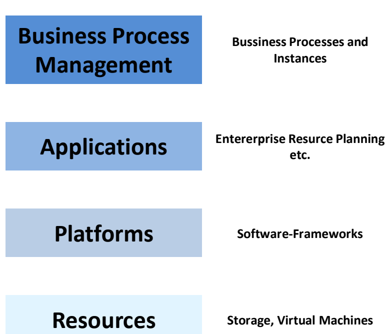


Fig. 1. Service-oriented enterprise architecture

The most thorough approach is to move business process management completely into the cloud. This approach is called Business Process Management as a service (BPMaaS): Business Process Management is provided as one or several services. There are already examples for BPMaaS. Oryx[14] a modeling platform, is offered as a cloud-based business process design tool called Signavio [15]. The Google Apps marketplace [16] offers several BPM solutions.

The usage of cloud computing for business process management is reduced in the SassS approach. Here, application functionality is provided as a service (SaaS). The BPM system, however, uses cloud services providing application functionality but is itself outside of the cloud. An example scenario is a BPM system interacting with several applications.

The next approach uses platform services (PaaS) to implement applications, which do not appears as services themselves. The application functionality is used for supporting business processes.

Also, only virtualized resources may be provided by infrastructure services (IaaS). An example is database storage as offered in Windows Azure [17]. By this means a scalable supply with database storage is created.

The four approaches differ in their properties. A BPMaaS solution provides the highest scalability, because all participating services are within the cloud. The provider of the BPM as a Service assures that enough resources are available. There also the time and financial effort can be minimized. There is nearly no time required for setting up the BPM systems and the highest scaling effects can be achieved. Furthermore, the BPMaaS realizes the idea of as separation of concerns. The BPMaaS user does not have to bother with software updates etc.

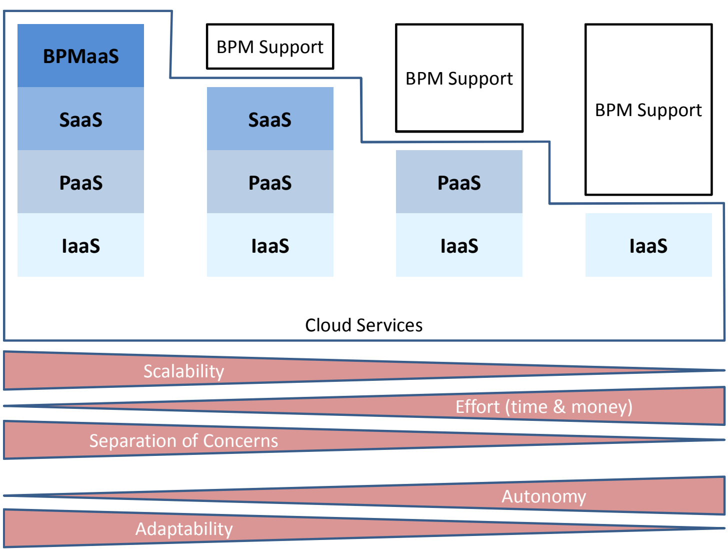


Fig. 2. Sets of cloud services used for business process management

On the other hand, using an IaaS approach provides the highest degree of autonomy. The user has complete control over all information; thus lock-in effects are avoided. Furthermore, there is a high degree of adaptability. It is possible to realize very specialized solutions.

There are two possible way to differentiate the model above, as shown in Fig. 3. First, enterprises may choose to differentiate the usage of cloud services dependent on the phase of the business process management lifecycle [7]. E.g. a company may use a BPMaaS based solution for business process design, but may use an PaaS-based solution for business process execution. Second, legacy services service, applications or data may be combined with a BPMaaS. However in this case the overall solution loses the generic advantages of cloud services, because the legacy objects become bottlenecks.

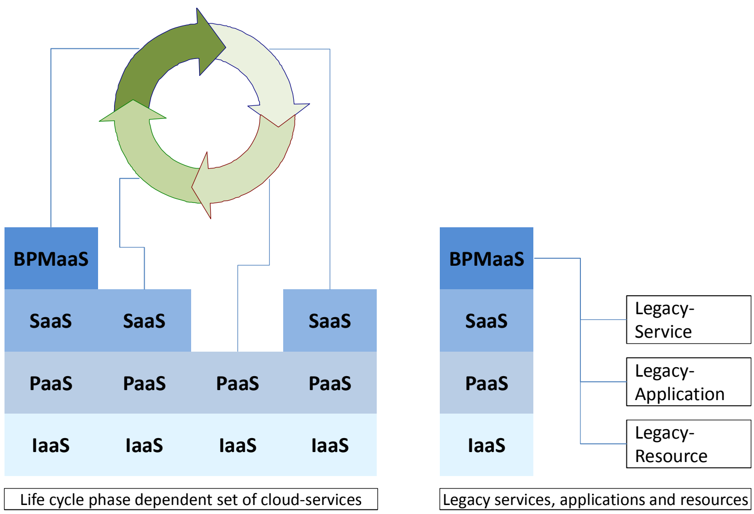


Fig. 3. Deviation from standard sets

3 Cloud-Based Process Enactment

Although BPMaaS offers the highest scalability, the scalability may be limited by architectural constraints of the business process management system. The crucial entity for the scalability of business process management systems is the process enactment[18]. Process enactment is the creation of executable instances of a process and their monitoring. Existing approaches for process enactment are often based on the architecture sketched in Fig. 4. It implements the ideas of the workflow reference model of the workflow management coalition. [8]. A (semi-) formal business process model is either directly or indirectly by a process engine to create process instances. E.g. BPMN [9] is often transformed into BPEL [7] and then executed by the process engine. The process engine also provides access to external services and resources for the process instances.

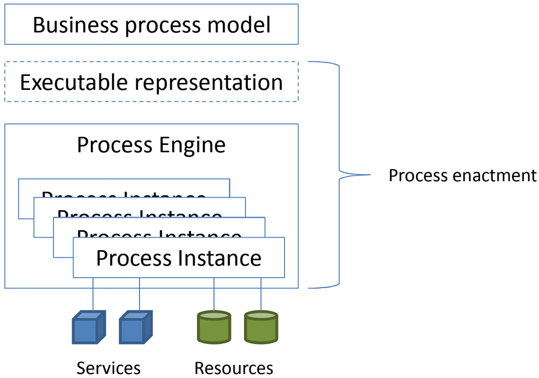


Fig. 4. Business process enactment

However, already very early research [10–12] showed, that engine approaches create performance and reliability bottlenecks. This is also true if one moves an engine approach to the cloud as shown in Fig. 5.

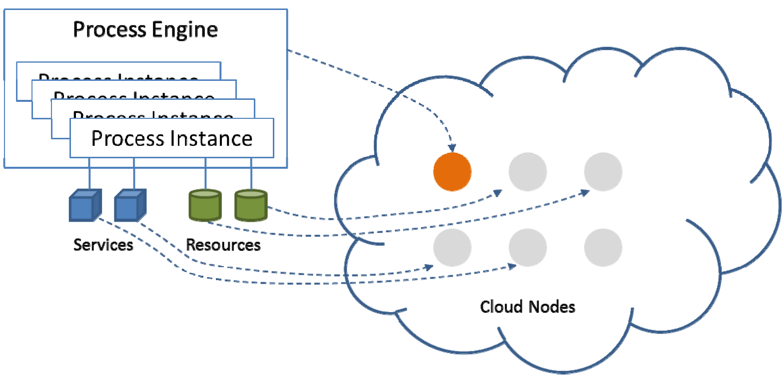


Fig. 5. Mapping of engine-based approaches to cloud nodes0

In this case, the service and resources can be mapped to different nodes of the cloud, e.g. different virtual machines running on different physical machines. However, the process engine is on one single node of the cloud. Thus, this node is a bottleneck.

To create a truly scalable business process enactment, a new approach using so-called service-systems shall be introduced. The first step is to analyze the properties of cloud environments. The formalization of cloud environments in the second step is the basis for a service system based process enactment described in the third step.

3.1 Cloud Environments

In the beginning, the provisioning of cloud services [19] had been very similar to the provisioning of services in an SOA [20]. A single cloud service was provided by a cloud service provider assuring scalability and reliability of the clouds service. Over time, however, several cloud services have been integrated to so-called cloud environments. Cloud environments are bundles of cloud services and resources. The services and resources of cloud-environments are selected to fulfill the requirements of a certain group of customers. Wide-spread examples are Office365 [21] and Google Apps [22]. They provide a bundle of office-related services such as text-processing, email, spreadsheet calculation and offer storage resources for texts, spreadsheets etc. The first cloud-environments had no means for configuring services and resources. They also lacked any possibility to extend or change the bundle of services and resources. Such static cloud-environments (SCEs) did not allow tailoring them according to individual customer requirements. Static cloud-environments quickly create a number of problems. Because services cannot be integrated into an existing cloud-environment, the user needs multiple separate cloud-environments. In this way, a "jungle" of isolated cloud-environments arises. Each of them has an own password, resource administration etc. Furthermore, it is difficult to exchange information between the static cloud-environments. Often, this has to be accomplished manually. Very quickly redundant data are created in the different cloud-environment and thus the danger of inconsistent information arises.

Classes of Cloud Environments

In reaction to the restrictions of static cloud environments, more enhanced classes of cloud-environments have been created by adding configuration and extensibility capabilities, see Table 1.

Table 1. Classification of Cloud Environments

Configurable	Configurable Cloud Environment (CCE)	Dynamic Cloud Environment (DCE)
Not configurable	Static Cloud Environment (SCE)	Extensible Cloud-Environment (ECE)
	Not extensible	**Extensible**

Configurable cloud-environments (CCEs) allow configuring services and resources. Extensible cloud-environments (ECEs) provide the possibility to extend the set of services and resources. Dynamic cloud-environments (DCEs) provide both the capabilities of CCEs and ECEs. That means, services and resources are both configurable and extensible.

Configure and Extension Operations on Cloud-Environments

The only possible interaction between a cloud service user and a static cloud environment had been the request for service itself. Configurable and extensible cloud environments introduce interactions to configure and extend the cloud environment according to user requirements. This is accomplished by a self-service approach: the user is able to do the configurations on his own. By this means he can quickly accomplish the required changes to the cloud-environments or extend it.

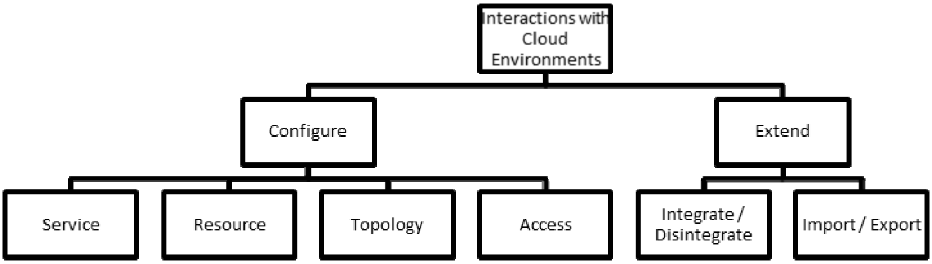


Fig. 6. Interactions with dynamic cloud environments

Configure interactions may be applied on four types of entities: services, resources, topology and access. The configuration interactions on services and resources are used to adapt them to individual requirements. Services and Resources interact within a cloud-environment. Thus, it may be necessary to adapt their topology that means the relations between services and resources. Services and resources can be associated. That means relations between different services can be established. E.g. dataflows between services can be created. The fourth interaction is the used to grant or limit access to service and resources. The user of a cloud environment may wish to grant other users access or to limit it.

The extension of cloud environments by services and resources is accomplished differently for services and resources. The extension of the cloud-environment by services is done by integrating them. The service remains outside the sphere of control of the cloud-environment. Otherwise, all resources would have had to be integrated into the cloud environment. The cloud-environment stays isolated from the details of service provisioning and manageability is facilitated. The extension of cloud-environments by resources is done by importing the resources. That means they are moved into the sphere of control [23] of the cloud-environment. By this means it is possible to manage the resource more efficiently, e.g. change the resource in the context of a transaction [24]. Thus, there is a trade-off between manageability and efficiency.

Both integrate and import interactions are associated by their inverse interactions the disintegration of services and the export of resources. The disintegration of services means that their availability with the cloud environment is stopped. The export of resources is the moving of resources out of the sphere of control of the cloud environment to a place specified by the cloud environment user.

Restricting the Use of Configure and Extensibility Operations

Theoretically, all configure and extension operations could be applicable to all services and resources and accessible for all users of the cloud environment. However, in practice, the availability of configure and extension operations is limited due to technical reasons and restricted due to administrative reasons. It may be not feasible to make configurable all types of services and resources. In the same way, it may require a too high effort to make all kinds of services available for integration or all kinds of resources importable. Therefore, the configure and extension operations may be restricted to avoid improper changes. Not only the use of certain configure and extension operations may be restricted, but also their domain and their cardinality. The operations may be restricted to certain types of services. E.g. only certain types of resources may be imported. Also the cardinality may be restricted. E.g. it is allowed to integrate only one instance of a certain service type.

3.2 Formalizing Cloud Environments

Cloud environment can be conceptualized as shown in Fig. 7 The extensibility of cloud environments embraces both services and resources. Therefore, different methods have to be applied to services and resources. The extension of cloud environments by services is done by integrating the services. As shown in Fig. 7, external services become accessible for the cloud-services by integrating them.

To add resources to the dynamic cloud environments, it is necessary to move them into the sphere of control [23] of the cloud-based information system. Otherwise, the transparent access of the cloud-services to the resource could not be guaranteed. The moving of a resource into the dynamic cloud environments is called import and means that the resources are encapsulated by an IaaS. In that way, a virtualized resource is created which is transparently accessible for the cloud-services.

The ability to disintegrate services and export resources is very important to cloud environments because many cloud-services and resources are billed by their usage. The export of resources is of particular importance to avoid a vendor lock-in [25]. Only, if there is the possibility to get back resources given to a cloud-service vendor, the change to another cloud-service vendor is feasible. Delete operations on service and resource configuration items are used to represent the disintegration of services and the export of resources from cloud environments. Disintegrate and export operations have to be preceded by detach operations on the services respectively resources.

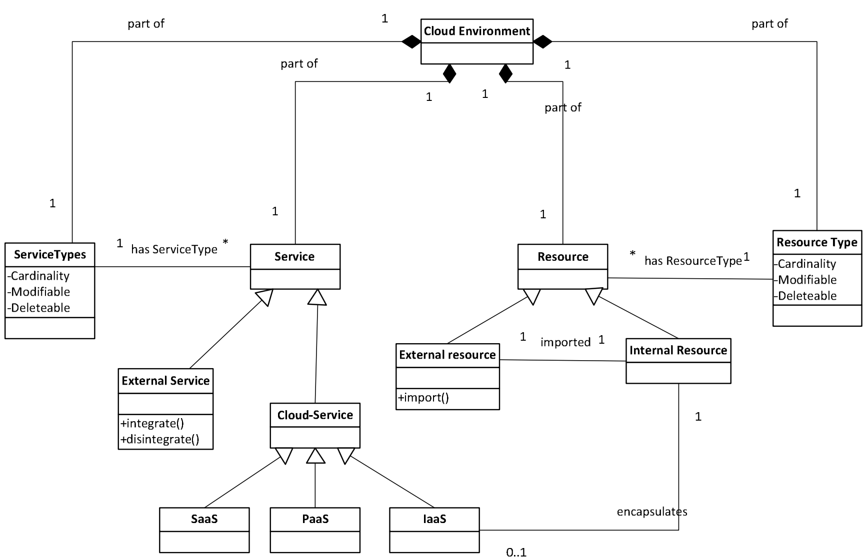


Fig. 7. Dynamic cloud-environment (DCE)

To be able restricting the use of configure and extensibility operations, it is necessary to extend the architecture of the cloud environment by service and resource types. Each service and resource type contains cardinality information, defining how many service or resource configuration items may be created. Furthermore, it contains an attribute to indicate whether the services or resources of this type may be modified or deleted.

The capabilities of a cloud environment to integrate services and import resources are differentiated into three levels. First, it is possible to integrate or import every type of service or resources respectively. This is expressed by the creation of new entities of service or configuration item types in addition to the creation of service and resource configuration items. Concrete services are assigned to the newly created configuration items. Second, only the integration of services and resources of predefined types may be possible. This is expressed by the creation of configuration items of a certain type. Third, also the cardinality may be restricted. In this case, it is only possible to assign services and resources to already existing configuration items or to change it according to the cardinality information in the of the associated service configuration item type or resource configuration item type.

The discharge of a service or resource is represented by the deletion of a configuration item and solving the assignment to this configuration item. By constraining the types of configuration items allowed to delete, it is expressed, that only certain types of services or resources may be discharged. Furthermore, it is possible to constrain the cardinality. E.g. it is possible to define, that the deletion of configuration can be done only, if at least one configuration item of a certain type remains.

4 Service System Based Process Enactment

To create a process enactment architecture which fully leverages the advantages of cloud computing, a new approach using so-called service systems is used. A service system is a configuration of services and resources, which are represented by so-called configuration items. Configuration items are differentiated into two subclasses: service and resource configuration items. The configuration capability of a service system is implemented by the modify operation. It can be restricted for defined types of service and resource configuration items by evaluating the modifiable attribute of the associated service type or resource type. The uses relationship of the service configuration item represents the usage relationships of the cloud-services. The act-upon relationship between service and resource configuration item is used to indicate which cloud-services act upon which resources. Both relationships are created and modified by the assign operation. The configuration of services and resources itself is represented by the modification of configuration items. To do so, a modify operation is defined for configuration items.

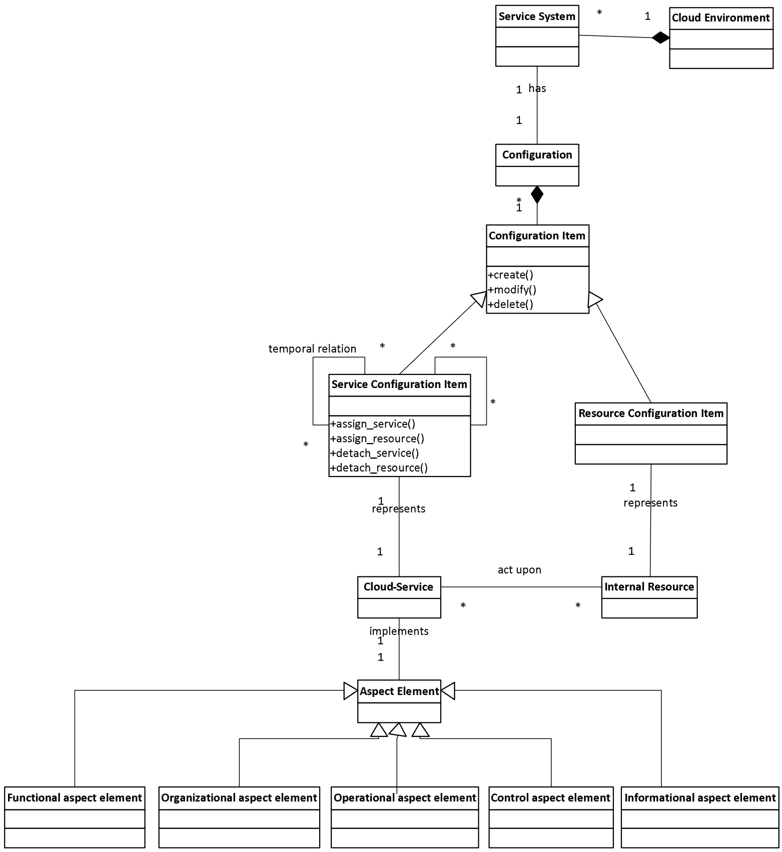


Fig. 8. Service System

To transform a business process into as services system, an aspect-oriented [26], [27] approach shall be applied. Aspects are disjoint sets of process functionality which is independently evolving. Aspects have been already described in workflow management [28] and software engineering [29]. There are five core aspects: the functional, control, informational, organizational and operational aspect [JaBu96]. The functional aspect defines the composition of a business process from sub-processes. The control aspects defines the temporal and conditional relations between activities. A mapping between the organization and activities is defined in the organizational aspect. The operational aspect defines applications or services used for executing activities. The informational aspect defines the data flows between the activities.

Business process processes are transformed into service systems by analyzing their elements and transforming the elements of the business process into aspect elements as shown in Fig. 9. Cloud-services implementing the aspect elements are mapped to the nodes of a cloud.

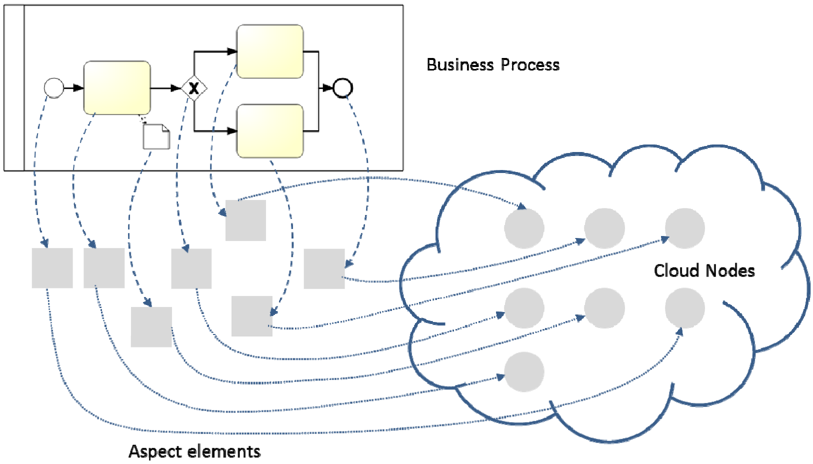


Fig. 9. Transformation of business processes to service systems

Both scalability and flexibility can be achieved without using a centralized engine approach. The cloud services implementing single aspect elements can be distributed to different nodes of the cloud. By using the configuration mechanisms provided by cloud environments, the composition of the service systems and its topology can be adapted easily.

5 Related Work

There are a number related papers addressing the same or other issues in cloud-based business process enactment. In [30] a mapping of a classical engine-based business enactment architecture to cloud-computing environments is described. The WS02 Business Process Server [31] has a multi-tenant architecture, but uses an engine-based approach. The challenges arising from multi-tenancy in business processes are discussed in [32]. The solution approach, however, only addresses SaaS cloud environments. In [33] an architecture for flexible process-based applications in hybrid clouds is developed. It is designed around a process service platform. However, it does not consider cloud services of higher abstraction levels such as software as a service. Furthermore, there are approaches addressing related, but not identical issues. Configurable process models introduced in [34] are used to implement multi-tenant processes. This work addresses modeling but not architectural issues. In [35] the NIST and DMTF cloud architectures are compared. Both architecture are very detailed but do not refer to business process enactment. In [36] an event-based distributed architecture is used to enforce SLAs in cloud-architectures supporting business processes. The integration of cloud computing with business intelligence is discussed in [37]. Cloud computing creates new possibilities for business process design by introducing content-centered collaboration spaces [38]. They enable the inter-organizational cooperation of users.

The services systems used in this paper relate to the term service system introduced by Maglio et al. [39]. They define a service system “as an open system capable of improving the state of another system through sharing or applying its resources and capable of improving its own state by acquiring external resources”. Thus, these service systems parallel with cloud-based service systems by the ability to acquire resources. In [40] the resource-event-agent enterprise ontology (REA) is used to conceptualize service systems. To do so, REA is interpreted with a Service-Dominant-Logic perspective. As a result, the Resource-Service-System model is defined. In [41] the Resource-Service-System model is used as a conceptual model of service systems. In [42] a tool for the conceptual design of service systems is developed. In [43] a concept for service systems is defined.

In industry, a number of cloud environments exist which can be used as a foundation of cloud-based information systems. Cloud-environments are bundles of cloud-services and resources. Wide-spread examples are Office365 [21] and Google Apps [22]. They provide a bundle of office-related services such as text-processing, email, spreadsheet calculation and offer storage resources for texts, spreadsheets etc. However, their ability to integrate services, import resources and to configure them highly fluctuates depending on the type of service or resource and cannot be modified due to marketing considerations.

6 Conclusion and Outlook

There is a growing need to enact business processes with an increasing number of instances and to reduce the cost of business process enactment. Both scalability and cost are issues addressed by cloud computing. The integration of services and the import of resources is facilitated by cloud environments.

To fully leverage the advantages of cloud computing, it is necessary to created new business process enactment architectures. In this paper, the so-called service system based enactment of business processes has been introduced. Service systems are a configuration of services and resources. Services can be associated with other services and resources. Using a service system based approach the enactment of business processes is distributed to services which can be easily distributed to different nodes of the cloud fabric. Thus, a high degree of scalability is achieved. Business processes are transformed to service system by identifying so-called aspect elements.

Future work will extend the concepts developed in this paper. In particular the definition of aspect elements has to be given more details. The transformation process from business processes to aspect elements and cloud services has to be specified rigorously.

Cloud computing will influence business process management in a number of different ways. First, cloud computing will change the way enterprise outsource computing utilities. Using cloud environments, it becomes much easier to procure computing services. Therefore it has to be expected, that many companies will prefer outsourcing via cloud computing instead of outsourcing complete business processes. Furthermore this reduces the danger to create new competitors by giving important knowledge to the outsourcing contractor. Second, cloud environments are capable to introduce new versions of software, platform and infrastructure services on the fly. These mechanisms can also be used to change business processes frequently. Thus, business processes can be adapted much faster to changing business requirements. Third, using a public cloud for business process support allows to minimize energy consumption and CO_2 production. First it is possible transfer the business process support to data centers that are in cold areas of the world. This means is used by more and more enterprises for reducing the power consumption necessary for cooling[44] . By using energy from renewable sources, also the CO_2 production can be reduced.

References

[1] Regev, G., Soffer, P., Schmidt, R.: Taxonomy of Flexibility in Business Processes. In: Proceedings Seventh Workshop on Business Process Modeling, Development, and Support (BPMDS 2006). Requirements for Flexibility and the Ways to Achieve it, Luxemburg, pp. S.90–S.93 (2006)

[2] Mell, P., Grance, T.: The NIST Definition of Cloud Computing, July 10 (2009), `http://csrc.nist.gov/groups/SNS/cloud-computing/` (accessed: January 06, 2011)

[3] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)

[4] Harms, R., Yamartino, M.: The economics of the cloud. Microsoft whitepaper, Microsoft Corporation (2010)

[5] Fang, Z.: BPM Architecture Design Based on Cloud Computing. Intelligent Information Management 02(05), 329–333 (2010)

[6] Anstett, T., Leymann, F., Mietzner, R., Strauch, S.: Towards bpel in the cloud: Exploiting different delivery models for the execution of business processes. In: 2009 World Conference on Services-I, pp. 670–677 (2009)

[7] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York Inc. (2007)

[8] Hollingsworth, D., et al.: Workflow management coalition: The workflow reference model. Document Number TC00-1003, no. 1.1 (1995)

[9] Silver, B.: BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0. Cody-Cassidy Press (2009)

[10] Alonso, G., Agrawal, D., El Abbadi, A., Mohan, C.: Functionality and Limitations of Current Workflow Management Systems. IEEE Expert. 12(5), 105–111 (1997)

[11] Wodtke, D., Weißenfels, J., Weikum, G., Dittrich, A.K.: The Mentor project: Steps towards enterprise-wide workflow management. In: Proceedings of the Twelfth International Conference on Data Engineering, pp. 556–565 (1996)

[12] Casati, F., Grefen, P., Pernici, B., Pozzi, G., Sánchez, G.: WIDE workflow model and architecture. Technical Report 96-19, University of Twente (1996)

[13] Zhou, Y.C., Liu, X.P., Wang, X.N., Xue, L., Liang, X.X., Liang, S.: Business Process Centric Platform-as-a-Service Model and Technologies for Cloud Enabled Industry Solutions. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 534–537 (2010)

[14] Decker, G., Overdick, H., Weske, M.: Oryx – An Open Modeling Platform for the BPM Community. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 382–385. Springer, Heidelberg (2008)
[15] BPMN 2.0 - Process Modeling with Signavio - Signavio | Simply Professional, http://www.signavio.com/en/bpmn (accessed: March 01, 2012)
[16] Google Apps Marketplace, https://www.google.com/enterprise/marketplace/ (accessed: May 30, 2011)
[17] SQL Azure - Features - Windows Azure, http://www.windowsazure.com/en-us/home/features/sql-azure/ (accessed: March 01, 2012)
[18] Kouvas, G., Grefen, P., Juan, A.: Business Process Enactment. In: Dynamic Business Process Formation for Instant Virtual Enterprises, pp. 113–132 (2010)
[19] Motahari-Nezhad, H.R., Stephenson, B., Singhal, S.: Outsourcing Business to Cloud Computing Services: Opportunities and Challenges
[20] Papazoglou, M.P., Heuvel, W.-J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 16(3), 389–415 (2007)
[21] Office 365 for Small Business - Hosted Productivity Software, http://www.microsoft.com/en-us/office365/online-software.aspx (accessed: May 30, 2011)
[22] Welcome to Google Apps, http://www.google.com/apps/ (accessed: May 30, 2011)
[23] Google Sites - Free websites and wikis, https://www.google.com/accounts/ServiceLogin?continue=http%3A%2F%2Fsites.google.com%2F&followup=http%3A%2F%2Fsites.google.com%2F&service=jotspot&passive=true&ul=1 (accessed: August 28, 2011)
[24] Bernstein, P.A., Newcomer, E.: Principles of transaction processing. Morgan Kaufmann (2009)
[25] Farrell, J., Klemperer, P.: Coordination and lock-in: Competition with switching costs and network effects. Handbook of Industrial Organization 3, 1967–2072 (2007)
[26] Schmidt, R., Assmann, U.: CompFlow: A Component-based and Aspect-separated Architecture for Workflow Support. In: Proceedings EDBT 1998 Workshop on Workflow Management Systems, Valencia (1998)
[27] Schmidt, R., Assmann, U.: Extending Aspect-Oriented Programming in Order to Flexibly Support Workflows. In: Proceedings of the ICSE 1998 AOP Workshop, pp. 41–46 (1998)
[28] Jablonski, S., Bussler, C.: Workflow management: modeling, concepts, architecture and implementation. Cengage Learning (1996)
[29] Lopes, C.V., Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J.: Aspect-oriented Programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
[30] Jiang, J., Le, J., Wang, Y., Sun, J., He, F.: The BPM Architecture Based on Cloud Computing. In: 2011 Fourth International Symposium on Knowledge Acquisition and Modeling, pp. 196–198 (2011)
[31] Pathirage, M., Perera, S., Kumara, I., Weerawarana, S.: A Multi-tenant Architecture for Business Process Executions. In: 2011 IEEE International Conference on Web Services (ICWS), pp. 121–128 (2011)
[32] Kapuruge, M., Colman, A., Han, J.: Achieving Multi-tenanted Business Processes in SaaS Applications. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) WISE 2011. LNCS, vol. 6997, pp. 143–157. Springer, Heidelberg (2011)

[33] Fehling, C., Leymann, F., Schumm, D., Konrad, R., Mietzner, R., Pauly, M.: Flexible Process-Based Applications in Hybrid Clouds. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 81–88 (2011)
[34] van der Aalst, W.M.: Business Process Configuration in the Cloud: How to Support and Analyze Multi-tenant Processes? In: 2011 Ninth IEEE European Conference on Web Services (ECOWS), pp. 3–10 (2011)
[35] (Gus) Samba, A.: Logical Data Models for Cloud Computing Architectures, http://www.computer.org/csdl/mags/it/2012/01/mit2012010019.html#Ref1 (accessed: March 03, 2012)
[36] Muthusamy, V., Jacobsen, H.-A.: BPM in Cloud Architectures: Business Process Management with SLAs and Events. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 5–10. Springer, Heidelberg (2010)
[37] Mircea, M., Ghilic-Micu, B., Stoica, M.: Combining Business Intelligence with Cloud Computing to delivery agility in actual economy. Journal of Economic Computation and Economic Cybernetics Studies 45(1) (2011)
[38] Erickson, J.S., Spence, S., Rhodes, M., Banks, D., Rutherford, J., Simpson, E., Belrose, G., Perry, R.: Content-Centered Collaboration Spaces in the Cloud. IEEE Internet Computing 13(5), 34–42 (2009)
[39] Maglio, P., Vargo, S., Caswell, N., Spohrer, J.: The service system is the basic abstraction of service science. Information Systems and E-Business Management 7(4), 395–406 (2009)
[40] Poels, G.: The resource-service-system model for service science. In: Advances in Conceptual Modeling–Applications and Challenges, pp. 117–126 (2010)
[41] Poels, G.: A Conceptual Model of Service Exchange in Service-Dominant Logic. In: Morin, J.-H., Ralyté, J., Snene, M. (eds.) IESS 2010. LNBIP, vol. 53, pp. 224–238. Springer, Heidelberg (2010)
[42] Karni, R., Kaner, M.: An engineering tool for the conceptual design of service systems. In: Advances in Services Innovations, pp. 65–83
[43] Kaner, M., Karni, R.: A Knowledge Framework for a Service Concept. In: 6th International Conference on Knowledge Management, Graz, Austria, pp. 231–235 (2006)
[44] Nordics Launch A United Cold Front On Data Centres | TechWeekEurope UK, http://www.techweekeurope.co.uk/comment/nordics-launch-a-united-cold-front-on-data-centres-62193 (accessed: March 02, 2012)

Preparing for the Era of Cloud Computing: Towards a Framework for Selecting Business Process Support Services

Ilia Bider[1,2] and Erik Perjons[1]

[1] DSV, Stockholm University, Stockholm, Forum 100, SE-16440 Kista, Sweden
[2] IbisSoft AB, Stockholm, Box 19567, SE-10432 Stockholm, Sweden
{ilia,perjons}@dsv.su.se

Abstract. The shift to the cloud computing creates new opportunities for the IT usage in business. New standard and customizable services that do not require high initial investment allow business people to choose services to support their business activities without involving technicians. Business process solutions providers are already moving their products to the cloud offering them as services. The question arises of how a business person, e.g. a department manager, can decide on which service suits best his/her needs. The paper investigates this issue in respect to the services that provide fully customizable operational support to business processes. The paper suggests a practical framework for defining requirements based on characteristics of the process to be supported by the service. The framework determines the needs of such capabilities as process flow support, shared spaces, team collaboration, etc., based on the high-level analysis of a process in question. The framework is aimed at serving as a basis for designing a practical methodology for selecting business process support services.

Keywords: cloud computing, services, business process, workflow.

1 Motivation

The current shift to the cloud computing is the fourth revolution in the application of computer technology to business that the authors have observed: counting the shift to mini-computers as the first revolution, shift to the personal computers as the second, and shift to the Web as the third one. Each technological revolution extended the usage of IT technology to the new areas, and gave much broader assortments of systems to business people. Each revolution also resulted in higher involvement of business people in systems development due to appearing easy to use tools, like dBase, Access, Dreamweaver, and user-centered methodologies, e.g. agile development.

The shift to the cloud computing is to continue the trend above. Inexpensive cloud computing services ready for deployment at the moment notice will allow business people to choose services they consider appropriate for them. The burden to consult the IT department whether there is an appropriate IT-infrastructure to support a

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 16–30, 2012.

system they need will be lifted from their minds. The same applies to financial concerns; it will be possible to test a new service without high initial investments.

Business process solutions providers are already moving their products to the cloud offering them as services [1]. There already exist a number of such services, ranging from those that provide totally domain independent solutions [2] to the ones that are focused on supporting processes in a specialized domains like Salesforce [3] (CRM), or ProjectPlace [4] (Project management solutions). With growing popularity of cloud-computing, it is only expected that the number of different process support services will grow. The question arises of how a business person, e.g. a department manager, can decide on which service suits best his/her needs. The paper investigates this issue in respect to the services that provide fully customizable operational support to business processes.

Services that provide customizable process support are based on different principles; some of them are totally workflow based, others are based on case-management [5], including adaptive case management [6]. The suitability of a particular type of services depends on the process in question. Some business processes can be streamlined and optimized making workflow services to be the right match for operational support. For others, a social software service, like wiki, can be an appropriate choice. Therefore, choosing the right type of business process support (BPS) service for a particular process requires understanding this process nature. For example, employing workflow requires splitting the process in a number of predefined operations that can be ordered in a systematic way, while employing a wiki-type service does not require even to identify operations.

The goal of this paper is to suggest a framework for high-level analysis of a business process that *allows to determine requirements on a service without building a detailed process model (as details are better to investigate in terms and concepts of an already chosen service).*

The goal is achieved by splitting the process into relatively large chunks of work, called steps, and investigating relationships between them, such as input/output dependencies, possibility of parallel execution, intersecting teams, and some others. Presence or absence of particular relationships is then used for identifying requirements on a service aimed at providing support for the process.

The ideas presented in this paper have been derived from analysis of our own experience of building BPS systems (including cloud BPS services [7]) and introducing them in operational practice. Our experience in the latter shows that not all service features that theoretically could be imagined as useful, are really useful in practice, and more nuanced analysis of the needs when choosing a BPS service is required than just desiring to have everything that is technically possible. More on our experience, see, for example, [8,9].

As far as literature is concerned, we have not found a practically-oriented framework for choosing computerized operational support for business processes. However, the literature does identify the limitations on the scope of applicability of particular methods; see for example definition of workflowability in [10].

The paper is written according the following plan. In Section 2, we introduce main concepts needed for the high-level analysis. In section 3, we present our framework and show how it helps to understand the complexity of a business process that we want to support. In Section 4, we list the capabilities that one can expect from BPS services. In section 5, we suggest guidelines for specifying which capabilities to choose based on the nature of the process in question. Section 6 discusses the results achieved and plans for the future.

2 Basic Concepts

2.1 Business Processes and Process Support Services

There are many definitions of what a business process is, each of them highlighting different aspects of this phenomenon. Actually, term *business process* encompasses two concepts (which often confuses outsiders), *business process type* and *business process instance* (or case). We give both concepts the following pragmatic definitions sufficient for the issues discussed in the paper:

- *Business process type* (BPT) is a plan/template for handling business situations of a certain type
- *Business process instance/case* (BPI) is a situation (being) handled according to the plan/template

The plan/template can include information on any or a combination or all of the following:

- A situation that warrants application of the plan, i.e. triggers a new instance creation
- A goal to reach
- Sub-goals and an order in which they could/should be achieved (goal decomposition)
- Operations/actions/activities that should be completed for achieving goals/sub-goals and the order in which they should be completed (operational decomposition)
- Rules of responsibility/participation (both for sub-goals and operations)
- Rules of collaboration/communication between participants pursuing common goals/sub-goals (communication/collaboration channels)

For example, consider a situation of developing a customized software system for a particular customer. A general plan for handling this situation can be presented as a simplified flowchart in Fig. 1. To this flowchart, any number of details can be added, e.g. the first step in Fig. 1 should be carried out by requirements engineers, the second step should produce use-case diagrams, or the third step requires using Java as a programming language. The more details are added, the more rigid the process will be. For example, setting the requirement that all programming should be done in Java will force the developers using this language even in cases where it does not fit, e.g. for development of operating systems.

The plan/template can reside in any or a combination of all of the following:

- In the heads of members of staff who participate in the process instances of the given type (tacit knowledge). This knowledge guides the process participants what can/should be done or/and what is prohibited, without much thinking about it.

– As written documents, including process maps and other kind of process description (explicit knowledge) on the paper or inside a computer, e.g., in the form of web-based hypertext. These documents contain explicit instructions of what can/should be done or/and what is prohibited to do.
– In software systems/services used to support running process instances (built-in knowledge). The usage of such systems/services forces to do some actions in a certain way and/or in a certain order, or/and prohibit to do it in other ways.

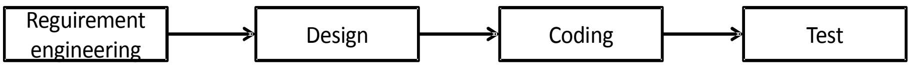


Fig. 1. A plan/template for handling a situation when there is a need to develop a customized software system

In other words, the knowledge on processes can range from being completely tacit (e.g., resides in the heads of the process participants), to being totally explicit (e.g., depicted in detailed process maps).

We define *Business Process Support* (BPS) Service as a cloud service that helps the participants of a business process instance to follow the plan/template defined by the business process type. It can, for example, automate certain operations or support coordination/collaboration between the workers who participate in the same process instance. Note that using a BPS service for operational support does not imply that the whole process needs to be defined in the term of this service, and the service needs to supports all operations included in the process.

2.2 Process Steps

As was pointed out in the introduction, the property of the process that we want to introduce is based on splitting the process in chunks of work called steps. To do this, we need to introduce some concepts related to the idea of process step. We start with identifying concepts related to the instances and then proceed to abstracting them to the concepts that belongs to the process type.

- Each process instance has a *goal* to reach, for example sell to a customer one or more particular products from the company's assortment for a given price.
- An instance goal, usually, can be decomposed into a number of *sub-goals* that could be pursued sequentially or in parallel.
- Pursuing a sub-goal produces *results* that are used when pursuing other sub-goals.
- Often, a sub-goal cannot be reached at once; thus there is a need for recording the *progress* achieved when pursuing this sub-goal.
- Reaching sub-goal require resources that can be divided into two categories: *passive*, like energy or money, and *active* or *agents* that perform actions, like people, or robots.
- To reach a sub-goal an agent or several agents need to perform one or more *operations/actions/activities*

As we consider process instances that belong to the same type to be similar to each other, we assume that their instance goals, results, sub-goals, are also similar. This allows us to define meta-concepts for all concepts listed above, i.e. meta-goal, meta-result, meta-sub-goals, etc. A meta-concept means having a pattern with place-holders (variables) that can be used for any instance. For example a meta-goal for the sales process can be roughly defined as a following sentence:

To sell customer X product Y for the minimum price Z having budget B for the effort

When it is clear from the context that we discuss process types rather than instances, the prefix meta- is omitted.

In the rest of the paper, we consider only "somewhat structured" processes. The minimal requirement on the process structure is that a (meta) goal of the process type is decomposable in several (meta) sub-goals, and pursuing of each sub-goal could be entrusted to different process participants or groups of participants.

Now, we are ready to introduce a new concept of *process step* as a *sub-goal with associated to it components* – results, participants, operations. The concept is applied to both instance level and type level. On the instance level, a step represents a particular sub-goal, result achieved so far, people engaged in achieving the sub-goal, and operations – the ones already completed and those that are planned. On the type level, the step represents a sub-goal template, roles of participants to be engaged, template for formatting the result. Graphically, process steps are represented as boxes (rectangles), as it is done in the systems development process in Fig. 1. This process will be used in the rest of the paper for illustrating the ideas being developed.

3 The Framework

To formulate requirements on a BPS service we need to understand the complexity of a business process to be supported from different points of views. Here, we are looking at the complexity of the process itself, not the complexity of operations included in the process. A process that includes complex operations completed in a strict order without any needs for participants of the process to communicate with each other is considered to be a simple process. A process that includes relatively simple operations completed iteratively is considered to be a complex process, especially if its participants need to collaborate when completing these operations.

Our analysis of process complexity is based on investigation of relationships between the steps identified in the process. Relationships are represented with the help of a set of square matrices where both columns and rows correspond to the process steps. Intersection between a row and a column in a matrix shows a relationship between two steps. The type of content in the cells depends on the relationship in question.

3.1 Input-Output

The *input-output* matrix shows dependencies of one step on the results achieved in another. A cell (*a,b*) in the matrix, where *a* refer to a column and *b* – to a row, specifies what result (i.e. output) from step *a* (if any) is used as input to step *b*. In addition to the name of result, a cell can be marked with asterisk (*) which means that the result is required for step *b* to be started the first time. An example of input-output matrix for the process in Fig. 1 is presented in Table 1. In Fig. 2, the input-output dependencies are presented in a graphical form.

Table 1. Example of input-output relationships

Output / Input	**Requirements**	**Design**	**Coding**	**Test**
Requirements				
Design	*Requirements specifications			Test results
Coding		*Design specifications		Test results
Test	*Test specifications		*Code	

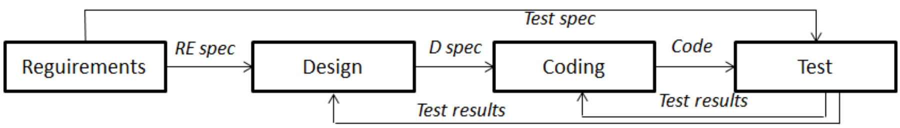


Fig. 2. Graphical representation of Table 1

Presence of a symmetric pair of non-empty cells (*coding, test*) and (*test, coding*) in Table 1, points to a loop in steps execution, i.e., return from *test* to *coding* for bug fixing (Fig.2). To make all loops explicit, we can take the "transitive closure" of the *input-output* matrix creating a derived matrix called the *transitive input-output* matrix, see Table 2. In it, cell (*a,b*) is marked with cross *x* if (*a,b*) is nonempty in the *input-output* matrix, or there is a sequence of steps $c_1,\ldots,c_n$ such that cells $(a,c_1),(c_1,c_2),\ldots,(c_{n-1},c_n),(c_n,b)$ are non-empty in the *input-output* matrix. In Table 2, there are two pairs of symmetric non-empty cells (*coding, test*), (*test, coding*) and (*design, test*), (*test, design*). The second pair points to the loop of going from *design* to *test* via *coding* and returning to *design* in case the requirements are not satisfied.

Table 2. The transitive input-output matrix derived from Table 1

	Requirements	**Design**	**Coding**	**Test**
Requirements				
Design	x			x
Coding	x	x		x
Test	x	x	x	

3.2 Parallel Execution

The *parallel execution* matrix shows whether two steps are allowed to be executed in parallel. If ongoing activity inside step *a* do not totally forbid carrying out activity in step *b*, then both cells (*a,b*) and (*b,a*) are marked with *x* (the matrix is symmetrical). If none of the steps can run in parallel, the parallel execution matrix will be empty.

Suppose our systems development process template provides for hard project deadlines and allows some degree of parallelism. For example, *requirements* is allowed to partially run in parallel with both *design*, and *coding*, meaning that test specifications from the requirements team are continued to be prepared while the design and coding are already in progress. Such case is depicted in Table 3.

Table 3. Example of parallel execution matrix

	Requirements	**Design**	**Coding**	**Test**
Requirements		x	x	
Design	x			
Coding	x			
Test				

In a process template that provides for very tight deadlines, all steps can be allowed to run in parallel. One starts designing as soon as basic requirements are gathered, and coding when some "implementable" part of the design has been completed. Such course might require a lot of re-doing, but it could be the only one possible if there is no possibility to extend the project length. This approach may succeed provided that the systems development team is experienced and accustomed to work in such a fashion.

3.3 Parallel Dependencies

By combining the *input-output* matrix with *parallel execution* matrix we can get a new view on complexity of a business process. Table 4 is produced by merging Tables 1 and 3 according to a simple rule: cell (*a,b*) get crossed in the new table only if the cell is not empty in both input-output matrix and parallel execution matrix. We will refer to the merged matrix as to *parallel dependencies* matrix. The cross in a cell (*a,b*) in this matrix means that steps *a* and *b* can run in parallel at the same time as *b* is dependent on the result from *a*. In Table 4, there is only one cell that is crossed – (*requirements, design*), which means that steps *design* and *requirements* can run in parallel while *design* depends on results from *requirements* (see deliberations in Section 3.2).

A cross in cell (*a,b*) of the *parallel dependencies* matrix requires special attention as it warrants tight coordination between these steps, otherwise the work done in step *b* may need to be totally re-done after substantial changes in the result from step *a*. Even more tight cooperation is required when both cells (*a,b*) and (*b,a*) are crossed.

Table 4. Example of parallel dependencies matrix

	Requirements	**Design**	**Coding**	**Test**
Requirements				
Design	x			
Coding				
Test				

Sometimes a cross in the *parallel dependencies* matrix appears due to the steps we have chosen are too big. In this case, we can try to remove parallel dependencies by decomposing (splitting) the original steps into smaller ones. For example, we can split *requirements* into two steps: *specifying requirements* (SR) and *specifying requirements tests* (SRT). Then, the input-output matrix may get the form of Table 5, and the parallel execution matrix will get the form of table 6. As the result the *parallel dependencies matrix* becomes empty.

Table 5. The new input-output relationships matrix

	SR	**SRT**	**Design**	**Coding**	**Test**
SR					
SRT	*Requirements specifications				
Design	*Requirements specifications				Test results
Coding			*Design specifications		Test results
Test		*Test specifications		*Code	

Table 6. The new parallel execution matrix

	SR	**SRT**	**Design**	**Coding**	**Test**
SR					
SRT			x	x	
Design		x			
Coding		x			
Test					

Note that it is not always possible to get an empty *parallel dependencies* matrix through decomposition (see deliberation in Section 3.2). Decomposition may not remove all parallel dependencies, or it can introduce new dependencies instead of the old ones. This, for example, happens if we allow steps *SR* and *SRT* run in parallel.

3.4 Weak Dependencies

Cell (*a,b*) in the *weak dependencies matrix* shows whether step *b* may require something more than the formalized result from step *a*, e.g. a historical trace of how the result has been achieved. For example, it is not unusual for the designers to need more information than exists in the formal requirements. They might need to understand the rationale behind one or more requirements, or need some other background information. Cell (*a,b*) in this matrix specifies what kind of information from step *a* might be needed to complete step *b*. An example of such matrix for our systems development process is given in Table 7.

The concept of *weak dependencies* reflects the needs for informal communication in the frame of a process instance. It is not always possible to include everything that might be needed for the next step in the formal results, as different instances might require completely different information from the previous steps. It is better to start looking for this information on the demand basis, i.e. when there is a need for it.

Table 7. Example of weak dependencies

	Requirements	**Design**	**Coding**	**Test**
Requirements				
Design	Rational behind requirements Communication with the customer			
Coding		Clarification of diagrams		
Test				

3.5 Teams and Their Relationships

The *teams* matrix shows the presence of collaborative teams and their relationships. The presence of teams is shown in the diagonal of the *team*s matrix: cell (*a,a*) is marked with the light gray color if the team for step *a* consists of more than one person. The non-diagonal elements show whether the teams participating in different steps intersect. If the teams for steps *a* and *b* intersect but not coincide, we mark both cells (*a,b*) and (*b,a*) with the light gray color. If the teams coincide, we mark these cells with the dark gray color.

An example of *teams* matrix for our systems development process is shown in Table 8. Here, we assume that each step does have a team; *requirements* and *design* teams intersect but not coincide; *coding* and *test* teams coincide.

Table 8. Example of teams matrix

	Requirements	**Design**	**Coding**	**Test**
Requirements				
Design				
Coding				
Test				

The diagonal of the *teams* matrix identifies steps that may require support for intra-step collaboration, which is discussed in Section 5. The non-diagonal part of the matrix is used for analyzing the needs for support of inter-step coordination/ collaboration, which is discussed in Section 3.6 and 5.

3.6 Inter-step Collaboration

By merging the *weak dependencies* matrix (Section 3.4) with the *teams* matrix (Section 3.5), we get a view on the needs for inter-step collaboration. The result of the merger of matrices in Tables 7 and 8 is presented in Table 9. In it, one non-empty cell (*requirements, design*) has the light gray background, the other one (*design, coding*) has the white background. In the first case, *requirements* and *design* teams intersect, thus additional information from one step to another can be carried out tacitly via intersecting members. In the second case, the *design* and *coding* teams do not intersect, thus there is a need to make information (other than formal design documentation) that might be needed from *design* for *coding* available on demand. Considering that the design team can be dissolved before coding starts, or be not easily available, this issue needs to get special attention.

Table 9. The weak relationships matrix merged with teams matrix

	Requirements	**Design**	**Coding**	**Test**
Requirements				
Design	Rational behind requirements Communication with the customer			
Coding		Clarification of diagrams		
Test				

4 BPS Services Capabilities

In this section we discuss capabilities one can expect for a BPS service to provide. The term capability here is understood as ability to provide support for certain aspect of running business process instances. The capabilities could be provided separately or tied up in a clump where one cannot be used without the others. The list of basic capabilities that we believe could be expected from BPS services is presented below.

The list has been compiled mostly based on our experience of supporting business processes, and analysis of BPS tools from other vendors. Many of the capabilities listed below are also mentioned in various research works. However, due to the lack of space, we cannot produce a detailed analysis of the literature on the topic in this paper. We do not insist that the list is comprehensive; in this paper it is only important that it includes capabilities the needs for which could be derived from the content of the matrices from Section 4.

1. *Information logistics support* (ILS) is aimed at providing process participants with all information they need to complete their work without being overwhelmed by the details that are not relevant. ILS is particular important for steps where the inputs-outputs constitute information objects, like documents, program code, test protocols, etc. ILS can be provided in two different ways:
   - By actually sending the results to the next step team, e.g. via email. We refer to such kind of logistics as to *conveyor belt logistics* [9].
   - By providing a shared space where the results are stored and made available for the participants of the “next step”. We refer to such kind of logistics as to *construction site logistics* [9].

   The ILS capability can also provide version control for the information objects that are produced more than once. Version control is easier to achieve by using the construction site logistics than the conveyor belt one.

2. *Intra-step collaboration support* is aimed at providing a team working on the same step with means to store/retrieve intermediate results and communicate with each other synchronously and/or asynchronously.
3. *Inter-step collaboration support* is aimed at providing the teams, or individuals working on different steps with means to access intermediate results obtained in each others' steps and communicate between the teams synchronously and/or asynchronously.

   Note that intra-step and inter-step collaboration may require different means of support. In the first case, the communication can be between people of the same profession who reside in the same department. In the second case, it can be between people of different professions who reside in different departments.

   Note also that term collaboration in this paper is used in a special way. It does not cover cases of accepting inputs from the previous steps, or forwarding outputs to the next steps. The latter two cases are considered as belonging to the ILS issues.
4. *Process flow restrictions enforcement* ensures that the rules establish for the process flow are strictly followed. Examples of such rules:
   - Ensure that the steps that cannot run in parallel run in turns.
   - Ensure that if a step needs a result (output) from some previous step it waits until the latter step is finished.
5. *Process flow support* ensures smoothness of the process flow; it sees to that the steps that can be activated (i.e., inputs are ready) are activated at once. This for example, can be done by informing process participants that they should start working on their step through sending them required inputs (when the conveyor belt ILS is employed), or notifying that the inputs have been placed in the shared space designated for them (when the construction site ILS is employed).
6. *Participation restrictions enforcement* ensures that "right" people are participating in various process steps. The rules can be established because of external legislation (e.g., Sarbanes-Oxley) or decided on internally. The rules concern who can participate in what steps, which information is available to each kind of participants, whether the step teams can intersect, etc.
7. *Resource assignment support*. This capability means automatic or semi-automatic formation of step teams based on qualifications and availability of process participants.
8. *Support for domain-specific operations*. This capability includes tools to complete operations inside the step, like compiling or testing a program. These tools can be general, like an office package (MS, OpenOffice, etc.), or specialized like compilers for specific languages.

5 Guidelines for Choosing Capabilities

The easiest way to choose a BPS service is when the process is workflowable in a high degree [10]. With the help of our framework workflowability can be defined as:

– The *parallel execution* matrix is empty - dependent steps are executed in turns
– The *teams* matrix is empty – one and unique person per step
– The *weak dependencies* matrix is empty (only formalized inputs are relevant for steps execution).

In this case, any service that provides a workflow solution, e.g., [1], would be suitable.

In a case where workflowability is not present and cannot be obtained by decomposition of steps (see Section 3.3), each capability in the list of Section 4 needs to be considered separately against the properties of the business process in question. Fig. 3 shows which matrices from Section 3 can be used for determining the needs for which capabilities. In addition to relationships between capabilities and matrices, the bottom part of Fig.3 shows which matrices are basic – white background, and which are derived – gray background and arrows into them. Note that in this paper, only capabilities 1-6 of the list from Section 4 are covered.

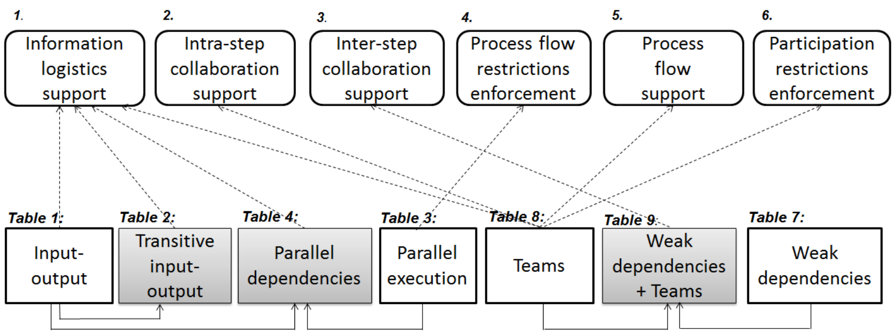


Fig. 3. Guidelines for identifying capabilities

The preliminary guidelines for choosing BPS service capabilities are as follows:

1. *Information logistics support.* The capability is desirable as long as the *input-output* matrix (Table 1) identifies results in form of information objects that need to be passed between the steps. It is less critical when the *teams* matrix (Table 8) identifies that the step teams intersect. In this case, responsibility of moving the results from one step to another could be assigned to the intersecting parts of the step teams and be completed outside the frame of a BPS service. If the teams coincide, there is even less need for information logistics support.

   In cases when the *transitive input-output* matrix (Table 2) shows that there are no iterative loops (no symmetric non-empty cells in it) and the *parallel dependencies* matrix (Table 4) is empty, the conveyor belt information logistics will work satisfactory. When the loops are present, there can be many versions of the same information objects. When these versions are just sent from one step to another, there is a risk that a wrong version will be used instead of the right one. Having a shared space (construction site ILS) where a new version totally substitutes the old one would be preferable in case of loops. Having version control for such shared spaces can give additional advantages in case one needs to understand the difference between the old and new version.

A non-empty *parallel dependencies* matrix (Table 4) requires even more attention to information logistics. This is especially so when the teams of two steps with parallel dependencies neither coincide nor intersect (Table 8). In this case, any new piece of information should also include explanation on whether it is a complement to what already has passed, or substitution of the old piece, or both. Having dedicated shared space with version control and explanatory comments would constitute appropriate information logistics support in this case.

2. *Intra-step collaboration support.* The capability is desirable when the *teams* matrix (Table 8) identifies steps that do have teams (cells marked by the light gray background in the diagonal of the matrix). The capability should allow to store and share the intermediate results. In addition, it may include messaging and on-line communication, like chat, voice or teleconferencing. The basic needs could be solved by a shared space structured according to the needs of the team, with or without version control capabilities. Such a space can include forums for discussions, and journals to record the internal or external events, e.g., communication with customer/supplies. A step shared space could be useful even when a step "team" consists of one member (the white background in the diagonal of the *teams* matrix). This is true when he/she cannot complete the whole step in one go and need to return to the step several times before it is completed.
3. *Inter-step collaboration support.* The needs for this capability can be identified by the merged *week dependencies + teams* matrix (Table 9). The capability is desirable when there are non-empty elements in the matrix. The needs for this kind of support is even greater if the teams for steps with weak dependencies do not intersect – non-empty cells in the matrix with white background. One way of arranging such collaboration is by having a communication channel between the teams. The (week) dependent step team can send a request for extra information, and get it back through the same channel. This will work if the members of the team which have the information are still available for questioning.

   Another way of arranging inter-step collaboration is possible if the shared spaces technique is employed for intra-step collaboration. If the step shared space is made accessible to the team of a dependent step, the members of the latter can themselves find the information they need. This will work provided that the shared spaces are structured in a way that makes it easy to navigate in them for the process participants who do not participate in the correspondent step.
4. *Process flow restrictions enforcement.* This capability is desirable in case of the *parallel execution* matrix (Table 3) is empty or sparse. If many steps can and should run in parallel one may have very little use of this capability.
5. *Process flow support.* This capability is very useful if the *teams* matrix (Table 8) shows that steps teams neither intersect nor coincide. However, it does not harm to have it even if they do intersect. In case when many steps can run in parallel and steps teams do not intersect, a more sophisticated coordination mechanism is required than just process flow support, see the ILS-related discussion above.
6. *Participation restrictions enforcement.* This capability might be needed if steps teams do not coincide, which can be easily figured out from the *teams* matrix (Table 8). The actual need for this capability depends on the reasons that are not

revealed by the steps relationships matrices. If the conveyor belt information logistics is employed, the restrictions are established by sending results only to the participants of the steps in which these results are to be used. If shared spaces are employed for information logistics support, the participation restrictions are realized by limiting access to some parts of shared spaces.

6 Discussion and Plans for the Future

In the Sections 3-5 above, we have demonstrated how a business process can be analyzed with the help of steps relationships matrices, and how the needs for various BPS capabilities can be derived from the content of these matrices. The ideas presented in this paper can be summarized in the process of choosing BPS services presented in Fig. 4. The ideal situation here would be if the only steps in the process that require human intervention where *identifying steps* and *filling basic matrices.* The rest would be done pretty formally, or even fully automated.

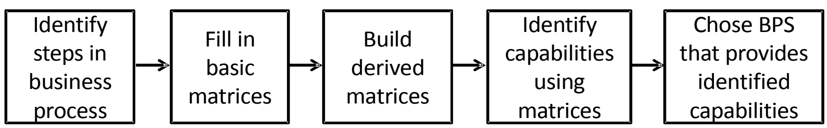


Fig. 4. A simplified process for choosing BPS

The first problem on the way to the ideal is the last step of the process – choosing a BPS services based on the list of capabilities. The following issues can be identified here:

– BPS vendors do not describe their services in terms of capabilities we listed. They, usually, do not provide information in such terms as information logistics, or process flow restrictions. The descriptions are in terms of a business domain at which the service is aimed, or/and in form of functional specification. To formalize the last step in Fig. 4, there is a need to introduce a standard on capabilities provided by BPS service vendors, or design a practical methodology of BPS service analysis that produces a list of capabilities. The latter approach seems to be more feasible. In addition, just having a list of capabilities provided by a BPS service may not be enough, as there can be dependencies between them, so that some capabilities cannot be provided without the others. These dependencies need to be revealed so that choosing a service that provides one useable capability with a set of unusable ones could be avoided.
– When choosing a BPS service, one also needs to take into account other requirements than capabilities from our list. Security provision and SLA level are typical examples of such requirements. Easiness to customize the service to a particular process is another example. Different vendors will be providing capabilities in different ways, e.g. conveyor belt vs. shared spaces, using different modeling notations to specify details of the process in order to make customization. As the final selection of service can depend on additional requirements, they also need to be listed and understood.

The second problem concerns the following. Suppose we have done analysis according to the guidelines in Section 5, and established a list of capabilities to be requested from a BPS service. Should we seek a service that provides all these capabilities? To answer this question, one needs to take into account factors that characterized the environment in which the process instances are running. To such factors, for example, belong:

– *People engaged in the process*. For example, with a low staff turnover, and already established efficient ways of personal communication, one might choose not to impose a new collaboration mechanism, as it can create a resistance to using a new service. An opposite situation, i.e. with a significant staff turnover, warrants standardization of collaboration mechanisms.
– *Dynamism of environment*. For example, if a process definition is expected to be the same during considerable period of time, it can be advantageous to have a capability for strict process flow enforcement. If, however, the process definition is to be changed quite often, this capability might be useless, especially if the time for customizing the service to a new definition is comparable with the time to the next change.

Our future plans include investigation of the two problems identified above, as well as testing the main ideas in practice. Designing computerized support for the process in Fig. 4 is also on our research agenda. To accomplish this task we would, probably, need to introduce more derived matrixes than it was done in Section 4, and establish some structure in the capabilities list from Section 5. The next problem on the agenda is choosing a BPS service suitable for multiple business processes.

References

[1] Apian cloud: BPM, `http://www.appian.com/bpm-software/cloudbpm.jsp`
[2] ActionFlow, `http//, http://www.actionflow.com`
[3] SalesForce, `http://www.salesforce.com`
[4] Projectplace, `http://www.projectplace.com`
[5] Van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. Data & Knowledge Engineering 53(2), 129–162 (2005)
[6] Swenson, K.D. (ed.): Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done. Meghan-Kiffer Press, Tampa (2010)
[7] iPB Reference Manual. On-line documentation, `http://docs.ibissoft.se/node/3`
[8] Andersson, T., Bider, I., Svensson, R.: Aligning people to business processes experience report. Software Process Improvement and Practice 10(4), 403–413 (2005)
[9] Bider, I., Johannesson, P., Perjons, E.: In Search of the Holy Grail: Integrating Social Software with BPM Experience Report. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 1–13. Springer, Heidelberg (2010)
[10] Baresi, L., et al.: Workflow design methodology. In: Grefen, Pernici, Sanchez (eds.) Database Support for Workflow Management: The WIDE Project, pp. 47–94. Kluwer (1999)

A Platform for Recombining Process Knowledge Chunks

Sandeep Purao[1], Arvind Karunakaran[2], and Brian Cameron[1]

[1] College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA, United States
{spurao,bcameron}@ist.psu.edu
[2] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, United States
arvindk@mit.edu

Abstract. We describe ReKon, a platform that contains fine-granular templates, each describing a process knowledge chunk, which may be combined, as needed, to support ad hoc processes needed in large projects. The platform allows user-lead (re)-construction of a process to deal with the unique and emergent needs of a project by leveraging prior knowledge encoded in process chunks. We have populated ReKon with process chunks created from more than 1200 real-world project templates contributed by four consulting organizations. The fine-granularity of process chunks contained in ReKon represents a direct response to the emergent nature of large projects that defies high-volume production processes. The paper develops the underlying meta-model and operations for ReKon that adapt and extend the combination quadrant in Nonaka and Takeuchi's knowledge creation framework. A two-phase evaluation illustrates the need for ReKon and points to its potential usefulness.

Keywords: ReKon, Process Management, Process Chunks, Granularity, Recombinable Process Chunks, Systems of Systems.

1 Introduction

Large and complex projects are now a norm, both in the physical world and in the software industry. In the software industry, as our understanding has evolved, we have crossed a threshold: we no longer construct software systems as monoliths [1]. Instead, we envision them as 'systems-of-systems' [2]. Constructing these is a challenge that further exacerbates the problems of schedule and effort variances, cost overruns and project failures [1, 3, 4]. The difficulties lie in the process of converting blueprints to workable products [1, 5]. As a result, the track record of projects to implement systems of systems tends to be below average (see, e.g. [6]). These challenges provide the context for the work we describe in this paper. For large and complex projects, such as those aimed at building 'systems-of-systems', there are few prescriptive workflows and processes. This lack of processes is an indicator of the project variations and the associated complexity [1].

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 31–45, 2012.

Although these projects share some characteristics, (e.g. they lack a well-defined end-point, involve multiple teams and even organizations) – these meta-level similarities do not dictate the actual work carried out, i.e. it is difficult to specify production workflows for these projects. Further, the projects are often emergent i.e. the work required is difficult to anticipate [7]. Comprehensive process prescriptions for projects aimed at building and deploying systems of systems are, therefore, difficult to pin down as well as difficult to translate and transfer across contexts.

A number of process design and management approaches may be explored to address the problems outlined above [8-11]. All of these, however, rely upon a key ingredient: knowing what to do and how to do it. Coarse-granular processes that provided these prescriptions for traditional software engineering efforts – often under the guise of systems development methodologies – cannot meet this need. The research question we address is, therefore, the following: *Is it possible to devise a platform that can provide fine-granular process assets that project members may combine, as needed, to address the unique needs of projects aimed at building systems of systems?*

This paper develops such a platform. It consists of fine-granular process chunks, mapped against broad project phases and tasks that potential users can combine as needed in response to the unique and emergent needs of the project. The process chunks are extracted from more than 1,200 real-world project templates collected for this research project from four leading IT consulting organizations. The paper describes the method followed for chunking and codification; the conceptual model that underlies the platform that draws from and extends upon Nonaka and Takeuchi's knowledge creation (SECI) spiral [12]; and an implementation. A two-phase evaluation of the platform with users demonstrates potential usefulness of the approach. The key contribution of the paper is a conceptual model of recombinable and fine-granular process knowledge chunks, a prototype implementation, and preliminary evaluation.

2 Related Work

2.1 A Motivating Example

Consider the following scenario faced by John, Mary and Sam, engaged in a systems integration project for one of their clients.

> John, Mary, and Sam are deciding how they should approach a new integration effort for Painters R Us. The effort involves constructing a catalog of services from a legacy scheduling application so that contractors and customers can use these, and outlining a set of processes to work with other legacy applications in a similar manner. Their concerns revolve around scalability and security but they are also unclear about how they might go about aligning their current business processes. Project Templates that codify lessons learned from earlier efforts do not permit a direct mapping against all the concerns they are facing. In each of these coarse-grain templates, often running into scores of sections, there are detailed instructions and worksheets for structuring different tasks such as working with clients to uncover requirements, designing, and testing.

John, Mary and Sam realize that their specific project with Painters R Us is likely to require several components that are difficult to find in a single template. Based on their exploration, they also know that much of what they will need is likely to be available but spread across several existing templates constructed by others. However, they do not know how they can leverage different parts of templates to construct one that may be best suited for their project. After spending several weeks trying to understand and select the templates that can guide their efforts, they are now locked in an internal debate and find themselves moving towards creating an ad hoc approach relying largely on their experience instead of leveraging the process knowledge contained in the excellent worksheets and examples available in the templates. They foresee returning to this quandary repeatedly as the project progresses.

The problems faced by John, Mary and Sam are not new. Templates (as containers of process knowledge) have been traditionally used in several organizations. They represent, among other things, rules, routines and best practices accumulated from past projects, which are codified into concrete artifacts. Prior work, however, tells us that such overt emphasis on codifying process knowledge into an explicit form can create certain pathologies [13]. For instance, excessive emphasis on codifying process knowledge can lead to problems such as information overload, making attention, rather than information, the scarce resource. Search costs go up, leading individuals to create more "new" templates instead of reusing existing ones [14]. The adherence to these project templates can also cause problems such as anchoring and unnecessary reuse [15] [16] triggering a vicious cycle, where templates can hurt - instead of help - task, project and organizational performance [13, 15].

2.2 Projects to Build Systems-of-Systems as the Context of Study

Projects which involve the building of systems-of-systems provide a useful context for this study. Systems-of-systems represent "a collection of … systems that (are brought) … together to obtain a new, more complex (entity that) offers more functionality … than simply the sum of the constituent (parts)" [17]; that is, it is a system composed of other autonomous systems such that no single constituent has visibility of the overall system [7]. Systems of systems display emergent behavior; and therefore, require different approaches to building. As a result, they are often confronted with problems discussed in the previous section.

Few workflows, processes, and methodologies have been proposed for building systems-of-systems [1, 2, 6]. For example, Lam and Shankararaman outline one for Enterprise Integration that is portrayed as three concentric rings that represent key aspects: – a process to solve the Enterprise Integration problem, the deliverables produced by the process, and risks that must be managed during the process [18]. The prerequisites for success and reasons for failure of projects involving systems-of-systems project are, however, still not well understood [3, 4, 6, 12-14]. On the other hand, the lengthy and complex nature of these projects leads to accumulation of process knowledge – which can be contained and codified into project templates (as forms, checklists, and best practice guides and more), contributing to the concerns outlined earlier related to 'information overload'. The 'emergent behavior' exhibited by projects of this nature can be out-of-sync with the rules and routines that those project templates try to enforce.

Past studies related to systems development focus on traditional software development or maintenance (as opposed to building 'systems of systems') (e.g. [19, 20]). A significant finding from these studies is that poor practices can lead to project escalation and runaway projects [15-17]. Although a body of knowledge is available elsewhere [9, 11, 21] [22] project failures persist and have been attributed to several reasons including unsuitability of approaches for the intended task [8-11, 19, 20, 23, 24]. In spite of this extensive stream, the relationship between process knowledge needs of project members and its relation to the success/failure rate of the project, however, remains largely unexplored [1-4, 7, 17]. This is especially critical for projects that are intended to implement systems of systems because process knowledge needs of project participants are not only unique across projects but can also change as a consequence of the emergent behavior of the project.

2.3 Process and Knowledge Management

Several definitions of processes have been offered in prior work, based on varying perspectives on process and workflows [25, 26, 27]. For example, according to the Workflow Management Coalition (WfMC), an international standards-setting organization, a business process refers to "a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships;" whereas a workflow refers to "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [28]. Both emphasize the need for answers to fundamental questions such as who does what and when and with the help of which resources. The many approaches to process design and workflow management – such as production workflows [26], collaborative processes [27], adaptive workflows [25] and others can, in effect, be traced to the variations in timing, accuracy and confidence in these answers. Therefore, we argue that the problems of knowledge management are intricately tied to the problems of process management.

There are many approaches and frameworks that deal with knowledge management [29-32] that may be adapted for process knowledge. In this regard, an early framework from Nonaka and Takeuchi [33] - popularly known as the 'SECI' framework - is useful to understand a key issue: the distinction between tacit and explicit knowledge. Tacit Knowledge is rooted in action and experience of an individual and is specific to a given context [33]. On the other hand, explicit knowledge refers to knowledge that could be articulated, represented and communicated, either through symbolic forms or through natural language [33]. In the context of business processes and workflows, these forms of knowledge are analogous to how processes are enacted in situated contexts by organizational players (using tacit knowledge) versus how processes codify the expectations of tasks and dependencies across tasks (as explicit knowledge). Nonaka and Takeuchi [33] describe knowledge creation – following this framework – as a continuous process that begins with the tacit knowledge of an individual [33, 34], and moves forward as a spiral through the phases of socialization, externalization, combination and internalization (returning to individual tacit knowledge) [12, 33, 34] (see Fig. 1).

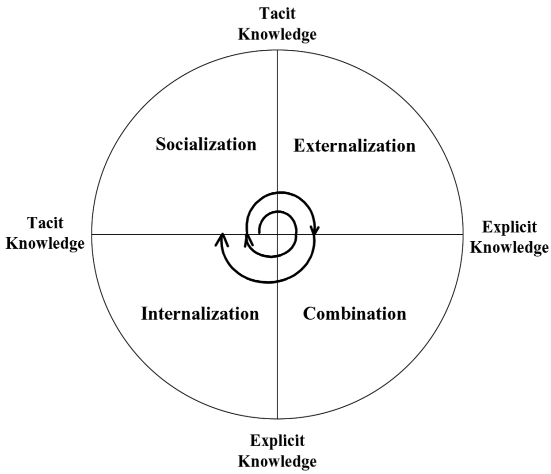


Fig. 1. The SECI Knowledge Creation Framework [34]

Based on the review of prior work related to building of systems-of-systems (see section 2.2 above), we argue that for process management issues related to the design and building of systems of systems, the tacit and explicit knowledge tends to be mutually dependent [25, 35, 36]. Tacit dimensions of process knowledge forms the background to give rise to a structure to develop and interpret explicit process knowledge [25, 35]. We, therefore, suggest that these reinforcing qualities are not always exploited by existing workflow systems, which tend to focus more on the codifying explicit process knowledge [25, 37] with a view to supporting production processes, and sometimes, allowing for degrees of variation. Process management systems and practices rarely, if ever, provide affordances to allow this mutual evolution of tacit and explicit process knowledge.

Consider, for example, projects that involve the building of systems-of-systems. Several anecdotes, lessons learned and best practices continue to be accumulated because of the inherent complexity within each such project. These are often codified as boilerplate templates (e.g. checklist, forms, effort estimation excel macros, requirements gathering word document and more). As new projects are tackled, it is likely that these instances of process knowledge (i.e. checklists, forms and other best practices) codified onto templates may not be fully congruent with the emergent needs. It is also likely, though, that they may not be totally useless or out-of-sync either. Parts of some templates may be useful and other parts may not be useful. For example, the first five sections of Template A might be relevant for Project X, but for Project Y, only sections 3-7 might be useful. It would therefore, make sense for the members and managers of Project X to combine sections 1-5 from Template A with sections 6-8 from Template B; and for members and managers of Project Y to combine sections 3-7 from Template A with sections 1-2 from Template B.

Such affordances to facilitate recombinations in order to address the mismatch between process knowledge needs of a project and process knowledge from prior efforts remains largely unavailable and unexplored [5, 6, 11]. This would require leveraging and some re-thinking of templates as carriers of process knowledge, and recasting their role to one that allows the mutual evolution of tacit and explicit process knowledge [38]. We elaborate this in the next section.

3 Recombinable Process Knowledge Chunks

The review of prior work above allows us to recast the problems with two fundamental assertions: first, we conceptualize templates as carriers of process knowledge; and second, we position templates as facilitators that balance between fluid and institutional domains of knowledge. We develop these two assertions next.

3.1 Templates as Carriers of Process Knowledge

We conceptualize templates as carriers of process knowledge [12, 34] that are created, stored and retrieved. Consider, for example, the template for specifying requirements shown in Fig. 2 below. It captures a structure for a requirements gathering task, and includes pointers that project participants can use to orchestrate the task, including potential sub-tasks. The screen-shot shows a fraction of a much larger document.

Such documents exemplify the dominant form in which the templates are specified by IT consulting organizations. They represent systemic knowledge assets [12], i.e., knowledge systematized and packaged in forms such as documents, specifications and manuals. They represent a key milestone in the SECI spiral that starts with tacit

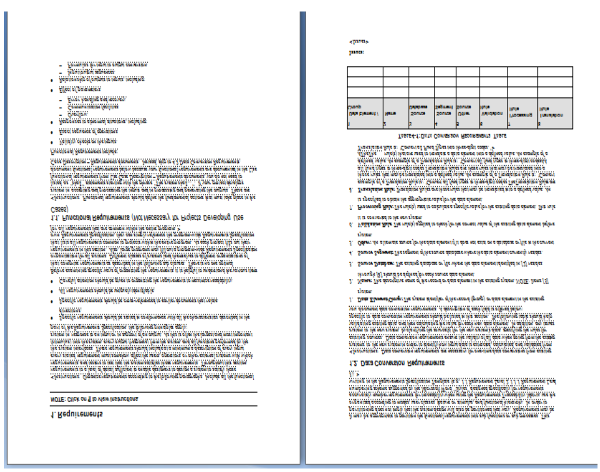


Fig. 2. A Process Chunk: Part of a Project Template for Gathering Requirements

knowledge, obtained through work experience (experiential knowledge), articulated in symbolic form (conceptual knowledge), captured in templates (systemic knowledge). This milestone, in Nonaka and Takeuchi's conceptualization [12], is followed by moving the templates to practice via routines (routine knowledge). The templates, as systemic knowledge assets, therefore, provide a visible point of entry as carriers of process knowledge. They provide the opportunity to move across the tacit-explicit as well as the individual-collective dimensions, as they contribute to the knowledge creation spiral.

3.2 Balancing Emergent Needs against Best Practices

The research challenge, therefore, is not solely focused on capturing the tacit process knowledge and making it explicit in the form of, say, procedures and templates. Instead, we conceptualize the research challenge as designing an approach that would balance the 'fluid' and the 'institutional' domains of process knowledge [38]. The argument in this paper is that processes for projects that must deal with unique situations should leverage not only past best practices but should also allow room to address emergent project needs. The 'fluid-ness' would allow appropriation and improvisation; the 'institutional' would emphasize rules, routines and procedures.

Finding a balance between these two domains [38] is a challenge that we try to address. We do this by conceptualizing a mini-spiral within the larger SECI spiral that allows individuals to balance situated performance with codified practice (see Fig. 3).

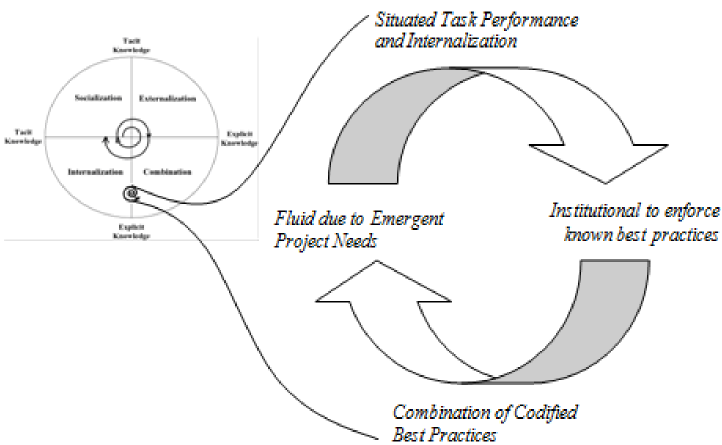


Fig. 3. The Mini-Spiral within SECI [40]

This mini-spiral – that straddles the Combination and Internalization quadrants in the SECI framework – is intended to facilitate reuse of process knowledge chunks. The ReKon platform, described next, is aimed at realizing this possibility.

4 The ReKon Platform

The meta-model for ReKon builds on the foundations outlined above. The meta-model conceptualizes process chunks as logical components needed for tasks within phases of a systems-of-systems development and implementation effort. The physical template chunks, then, represent instances of these process chunks.

Process chunks and Development projects

$l,m \in L$	Logical Process chunks
$p \in P$	Phases in a systems-of-systems development project
$t,u \in T$	Tasks in a systems-of-systems development project
need (l,t,p)	Process chunk l is needed for Task t, Phase p {0,1}

Templates and Process knowledge chunks

$s \in S$	Source Templates
$j, k \in K$	Physical process knowledge chunks
part (k,s)	Physical process knowledge chunk k is from source template s{0,1}
instance (k,l)	Physical process knowledge chunk k is an instance of Logical process chunk{0,1}

Retrieval

{k | (retrieve (instance (k,l) | need (l,t,p))}
Retrieval of Physical Process knowledge chunks {k} that represent instances of Logical Process chunks (l) needed for Task t, Phase p

The simple meta-model provides the foundation for the mini-spiral within SECI (see Figure 3 above). The users may retrieve and combine process chunks, internalize, and even contribute to the ReKon repository, continuing the spiral.

4.1 Creating Process Knowledge Chunks from Templates

To populate ReKon, a large set of templates (~1220) were collected from four leading IT consulting organizations. The classification of templates was facilitated by a matrix of ‘Phases’ and ‘Tasks,’ constructed by consulting Project Management Institute’s PMBOK [22], Lam and Shankarraman’s ‘Enterprise Integration’ methodology [18] and Brownsworld’s ‘Systems of Systems Navigator Approach’ [1].

Phases include Planning, Market Research, Requirements Gathering, Tool Comparison, Design, Development, Assessment, Implementation, Testing, and Deployment; and Tasks include IP Waiver, Status Reports, Reviews, Statement of Work, Requirements Declaration, R.F.P Development, Tool Guidelines, Client Interaction (Knowledge Transfer) and Client Interaction (Knowledge Elicitation). At the intersection of the Phases and Tasks are Cells, where the fine-granular process chunks are placed. We created these template chunks through parsing a single coarse-granular template into logical task-level units. For example, a client interview protocol represents a process chunk – a fine-granular knowledge unit - for conducting interviews (task) during gathering requirements (phase). Interview protocols available in multiple templates – say, for different variety of clients (e.g., SMEs, Large Enterprises), or for different types of projects (for e.g., Web Development, Legacy System Maintenance) - were separated and made available. Project members may access these (retrieval) and combine to create new templates (target template).

Table 1. Inter-Coder Reliability for Chunking and Coding Templates into the ReKon Matrix

Round	Templates Coded	Number of Coders	Inter-Coder Agreement
1	122	2	78%
2	122	3	86%

This procedure was done in a structured manner. First, a random sample of 122 documents (approximately 10% of the set) was chosen for the chunking and initial classification. Coders established common heuristics for chunking and classifying these templates. After the first round, the chunking and classifications were compared to check consistency and differences were resolved via discussion. The common terms were enhanced based on discrepancies and the following discussion. For the second round, another coder was added, and chunking and coding was done on a separate random sample of 122 documents (another 10% of the set).

Table 1 shows the results and inter-coder agreement obtained during the two rounds of coding, which were 78% and 86% respectively, showing high levels of agreement [41]. The complete set (~1220 templates) were then divided and randomly assigned to coders who chunked a single template into logical fine-granular process units, and then assessed the fit for each template to a particular cell in the matrix. Finally, the templates assigned to each cell were examined to select the best templates. This involved rules of thumb such as number of sections, thoroughness of descriptions, and availability of examples. The result includes physical process chunks ($j, k \in K$) that are instances (instance(k,l)) of logical process chunks ($l \in L$) that correspond to knowledge needs (need (l, t, p)) of phases ($p \in P$) and tasks ($t \in T$).

4.2 A Prototype Implementation

The process chunks created as above were used to populate and implement the ReKon platform. The prototype contained 90 cells in the matrix. Of these, process chunks could be created for 36 cells from the templates contributed by the consulting organizations. The total number of process chunks in the matrix was 92, that is, an average of 2.5 process chunks in each non-empty cell. The ReKon implementation was a conceptualized as a simple browser that would give potential users access to the fine-granular process chunks (Template Chunks) based on a selection of the broad project phase, and a task within that phase. Two versions of the ReKon project Template Browser were created to ensure that the front-end did not play a significant role in how it was perceived. Figure 4 below shows one of the implementations.

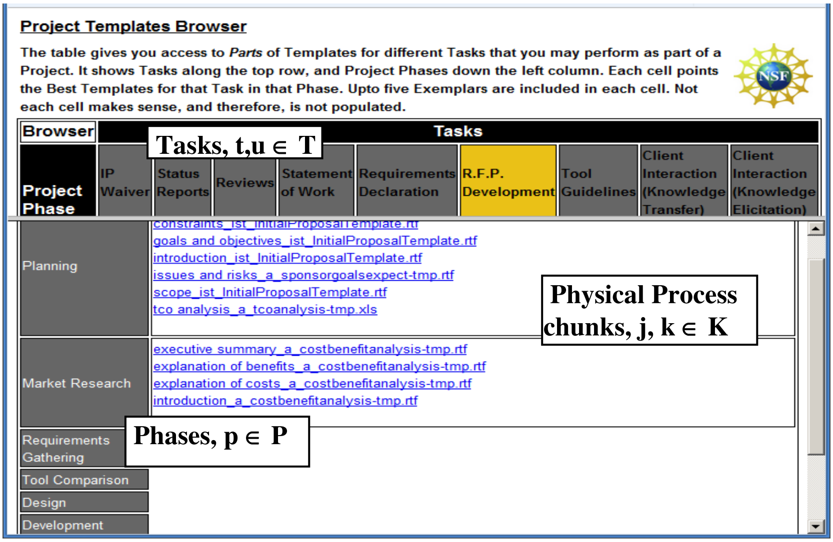


Fig. 4. A Snapshot of the ReKon Prototype

It outlines the Phases along the leftmost column, and the Tasks along the top row. Choosing a Task in the top row shows the template chunks available to structure the task for each Phase. The figure shows the Task RFP development and Process chunks available for this task for the Phases: Planning and Market Research. The two versions of the prototypes formed the basis for the evaluation effort.

5 Evaluation

Evaluation was conducted in two phases, broadly described as pre- and post- ReKon. The pre-ReKon phase was carried out to assess users' evaluation of coarse-grain templates. The post-ReKon phase then assessed users' evaluation of process chunks contained in ReKon. Users were recruited from a second course in a series (titled Advanced Enterprise Integration) engaged in working on real-world projects to implement integration solutions (systems-of-systems). The Pre-ReKon evaluation, conducted prior to the introduction of the ReKon Platform assessed the use and relevance of (sections within) coarse-grain templates traditionally used. The students were also asked to comment on the comparative usefulness of a coarse-grain template versus a hypothetical scenario where different sections from the template would be available separately. The Post-ReKon evaluation, conducted after allowing the students a few weeks to explore the platform assessed concepts such as granularity, size, appropriateness of classification and relevance of process chunks, as well as their usefulness for project needs. The evaluation was considered formative [42] because its intent was to provide input to further improving the platform.

5.1 Results from the Pre-ReKon Assessment

The results from the pre-ReKon assessment provide an interesting snapshot of user perceptions. The results, (n=28) are summarized below. These prospective users were asked to respond to several prompts to understand their perceptions about having the large templates available for reuse versus the possibility of having template chunks (as process knowledge chunks) available for reuse. Their responses were recorded on a scale of 1 (agree) to 5 (disagree). For the purpose of this paper, we elect to report on the responses to two prompts from this assessment.

One prompt, “Although I may not have used all sections, it is useful to have the complete template,” resulted in an average score of 1.42, i.e., most respondents leaned towards agreement. The respondents were also asked to provide free-form comments in response to the prompts. Representative responses included the following:

- useful to have the template because most of the information we have to come up with ourselves so to have a guide line to fill in is very helpful to the success of this project
- it is difficult to determine if a section is relevant or not. Figuring out what needs to be included is work in itself

Another prompt, “It is better to have each section available separately, so we can create the document we need by combining the sections relevant to our project,” resulted in an average score of 2.42, suggesting much more ambivalence. Free-form comments in response to this prompt included the following:

- Most groups will not use all sections and it may be easier to make your own document
- I'd rather error on the side of caution when it comes to including all possible subsections. Having the sections available separately poses the risk of missing something

The responses indicate strong ambivalence. The apparent paradox underscores the arguments about balancing the fluid and the institutional. Responses about relevance of individual sections of the coarse-grain template were also analyzed (See Table 2).

Table 2. Relevance of Individual Sections of the Coarse-Grain Template

Selected Sections	Relevant to
Scope Definition	97% of respondents
Project Assumptions and Constraints	91% of respondents
Related Projects	33% of respondents
Test Approach	51% of respondents

Although some sections of the Template were found useful across the board, others were not considered relevant. Each element in the template was considered not relevant by at least one respondent. These responses provided a baseline that allowed a move to understand whether the effort to create and classify process chunks, the ReKon platform, would be perceived to be appropriately structured. Because the users were allowed to explore the platform for a few weeks, and assessing because the platform continues to go through refinement, final assessment such as usefulness and effectiveness were not evaluated.

5.2 Results from the Post-ReKon Assessment

Although 60% of the participants had indicated preference for having template sections in the Pre-ReKon assessment, the exact nature of these sections was open to question. During this research, the ReKon platform was populated with process chunks. The post-ReKon assessment was, therefore, aimed at assessing properties such as granularity, size, appropriateness of classification, and relevance of process chunks for project needs. Two separate prototypes were used with different front-ends for this evaluation, and randomly assigned to two user groups. Other than a significant difference in the number of minutes spent exploring the prototype, the answers to other questions did not indicate a difference. Prima facie, this suggests that the user interface did not influence the outcome, allowing us to report assessment of the underlying process chunks. Table 4 shows the results.

Table 3. Assessment of Process Chunks

Criteria	Outcome (N=29)
1. Size of Process Knowledge Chunk (1 – Too Small ; 5 – Too Long)	2.65 (SD=0.93)
2. Knowledge Needs satisfied by a Process Knowledge Chunk (1 – All needs ; 5 – Very little Needs)	2.71 (SD = 1.01)
3. Relevance of Process Knowledge Chunk (1 – All relevant ; 5 – None relevant)	2.82 (SD = 1.02)

In addition, the respondents were asked to assess whether there were ‘too many’ (1 on the scale) or ‘too few’ (5 on the scale) process knowledge chunks in a cell. For this question, two cells were chosen: one with 3 units, and another with a larger number (6+) of units. The respondents’ answers were 3.27 and 4.27 respectively for these two cells. The results are encouraging because it is difficult to operationalize rules of thumb such as how large a knowledge unit should be, and how many process chunks should be available for a task/phase combination. A more direct assessment was provided by whether the process chunks satisfied knowledge needs (Item 3 in the table above), and whether the knowledge unit was considered relevant (Item 4 in the table above). These were assessed for a knowledge that the participants randomly chose. The overall results for classification, however, indicated that a given knowledge unit could be classified under multiple phases and tasks pointing to additional future work.

Un-structured comments from the respondents provided further understanding of the results. Many participants suggested that additional meta-data, or “*further explanation may be needed in the templates. … to have a better understanding.*” One participant suggested that a“*… quick view feature that open it up in a tiny thumbnail to view*” would let them locate and retrieve appropriate information more quickly. Several participants commented that some process chunks need to be more granular as *"some of the "chunks" … should possibly be re-worked to make them easier to understand"*. A few felt that it was more helpful *"to have one complete document to look (at) and …then fill in the sections that are relevant to our project"*. Together, these responses and not only provided formative feedback, but also added further support to the underlying proposition of the need to balance the 'fluid' and 'institutional' domains of knowledge [38].

6 Conclusion

Designing and deploying processes for large projects that are characterized by scale, complexity and significant variations is a difficult problem. The problems presented by projects aimed at building 'systems-of-systems' provide a useful domain to study these process design and deployment concerns. The traditional response to process design – packaged and delivered as methodologies for systems development and maintenance [1, 2, 17] – is clearly inadequate in this context. We have traced this concern to the problem of *meeting process knowledge needs* of participants in such projects.

In our work, we have recast this problem as one that requires balancing two conflicting demands – use of institutionalized work practices, often codified as templates that provide anchors to the organizational processes, and ensuring fluid-ness that is required due to the varied and emergent project needs. The solution we have proposed rests on a refinement of the SECI framework [33], with addition of a mini-spiral [40]. The paper describes a platform that contains process chunks, and allows project managers to reuse and recombine these process chunks, as needed. The paper also describes the meta-model that underlies the process chunks and their organization within the platform, which is populated with physical process chunks created from more than 1,200 templates contributed by four leading consulting organizations.

Evaluation results indicate that the fundamental ideas underlying recombinable, fine-granular process knowledge assets are likely to be valuable for configuring new processes in response to large, complex and varied projects (similar to those involving the building of systems of systems). Our future research is aimed at improving the ReKon platform based on feedback obtained, employing the platform in other domains to understand the possibilities of creating, storing and recombining process knowledge chunks, and additional empirical studies to further understand effectiveness of the ideas, both as a mini-spiral that facilitates reuse of process knowledge chunks, and its potential implementation as a platform described in this paper.

Acknowledgements. The work reported has been funded by the National Science Foundation under award numbers 722112 and 722141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF). We also acknowledge contributions of project templates by and discussions with consulting organizations.

References

1. Brownsword, L.: System-of-Systems Navigator: An Approach for Managing System-of-Systems Interoperability, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2006)
2. Boehm, B.: Some future trends and implications for systems and software engineering processes. Systems Engineering 9(1), 1–19 (2006)
3. Lam, W.: Investigating success factors in enterprise application integration: a case-driven analysis. European Journal of Information Systems 14(2), 175–187 (2005)
4. Purao, S., Paul, S., Smith, S.: Understanding enterprise integration project risks: A focus group study (2007)
5. Brownsword, L., Smith, J.: Promises, expectations, and realities of interoperability: from cots to systems of systems (2007)
6. Charette, R.: Why software fails. IEEE Spectrum 42(9), 36 (2005)
7. Fisher, D.: An emergent perspective on interoperation in systems of systems, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2006)
8. Fichman, R., Keil, M., Tiwana, A.: Beyond valuation: Options thinking in IT project management. California Management Review 47(2), 74–96 (2005)
9. Keil, M., et al.: A framework for identifying software project risks. Communications of the ACM 41(11), 76–83 (1998)
10. Keil, M., et al.: Why software projects escalate: The importance of project management constructs. IEEE Transactions on Engineering Management 50(3), 251 (2003)
11. Wallace, L., Keil, M.: Software project risks and their effect on outcomes. Communications of the ACM 47(4), 68–73 (2004)
12. Nonaka, I., Toyama, R., Konno, N.: SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation. Long Range Planning 33(1), 5–34 (2000)
13. Garud, R., Kumaraswamy, A.: Vicious and Virtuous Circles in the Management of Knowledge: The Case of Infosys Technologies. MIS Quarterly 29(1), 9–33 (2005)
14. Hansen, M., Haas, M.: Competing for attention in knowledge markets: Electronic document dissemination in a management consulting company. Administrative Science Quarterly, 1–28 (2001)
15. Haas, M., Hansen, M.: When using knowledge can hurt performance: The value of organizational capabilities in a management consulting company. Strategic Management Journal 26(1), 1–24 (2005)
16. Parsons, J., Saunders, C.: Cognitive heuristics in software engineering applying and extending anchoring and adjustment to artifact reuse. IEEE Transactions on Software Engineering 30(12), 873–888 (2004)
17. Popper, S., et al.: System of systems symposium: Report on a summer conversation. Potomac Institute for Policy Studies, Arlington, VA, pp. 21–22 (2004)
18. Lam, W., Shankararaman, V.: An enterprise integration methodology. IT Professional 6(2), 40–48 (2004)
19. Keil, M.: Pulling the plug: software project management and the problem of project escalation. MIS Quarterly, 421–447 (1995)
20. Lyytinen, K., Hirschheim, R.: Information systems failures—a survey and classification of the empirical literature (1988)
21. Tiwana, A., Keil, M.: The one-minute risk assessment tool (2004)
22. Duncan, W.: PMBOK–A Guide to the Project Management Body of Knowledge. ZDA: Project Management Institute, PMI (1996)

23. Napier, N., Keil, M., Tan, F.: IT project managers' construction of successful project management practice: a repertory grid investigation. Information Systems Journal (2008)
24. Verner, J., Evanco, W.: In-house software development: what project management practices lead to success? IEEE Software 22(1), 86–93 (2005)
25. van der Aalst, W., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2002)
26. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall PTR, Upper Saddle River (2000)
27. Sheth, A.: From Contemporary Workflow Process Automation to Dynamic Work Activity Coordination and Collaboration. Siggroup Bulletin 18(3), 17–20 (1997)
28. The Workflow Management Coalition, Terminology & Glossary, Document No. WFMC-TC-1011, http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf
29. Petri Nets World, http://www.daimi.au.dk/PetriNets/
30. Han, Y., Sheth, A.: On Adaptive Workflow Modeling. In: Proceedings of the 4th International Conference on Information Systems Analysis and Synthesis, Orlando, Florida, pp. 108–116 (1998)
31. Ellis, C.A., Keddara, K., Rozenberg, G.: Dynamic change within workflow systems. In: Comstock, N., Ellis, C.A. (eds.) Conf. on Organizational Computing Systems, pp. 10–21. ACM SIGOIS, ACM, Milpitas, CA (1995)
32. BPMI.org, BPML|BPEL4WS: A convergence path toward a standard BPM stack, August 15 (2002)
33. Nonaka, I.: A dynamic theory of organizational knowledge. Organization Science 5(1), 14–37 (1994)
34. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, USA (1995)
35. Polanyi, M.: The Tacit Dimension, New York (1967)
36. Tuomi, I.: Data is more than knowledge: implications of the reversed knowledge hierarchy for knowledge management and organizational memory (1999)
37. Schultze, U., Leidner, D.: Studying Knowledge Management in Information Systems Research: Discourses and Theoretical Assumptions. MIS Quarterly 26(3), 213–242 (2002)
38. Graham, A., Pizzo, V.: A question of balance: case studies in strategic knowledge management. European Management Journal 14(4), 338–346 (1996)
39. Garud, R., Kumaraswamy, A., Sambamurthy, V.: Emergent by Design: Performance and Transformation at Infosys Technologies. Organization Science 17(2), 277 (2006)
40. Karunakaran, A., Purao, S., Cameron, B.: From 'Method Fragments' to 'Knowledge Units': A Fine-Granular Approach. In: International Conference on Information Systems, Phoenix, AZ (2009)
41. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
42. Weston, C., McAlpine, L., Bordonaro, T.: A model for understanding formative evaluation in instructional design. Educational Technology Research and Development 43(3), 29–48 (1995)

An Exploratory Study on Collaboratively Conceptualizing Knowledge Intensive Processes

Juliana Baptista dos Santos França, Joanne Manhães Netto, Juliana do E.S. Carvalho, Flávia Maria Santoro, Fernanda Araujo Baião, and Mariano Pimentel

Department of Applied Informatics, Federal University of the State of Rio de Janeiro (UNIRIO), Brazil
{juliana.franca,joanne.netto,juliana.carvalho, flavia.santoro,fernanda.baiao,pimentel}@uniriotec.br

Abstract. The relevance of the knowledge involved in organizational activities has already been addressed since earliest management theories. In this context, several works in the literature discuss how a so-called Knowledge Intensive Process (KIP) may be better understood and managed. The first step towards these goals is the identification of its elements. This is not a trivial task, since KIP involve many subjective and complex concepts that are typically tacit to stakeholders, and thus subject to different interpretations. However, a common interpretation of a KIP among all its participants is essential to prevent communication and comprehension problems. This paper presents an ontology that defines concepts and relationships of a KIP. We discuss the results of an exploratory study where a KIP was described by its participants in a collaborative manner, using a storytelling technique. The goal was to explore the use of the ontology as a basis for identifying the elements within the process description.

Keywords: Knowledge Intensive Process, Knowledge Intensive Process Ontology, Process Representation.

1 Introduction

A Business process is a set of resources, together with interrelated and interactive activities, that transform inputs into services or products (outputs). Typically, business processes are planned and carried out to add value to the organization. A business process may be represented by a business process model (and its corresponding diagram in a graphical notation), which usually comprises the control flow of well-structured activities that an organization performs to achieve its objectives.

However, this traditional way of representing a process is not suitable for the so-called Knowledge Intensive Processes (KIP). This type of process comprises sequences of activities based on intensive acquisition, sharing, storage, and (re)use of knowledge, so that the amount of value added to the organization depends on the actor knowledge. They are naturally more complex, since they deal with diffuse and tacit definitions, unpredictable decisions, creativity-oriented tasks and paths, dynamic execution that evolves based on the experience acquired by the actors. All these

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 46–60, 2012.

characteristics difficult the identification of well-structured activities and their control flow in a KIP, as well as KIP representation as a whole. The work of Nurcan and Edme [20] supports the representation of business process with low or high structure, considering in first line its objectives and strategies linked to an intentional driven modeling. Following this understanding, the representation of the process becomes operational, thinking about activities that comprise the process.

Some traditional process modeling approaches like Event Driven Process Chain (EPC) [11], Business Process Modeling Notation (BPMN) [15], Process Specification Language (PSL) [21], and Business Process Modeling Ontology (BPMO) [2]; have been adapted to allow the representation of the intrinsic elements of knowledge within business processes, but these methods do not include all the features necessary to describe a KIP. Besides, the literature shows a set of approaches dedicated to highly-intensive knowledge processes representation including Business Process Knowledge Method (BPKM) [16], Knowledge Transfer Agent (KTA) [23], DECOR [1], CommonKADS [22], Knowledge Modeling Description Language (KMDL) [9], and the work of Donadel [3]. However all of them, as well as traditional process modeling approaches, do not reach all the KIP relevant elements, as shown in [5].

Based on the difficult to represent and organize the knowledge involved in intensive knowledge processes, the Knowledge Intensive Process Ontology (KIPO) was proposed in order to address existing limitations through a new approach that considers the concepts of KIP that relate to traditional business processes [5].

In this paper, we discuss the results of an exploratory study where a knowledge intensive process was described by its participants, in a collaborative manner, using a storytelling technique. The goal was to explore the use of the ontology as basis for identifying the elements within the KIP narrative and organize the knowledge involved in its description. The paper is organized as following form: Section 2 presents related work about modeling KIP; Section 3 describes the KIP ontology; Section 4 discusses the exploratory study and Section 5 concludes the paper and highlights future perspectives of this research.

2 Knowledge Intensive Processes

The Process-Oriented Knowledge Management approach intends to organize and support the organizational processes, as well as to describe the conversion of knowledge within the process. Its objective is to identify, model, analyze and optimize knowledge intensive processes [10].

2.1 KIP Fundamentals

According to [18], KIP are sequences of activities based on the acquisition and intensive use of knowledge, regardless the business type or size. KIP can only be partially mapped through a traditional process model, due to unpredictable decisions and tasks guided by creativity. For [9], new objects of knowledge or information are created by the conversion of existing ones in the process.

Gronau et al [9] propose a list of requirements for modeling KIP based on: (i) Modeling goals: Which goals shall be reached with the modeling? Are they only documentation purposes or do they require an analysis of weak spots and definition of a new process? (ii) Integration of process and knowledge: There should be a unique approach that combines or integrates the process definition with the flow and transfer of knowledge. (iii) Tacit knowledge: Which definition and appreciation of knowledge is used by the models approach? Is there a differentiation between explicit and tacit knowledge? (iv) Knowledge conversion: Are different mechanisms of knowledge conversion considered and expressed separately in the process model? (v) Knowledge flow: Is there a differentiation between information flow and knowledge transfer? (vi) Offer and demand: Is it possible to indicate in the model differences between the offer of knowledge and its demand? (vii) Person-related knowledge: Is the modeling of knowledge restricted to organizational units or is it possible to show knowledge bound to persons? (viii) Comparison of intended and actual level of knowledge: Is it possible to compare the knowledge levels required for jobs with the knowledge people actually have? (ix) View representation: Is it possible to navigate through the models using different views? (x) Knowledge map: Is it possible to generate knowledge maps from the results of process modeling?

Moreover, to enhance the representation of knowledge to business processes, [3] highlighted the key features required to support a KIP as follow: (i) The guidance value stream, making it easier to obtain results with the representation. (ii) Representation of the business model, integrated with the knowledge representation to add value to knowledge within the business structure of the organization. (iii) Prioritization of tasks for the organization of knowledge of what actions should be performed first. (iv) Artifacts of knowledge when there is a need to differentiate knowledge representation from business representation. (v) Dynamic artifacts to enable the representation of the dynamic elements that behave differently depending on the context. (vi) Representation of knowledge skills involved in each process. (vii) Concepts domain to enable the contextualization of the representation environment.

According to the above mentioned, organizing the knowledge in processes like this is not an easy task. Besides, KIP commonly presents a diversity of information sources, and its execution involves many participants and the assistance of many experts, who carry out actions with high levels of creativity and innovation [10]. Several process modeling techniques are found in literature as likely to represent KIP.

2.2 KIP Representation Approaches

The CommonKADS [22] focuses on knowledge representation. Various stages of modeling attempt to establish a structured approach so that knowledge can be managed with the support of technical and engineering tools. Three basic points characterize these demands: the details of the skills involved in process execution, the representation of the processes through artifacts and semantic analysis, and the opportunities for improvement regarding the process and use of knowledge.

The BPKM - Business Process Knowledge Method [16] - presents a meta-model for integrating business process modeling aspects with Knowledge Management.

This meta-model transcribes the four perspectives of a workflow: task (which tasks are executed in the workflow process), organizational (who performs the specified task), logical (in which order these tasks are executed), data (which data is consumed, produced or exchanged between tasks). The meta-model was extended to include knowledge management tasks that support business processes (knowledge perspective) represented by the elements: Knowledge Management Task, Knowledge Object and Knowledge Archive.

The Knowledge Modeling Description Language (KMDL) [9] represents tacit knowledge of the process, besides the explicit knowledge. Thus, the different possibilities of knowledge conversion can be modeled and the flow of knowledge between actors is depicted. Two other approaches of knowledge representation are the Knowledge Transfer Agent (KTA) Modeling Method [23] and the DECOR Project [1]. The first describes how to create knowledge transferring models. The method consists of modeling in three distinct level of detail and possibilities of analysis. DECOR Project delivery context-sensitive organizational knowledge and has its focus in representation of knowledge processes across diagrams embedded in organizational memory. In the method proposed by [3] the value chain of the organization is mapped and the aspects of knowledge that can influence the organizational processes are represented.

The main deficiency observed in those proposals is that none of them includes or addresses all the requirements discussed in literature. Besides, they do not clearly represent important characteristics of KIP, such as: agents that influences the actions; dynamic aspects; collaboration; communication and interaction among actors while they produce knowledge; decision making rationale based on experience and creativity; and rules that might interfere on agents decisions. Moreover, some proposals do not differentiate between tacit and explicit knowledge [16][3][23][22]; and others do not address the representation of artifacts and dynamic aspects of processes and modeling agents [9].

Based on this analysis, the following sections present the Knowledge Intensive Process Ontology [5], which identifies KIP characteristics, and was built to represent the concepts and relationships of a KIP more adequately, providing a common and precise understanding about what exactly is a knowledge intensive process and what takes place in a KIP environment.

3 The KIP Ontology

Ontology is an explicit and formal representation of a shared conceptualization [7]. It is used to create an unequivocal abstraction of reality; one that is comprehensible by humans, for communication purposes. Based on the limitations of the methods described in the last section, we propose an ontology to precisely represent the concepts of this domain, thus providing a common understanding of a KIP environment. The high level Knowledge Intensive Ontology (KIPO) [5] is presented in Figure 1, which highlights the five components proposed for KIP conceptualization: Collaborative ontology, Business process ontology, Business rules ontology, Decision Ontology, and Knowledge Intensive Process Core Ontology (KIPCO).

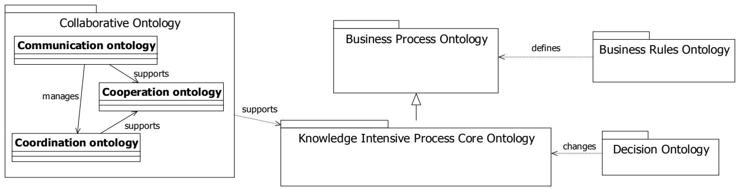


Fig. 1. KIP Ontology Components

The Business Process Ontology (BPO) component is based on the BPMN meta-model [15]. Although it is difficult to display a KIP with all the details of their flows and information previously defined, this is not reason to dismiss the properties applied in the modeling and description of knowledge in structured business processes. In a high level of abstraction, a KIP may be represented as a set of (macro) activities, typically with a simple control flow among them.

The Collaborative ontology was developed by Oliveira [14]. The author defines cooperation as essential to the evolutionary process; communication as a process where people can exchange information, express wishes, emotions and ideas; and coordination as representation of domain elements that are used to promote organization and harmony between concepts of communication and cooperation ontology. These elements are required due to the high degree of tacit knowledge exchanged among stakeholders, since a KIP may evolve along each instance, according to the participant´s interaction. The literature also cites [19] that proposes a process meta-model, which can deal with both well-defined and wickled work procedures and their interactions. However, its focus is on cooperative work processes representation and the interest of KIPO is to understand the cooperation, but the communication and coordination inherent in a KIP too.

The Decision Ontology (DO), presented in [17], makes it possible to adequately explicit all the rationale followed by a professional when making a decision, including the representation of which factors led a stakeholder to make a particular decision. The Business rules ontology is based on [12], and enables a precise and correct representation of the domain rules in which the KIP will be instantiated.

The Business Rules Ontology follows the proposal from Lopes et al [12] that describes the set of Business Rules that restrict a KIP domain. Business rules are relevant for a KIP since it typically defines restrictions that must be followed in the domain of a KIP, and that are the reason for several decisions made by a KIP executor.

The core component of KIPO is the Knowledge Intensive Process Core Ontology (KIPCO), which contains specific KIP elements specializing BPO concepts. KIPCO concepts are further described in this section.

Figure 2 illustrates the proposed KIPO (Knowledge Intensive Process Ontology) in Unified Modeling Language (UML) notation, where gray items are reused from the UFO (Unified Foundational Ontology) formalization [8]. Although some of the KIP properties are addressed by the above-mentioned sub-Ontologies that compose KIPO, the appropriate representation of a KIP requires additional elements/concepts, and relationships. By directing attention to the construction of KIPCO that is core of KIPO, its construction methodology was directed by the five questions from Table 1 that are considered the questions of competence according to the methodologies like [13]. More details about KIPCO construction are described in [5].

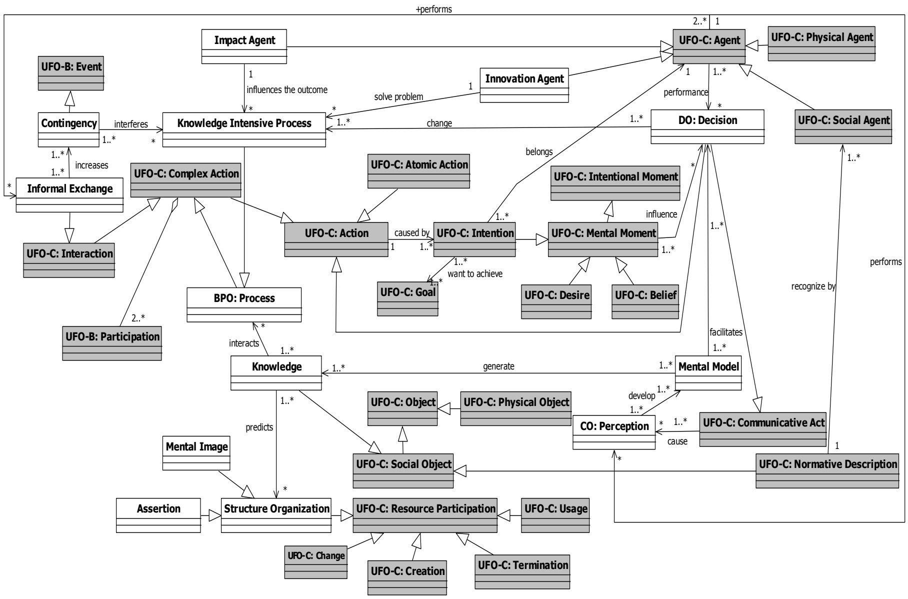


Fig. 2. Knowledge Intensive Process Ontology – KIPO

As shown in Table 1, the answers aim to elicit characteristics of KIP for the composition of KIPCO and consequently for the composition of KIPO, as a whole. For example, as KIP are processes that have the influence of, the first question focuses on the definition of agents that must interact in such processes. Impact agents and Innovation agents are considered as KIP actors. The second question is concerned with the type of interaction that occurs in KIP. Since these processes are highly dynamic and part of knowledge is tacit, many interactions occur informally among the agents to solve problems, make decisions, cooperate within the process execution, and build new knowledge.

The goal of KIPO is to organize the knowledge involved in KIP, and for that, abstractions of the real world must be made. Possible abstractions correspond to the business processes itself. Finally, Table 1 describes which data are exchanged in KIP and which is manipulated and constructed within such processes. The answer is mental models, contingency and decisions.

Regarding the definition of products generated and manipulated by a KIP, their relevance lies in discovering where knowledge is registered to possibly be reused. These products incorporate the knowledge; the perception; the structure in which knowledge is organized; the mental image developed on the agents minds, and the assertions that might present the knowledge formalism.

KIPO may be considered as the consolidation of the concepts from all its components and their relationships. While current KIP modeling methodologies do not fully consider these concepts, the KIP ontology includes them. It is up to the process modelers to appropriately choose a methodology that addresses the different concepts presented in the KIP ontology, allowing the knowledge generated by the process instances to be modeled, stored and reused.

Table 1. Questions used to build KIPCO

Group	Concept	Definition
What types of agents must interact during a KIP?	Impact Agent	This agent performs many tasks at once. The necessary know-ledge to execute KIP actions, normally is found in agent tacit, or is based on previously experiences.
	Innovation Agent	One who is responsible for solving issues in the process with innovation and creativity.
How the interac-tions occur in a KIP?	Informal Exchange	Exchange that occur informally, face to face, or based in docu-mentation.
Which elements are abstractions of the real world?	Business Process	Set of structured activities that seek the transformation of their inputs into services or product.
	Knowledge Intensive Process	Can be semi-structured, structured and unstructured depending on your abstraction, possessing a high degree of dynamism in the objectives' change, high complexity, and dependent on the explicit and tacit knowledge of people involved in the process and the activities that compose it.
Through of what the information are transmitted?	Mental Model	Allows interpretation and improvement of information that create knowledge.
	Contingency	Significant dependence in influences the environment. Tells what motivated interference in the execution process.
	Decision	Identifies information related to the decision as a whole. Informs the solutions taken by the agents so that the process is executed.
What are the elements pro-duced by, or manipulated during, a KIP?	Knowledge	Experiences, values, contextual information and insights that create a framework for improvement and incorporating new experiences and information. The knowledge is derived and applied in people's minds.
	Organizational Structure	The structure in which knowledge is organized.
	Mental Image	Knowledge Organization still remain in the mental sphere. Is developed on the agents with basis in the knowledge built.
	Assertion	They are representations of sense completely abstracted, capable of verbal expression. Present the formalism of knowledge built in process explained.
	Perception	Represents the action of perceiving the message exchanged by agents.

4 Exploratory Case Study

A case study is an empirical research strategy applied to investigate contemporaneous events in their real-life context; those in which the frontier between the analyzed event and its context is not clearly defined. By following this strategy, the researcher has little or no control over the events; therefore he/she cannot manipulate a relevant behavior [24]. In a case study approach, exploratory studies are suggested to conduct initial investigations over a phenomenon in order to build or refine a hypothesis or a theory; explanatory studies, on the other hand, are then applied to confirm or deny the hypothesis or theory [4].

In this work, we conducted an exploratory case study in order to investigate the KIPO [5] with regard to two perspectives: (i) its adequacy for modeling a real KIP; and (ii) its comprehensibility by the stakeholders involved in the KIP execution. More specifically, we evaluated the usage of the KIPO as a basis for discovering elements that characterize a KIP from its description.

The exploratory study was conducted by three analysts in the context of a post-graduate course, and they have deep knowledge about the KIP reported. Applying the Storytelling [6] technique, eight first-year master students were asked to collaboratively tell a story describing the process of elaborating a master thesis, using the TellStory application [6]. During 15 days, each student accessed the tool to create story events that he/she found important for this process and to express their opinions about other events created by another colleague. To come up with a unique and collaborative story, they interacted with each other in an asynchronous way through the tool, highlighting their points of view, reporting their previous experiences and arguing about what was likely to be done in different ways, based on specific knowledge that each one had about a given activity. The analysts did not interfere during the story telling.

At the end of this stage, the history produced by the students was handed to the three analysts, who were asked to separately identify concepts and relationships instances from the KIPO within the story text. Each analyst mapped knowledge elements and built his/her instance of the ontology. Since the focus of the exploratory study was on identifying knowledge elements and instantiating them using the ontology, and not on the modeling language being adopted, this decision was left to the analysts; therefore, each analyst modeled his/her ontology instance using the notation of his/her choice, as long as it comprised notational constructs for concepts and relationships among them.

The analysts were also instructed to report cases in which some identified knowledge element had not a corresponding class in the ontology. Figures 3 and 4 illustrate two out of the three elaborated KIPO instances.

The story told by the students was grouped into eight activities: *Select theme, Set main purpose and specific goals of research, Identify a research problem, Search literature related to the research problem, Review theme and issues, Propose a solution for the problem, Define research method, and Write dissertation*. This organization was carried out to promote the discussion of the actions conducted by students in this KIP. As an example, about the activity "Select theme", the following information has been reported (according to passages extracted from the story):

Participant 1: "Writing the dissertation is the most important and relevant stage of the process; it could lead to an important goal..." "For some people the choice of the theme occurs before starting the course. For others, the issue arises on the basis of ideas developed by teachers and classmates... ";

Participant 2: "I think the theme will also depend on the supervisor.";

Participant 3: "I think this issue is the easiest. The major obstacle is finding a niche to think about a problem and its solution."

Analyst 3 identified ten concepts within this part of the history, which, according to KIPO, are found in the Business Process Ontology (BPO), Knowledge Intensive Process Core Ontology (KIPCO), and Decision Ontology (DO). The concepts of BPO correspond to the activities of KIP. The concepts pointed by KIPCO are related to knowledge, informal exchange, mental model, impact agent and innovation agent. Finally, concepts of decisions to be made and alternatives chosen were also identified. These concepts are listed bellow and appear in the instance of KIPO in Figure 3.

- BPO::Activity::Write paper
- BPO::Activity::Select interest theme
- KIPCO::Knowledge::Select research goal
- KIPCO::Informal_exchange::exchange the ideas developed by students and teachers
- KIPCO::Mental_model::Identification of problem to be worked
- KIPCO::Mental_model::Actions to be taken to solve the research problem
- KIPCO::Agent_Impact::Student
- KIPCO::Agent_Innovation::Student
- KIPCO::Agent_Innovation::Supervisor
- Decision_Ontology::Decision::Decide when to select the theme
- Decision_Ontology::Chosen_alternative::Select theme before get in course
- Decision_Ontology:: Chosen_alternative::Select theme from ideas exchange
- Decision_Ontology:: Decision::Decide research theme

Analyst 2 identified nine concepts. The concepts supported by KIPCO are related to Structure Organization, Knowledge, Innovation Agent, and Impact Agent. The DO provided concepts concerned with decisions to be made in the process, regarding the definition and solving of the research problem, besides the research topic. The concepts identified by the Analyst 2 are listed below and are present in the instance of KIPO in Figure 4:

- KIPCO::StructureOrganization::Write Dissertation
- KIPCO::Knowledge::Problem Solving
- Decision_Ontology::Decision::Research Problem
- Decision_Ontology::Decision::Define Goals
- Decision_Ontology::Decision:: ResearchProblem
- Decision_Ontology::Decision::InterestTopic
- KIPCO::Agent_Innovation::Supervisor
- KIPCO::Agent_Impact::Supervisor
- KIPCO::Agent_Impact::Student

The analysts focused on identifying instances of the KIPCO sub-ontology, since it contains the core KIPO concepts for a KIP. Nevertheless, we argue that the results of this exploratory study show the applicability of this ontology in identifying relevant knowledge elements from a KIP description. The instance created by Analyst 2 (Figure 4) contains 2 relationships that were not prescribed in KIPO: the instance represents that a *perception* can develop *knowledge* (while in KIPO a *perception* develops *mental model*), and that *informal exchange* develops a *perception* (while in KIPO an *informal exchange* increases *contingency*).

Table 2 shows how each knowledge element found in the story text was represented by each analyst in his/her ontology instance. For example (line 3), a Student was represented as an Innovation Agent by analyst 1, as an Impact Agent by analyst 2, while analyst 3 considered a Student as both Impact Agent and Innovation Agent. This table has been consolidated jointly by the three analysts. They tried to approximate semantically the terms used by each one as much as possible.

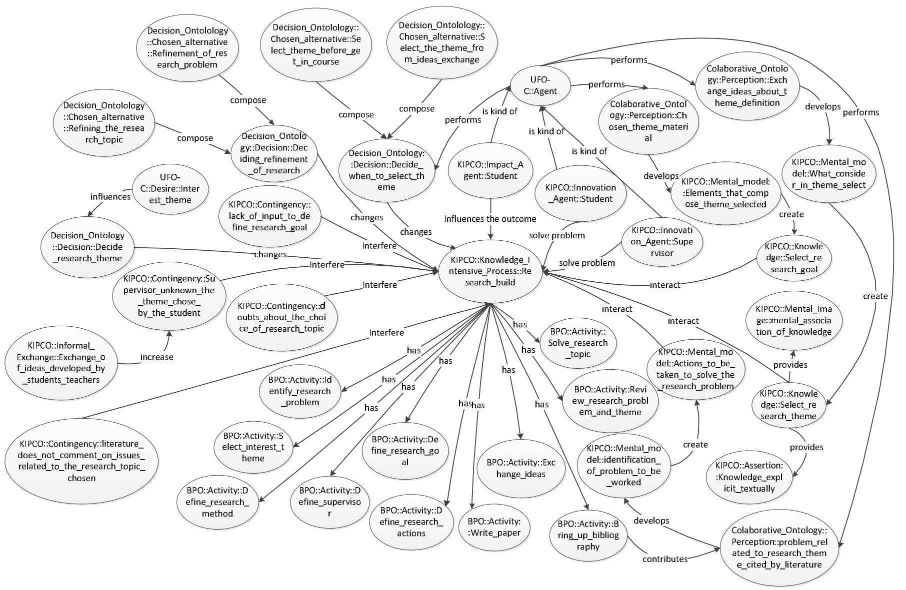


Fig. 3. KIPO instance created by Analyst 3

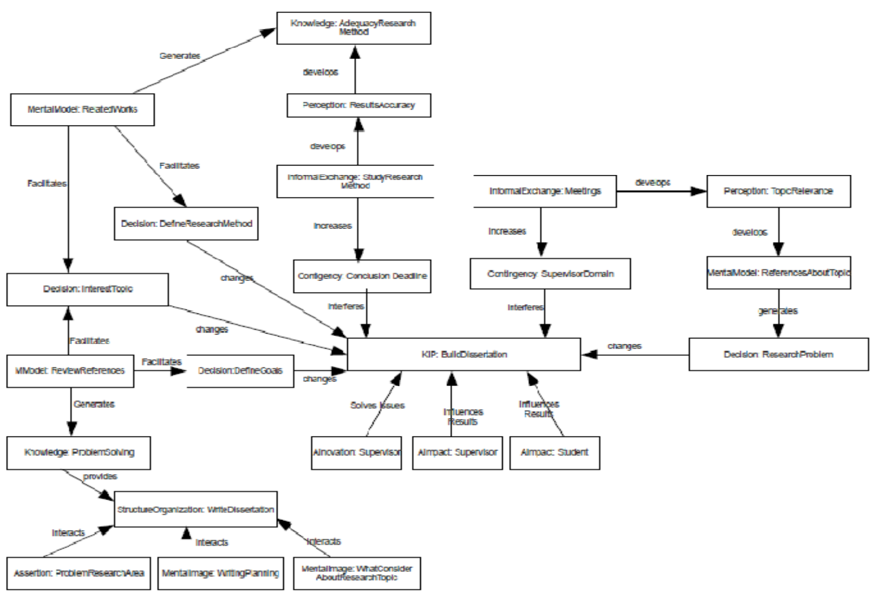


Fig. 4. KIPO instance created by Analyst 2

Table 2. Knowledge elements mapped according to KIPO

Knowledge element	Analyst 1	Analyst 2	Analyst 3
Build dissertation	Knowledge Intensive Process	Knowledge Intensive Process	Knowledge Intensive Process
Supervisor	Impact Agent	Impact Agent Innovation Agent	Innovation Agent
Student	Innovation Agent	Impact Agent	Impact Agent Innovation Agent
Related work cited in literature related to the topic	Informal Exchange	Mental Model	Contigency Perception (Collaboration Ontology)
Research problem identification	Contingency	Decision (Decision Ontology)	Mental Model Activity (Process Meta-model)
Activities to be performed in order to solve the problem	Contingency	Knowledge	Mental Model e Activity (Process Meta-model)
Define the research goals	Decision (Decision Ontology)	Decision (Decision Ontology)	Activity (Process Meta-model) Knowledge
Idea Exchange by students and teachers (meetings)	Informal Exchange	Informal Exchange	Informal Exchange Activity (Process Meta-model)
Supervisor does not know the topic chosen by the student (supervisor domain)	Informal Exchange	Contingency	Contingency
Decide interest topic	Contingency	Decision (Decision Ontology)	Decision (Decision Ontology) Knowledge Activity (Process Meta-model)
Search references/Review references	Mental Model	Mental Model	Activity (Process Meta-model)
Write dissertation	Mental Model	Structure Organization	Activity (Process Meta-model)
Define research method	Decision (Decision Ontology)	Decision (Decision Ontology)	Activity (Process Meta-model)
What to consider while choosing the research topic	x	Mental Model	Mental Model
References about the topic chosen	x	Mental Model	Perception (Collaboration Ontology)
Chose the topic from ideas exchanged	x	Perception (Collaboration Ontology)	Alternative chosen (Decision Ontology)
Interest topic	x	Mental Image	Desire (UFO-C)
Chose the topic before starting the course (personal affinity)	Informal Exchange	x	Alternative chosen (Decision Ontology)
Review topic and research problem	Decision (Decision Ontology)	x	Decision (Decision Ontology) e Activity (Process Meta-model)
Writing planning	Decision (Decision Ontology)	Mental Image	x
Problem from the research area	Informal Exchange	Assertion	x
Exchanged ideas about the topic definition	x	x	Perception (Collaboration Ontology)

Table 2. (*Continued*)

Elements that compose the topic chosen	x	x	Mental Model
Decide the moment to chose the topic	x	x	Decision (Decision Ontology)
Doubt about the research topic to choose	x	x	Contingency
Lack of resources to help defining the research goal	x	x	Contingency
Refinement of the research problem	x	x	Alternative chosen (Decision Ontology)
Refinement of the research topic	x	x	Alternative chosen (Decision Ontology)
Define research activities	x	x	Activity (Process Meta-model)
Define supervisor	x	x	Activity (Process Meta-model)
Study research methods	x	Informal Exchange	x
Adequacy of research method	x	Knowledge	x
Conclusion deadline	x	Contingency	x
Necessity to review literature	Perception (Collaboration Ontology)	x	x
Hypothesis	Informal Exchange	x	x
Results Accuracy	x	Perception	x

Result Analysis and Discussion

The third Analyst elaborated an ontology instance (Figure 3) that covered a larger number of concepts and relationships from the KIP domain, when compared to the second Analyst (Figure 4). This occurred since Analyst 3 knew more about the ontology, thus better understanding its concepts applicability in several scenarios.

No new classes were identified, that is, the concepts contemplated in KIPO were sufficient to identify all KIP elements present in its description. Only two new relationships were presented by Analyst 2. It points to the adequacy of the ontology to represent the process described. Figure 5 depicts the analysis of the mapped elements among all three analysts: 35 elements were mapped; 37% were found in all three instances, and 23% were mapped by at least two analysts, who classified them differently. Some elements were identified only by one analyst, which represented 40% of the total number of elements. The third analyst mapped 80% of the elements, and 20% were also identified by other analysts.

A comparison of the ontology instances points to some difficulties found by three analysts. All the analysts agreed that the story told by the students was not rich in details; therefore it was not easy to identify many of the elements. The divergence in the elements mapping was also due to the fact that each analyst had a distinct interpretation of the described process. This was somehow expected, due to the inherent and well-known ambiguity of natural languages.

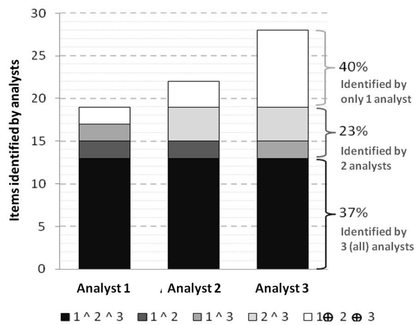


Fig. 5. Common elements identified by analysts

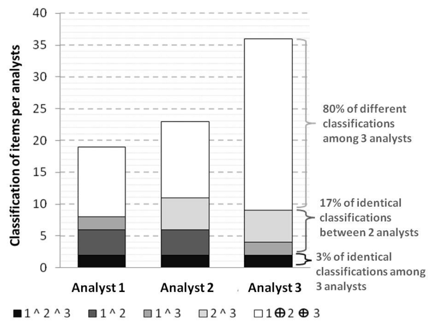


Fig. 6. Ranking analysis of the elements

Figure 6 presents a ranking analysis of the elements mapped by the analysts. This analysis shows that different classifications were the most part (80%). This is an indication that the ontology does not yet provide a good common vocabulary for analysts. The analysts reported that they have had trouble in understanding the meaning of each ontology element; thus, we can conclude that the description of the concepts is not yet been clearly shown by the author.

These concepts only became clear after a meeting conducted by the ontology author with the analysts to review Table 3. This enabled a deeper understanding of the concepts. Thus, another possible conclusion is that the differences in classification of the elements are associated with the fact that concepts descriptions are not clearly provided. Based on it, the need for more relationships pointed by Analyst 2 may have been caused by his difficulty in understanding KIPO.

5 Conclusions

This work presented a case study that explored the conceptualization and representation of a knowledge intensive process (KIP) based on the Knowledge Intensive Process Ontology (KIPO) previously proposed. We evaluated the potential of the KIPO in providing the knowledge organization and an adequate understanding of a KIP. The exploratory study was conducted by three analysts, who generated Ontology instances representing the KIP of elaborating a MSc dissertation. The Ontology instances were generated in a collaborative manner, using the TellStory [6]. The results showed that the set of concepts and relationships (together with their properties) of KIPO were enough to act as a structural model for the KIP being addressed.

The study also evidenced that some Ontology concepts required a more detailed description, to enable a more explicit differentiation among them and facilitate their identification in real scenarios, We also observed that the Ontology instance elaborated by the analyst who had proposed KIPO was richer (that is, containing more concepts) than the other 2 elaborated by the other two analysts. This result evidences that the KIP conceptualization represented by KIPO should be better explained to be fully internalized and comprehensible to be applied. Finally, our reflections on the results from this study led us towards the evolution of KIPO, especially with regard to their concepts descriptions, so that its elements are presented with less ambiguity, thus achieving a higher precision in the Ontology model and its instantiation.

The limitations of our study include the lack of details in the story that described the KIP being addressed, thus making it more difficult for the three analysts to map knowledge items to the Ontology concepts. The representation of distinct ontology elements with similar semantics, as well as the different number of elements in the three instances, hardened the consolidation of all instances into the same mod-el. All the obtained results point to the need of a more precise conceptualization for KIP, using a foundational ontology as a basis, such as DOLCE or UFO.

References

1. Abecker, A.: DECOR Consortium: DECOR – Delivery of context-sensitive organizational knowledge. In: E-Work and E-Commerce. IOS Press (2001)
2. Cabral, L., Norton, B., Domingue, J.: The Business Process Modelling Ontology. In: 4th International Workshop on Semantic Business Process Management (SBPM 2009), Workshop at ESWC 2009, Crete (June 2009)
3. Donadel, A.C.: A method for representing knowledge-intensive processes. 2007. MSc Dissertation. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Brazil (2007) (in Portuguese)
4. Easterbrook, S.M., Singer, J., Storey, M., e Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J. (eds.) Guide to Advanced Empirical Software Engineering. Springer (2007)
5. França, J.S.F., Santoro, F.M., Baião, F.A.: Towards Characterizing Knowledge Intensive Processes. In: International Conference on Computer-Supported Cooperative Work in Design (CSCWD 2012), China (accepted for publication, 2012)
6. Gonçalves, J.C., Santoro, F.M., Baião, F.A.: Business Process Mining from Group Stories. In: 13th International Conference on Computer Supported Cooperative Work in Design, Santiago, Chile, pp. 161–166 (2009)
7. Guarino, N.: Formal Ontology, Conceptual Analysis and Knowledge Representation. International Journal of Human and Computer Studies 43(5/6), 625–640 (1995)
8. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Universal Press, The Netherlands (2005) ISBN 90-75176-81-3
9. Gronau, N., Weber, E.: Management of Knowledge Intensive Business Processes. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 163–178. Springer, Heidelberg (2004)
10. Gronau, N., Muller, C., Korf, R.: KMDL – Capturing, analyzing and improving Knowledge-intensive business process. Journal of Universal Computer Science 11(4), 452–472 (2005)
11. Korherr, B., List, B.: A UML 2 Profile for Event Driven Process Chains. In: Tjoa, A.M., Xu, L., Chaudhry, S. (eds.) Proceedings of the 1st IFIP International Conference on Research and Practical Issues of Enterprise Information Systems. IFIP, vol. 205, pp. 161–172. Springer, Boston (2006)
12. Lopes, M., Baião, F., Siqueira, S.: Expressing Business Rules in a Foundational-based Domain Ontology: Towards Higher-quality Conceptual Models. In: International Conference on Information Integration and Web-based Applications & Services, Paris (2010)

13. Noy, N.F., Mcguinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratoty (2001)
14. Oliveira, F.F.: Ontology Collaboration and its Applications. MSc Dissertation. Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, Brazil (2009) (in Portuguese)
15. OMG: Business Process Modeling and Notation (BPMN). Version 2.0 (2011), `http://www.bpmn.org/` (accessed in: June 26, 2011)
16. Papavassiliou, G., Ntioudis, S., Abecker, A., Mentzas, G.: Managing knowledge in weakly-structured administrative processes. In: 3rd European Conference on Organizational Knowledge, Learning and Capabilities, 3, Atenas, Grécia, Abr. 5-6 (2002)
17. Pereira, A., Santoro, F.: Cognitive Decision Making Process as Context Information. In: The 15th IFIP WG8.3 International Conference on Decision Support Systems (DSS 2010), Lisboa, Portugal (2010)
18. Richter-Von Hagen, C., Ratz, D., Povalej, R.: Towards self-organizing Knowledge-intensive Processes. Journal of Universal Knowledge Management (2005)
19. Nurcan, S., Rolland, C.: Meta-modelling for cooperative processes. In: The 7th European-Japanese Conference on Information Modelling and Knowledge Bases, Toulouse, May 27-30, pp. 361–377 (1997)
20. Nurcan, S., Edme, M.-H.: Intention Driven Modelling for Flexible Workflow Applications. Special issue of the Software Process: Improvement and Practice Journal on Business Process Management, Development and Support 10(4) (2005)
21. Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., Lee, J.: The Process Specification Language (PSL) Overview and Version 1.0 Specification (2000), `http://www.mel.nist.gov/psl/` (accessed in June 27, 2011)
22. Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R., Shadbolt, N., De Velde, W.V., Wielinga, B.: Knowledge Engineering and Management: The CommonKADS Methedology. MIT Press, Cambridge (2002)
23. Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., Easterbook, S.: Analyzing Knowledge Transfer Effectiveness – An Agent-Oriented Modeling Approach. In: Proceedings of the 40th Hawaii International Conference on System Sciences, USA (2007)
24. Yin, R.K.: Case study research: design and methods, 3rd edn. Sage Publications Inc. (2003)

A Proposal for Ownership Representation in the Context of Business Process Models

Fabiana Jack Nogueira Santos[1], Julio Cesar Sampaio do Prado Leite[2], Claudia Cappelli[1], Flávia Maria Santoro[1], and Thaís Vasconcelos Batista[3]

[1] Applied Informatics Department, UNIRIO, Rio de Janeiro, Brazil
[2] Departamento de Informática, PUC-Rio, Rio de Janeiro, Brazil
[3] Computer Science Department, UFRN, Natal, Brazil
{Fabiana.Nogueira,Claudia.Cappelli, Flavia.Santoro}@uniriotec.br, thais@ufrnet.br
www.inf.puc-rio.br/~julio

Abstract. The literature on Business Process Management (BPM) confirms the importance of establishing process ownership but conventional approaches as BPMN or EPC do not offer a way to represent the process owner. The aspect-oriented business process (AO-BPM) approach makes this issue more prominent because of the separation between crosscutting concerns and the core elements of a process. In this paper we present a way to represent the ownership based on the Strategic Actor model from i* and incorporate this approach to the EPC meta-model. We also provide a proof of concept by means of an example that illustrates our solution.

Keywords: Business Process Management, Process Owner, Aspect Owner, Business Process Models, Crosscutting Concerns, Strategic Actor.

1 Introduction

In this paper we argue that process ownership information should be considered in the context of business processes models. We understand that the available representations schemes, aka meta-models, for Business Process Management (BPM) have left out capabilities for representing information regarding process ownership. This came to our attention in the context of improving business process models with the aspect-oriented view (AO-BPM) [1], where we understand that in bringing up crosscutting processes we would need to be more careful about process ownership. In particular, it is important to stress that, in general, representation schemes for BPM just have the possibility of naming an actor for a given lane. However, in most cases, this actor represents either the organization sector or the person who carries out the workflow. Ownership is a more complex concept and should not be left out from the BPM model as it brings out several benefits.

Given this context, we started to work in proposing a representation scheme to fill this gap. The general requirements for this representation can be generalized as follows: a) to be able to describe ownership, b) to be able to differentiate among design time and operation time, c) to be able to provide integration with existing BPM models, d) to consider that processes can be of crosscutting nature to other processes.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 61–75, 2012.

Our proposal is framed on the actor model presented in i* [2] [3]. In the i* framework, the concept of strategic actor is central for representing the strategic view. According to [2], the strategic view helps achieving a deeper understanding of an organization. The Strategic Actor Diagram (SAD) [3] is a model that explicitly represents the actors in the context of intentional modeling. Thus, the SAD was chosen since process ownership should be anchored on owner's intentions (goal, objective). However i* is not a compatible BPM model, since it does not represent process or processes flows. We adapted the ideas of the SAD in order to propose a representation schema that would tackle the above enumerated requirements.

To achieve this goal, we provide a review of the literature regarding process ownership in Section 2. In Section 3 we present some related works about representation of the process ownership and social modeling. Section 4 summarizes the ideas of AO-BPM. Section 5 presents our proposal and the argumentation of the reasons why it covers the general requirements stated above with a proof of concept, by means of an example. We conclude the paper in Section 6 stressing our results and laying out the possibilities of research in order to improve our approach.

2 Process Ownership

The process owner is an important element of any business process management initiative. Hammer [4] has identified two groups of characteristics that indicate how well the business processes can perform and keep the performance. The first group is applied to individual processes and the other one to the whole enterprise.

The first group is named *process enablers*. They determine how well the process can function over the time. The following elements are part of this group: comprehensiveness of a process design, ability of people who perform the process, appointment of a top-level process owner to supervise the process implementation and performance, match between process needs and organization's information and management systems, and quality of the metrics used to evaluate the performance of the process. The second group named *enterprise wide capabilities* focuses on the enterprise capabilities, they are: leadership, culture, expertise, and governance.

The enterprise able to put the elements of the first group in place will have the capabilities of the second group. These groups are highly dependent on each other. They are part of a framework, named PEMM (Process and Enterprise Maturity Model) [4] that help companies to plan and to evaluate their process-based transformations before executing them. According to this framework, each enabler is at some level of development. The enterprise can be considered at a given level only if all enablers belong to the same level, as such the dependence between the enablers is mutual. The relationship between the groups and process performance is such that, if organizational capabilities are stronger, there will be stronger enablers and, as consequence, better process performance.

We highlight the importance of the process owner as an enabler to evaluate process transformation success as proposed in PEMM [4]. According to Hammer [4], the process owner is a senior executive who has responsibility for the process and its results. At level 1 of strength, the activities of this enabler are to identify and to document the process, to communicate it to all the performers and to sponsor small-scale change in projects. At level 2, the process owner articulates the goals of the process' performance and a vision for its future, sponsor redesign and improvement efforts, plan their implementation, and ensure that they are in compliance with the design. At level 3, the process owner works with other process owners to integrate processes to achieve organizational goals. At the very top level 4, the process owner develops a strategic plan for the process, participates in organizational strategic planning and collaborates with his partners for customers and suppliers to sponsor inter-enterprise process redesign.

Jeston and Nelis [5] mention the existence of 3 pillars in Business Process Management: Processes, People, and Technology. The processes refer to the activities performed by the organization and they are associated with objectives and goals. People are considered the key to processes implementation. Technology is the supporting tool to processes and people. People are the core in change management phases during business process adoption. Hengst et al. [6] list 3 types of stakeholders usually involved in change management initiatives: the problem owner, the decision makers, and the analysts or consultants. Each one belongs to one organizational level.

According to [5] there are two categories of management in Business Process Management. One is the management of the business processes integrated with the organizational management and the other regards the management of the processes improvement. In the first category, the managers, owners or administrators of business processes have some responsibilities such as to specify goals and metrics associated to the objectives established, to communicate the objectives, goals, and initiatives to the executors, to monitor and manage the progress of objectives, to motivate the group to overpass the objectives and to solve process disturbances, and to encourage the group in identifying process improvements. One possibility to put this kind of management in practice is the division between senior and middle managers. The former is responsible for the end-to-end process and the latter is responsible for sub-processes or individual processes that are part of the end-to-end process.

Kohlbacher [7] states that a business process must have associated a manager with end-to-end responsibilities. The author presents a research with 44 Austrian metal and machinery industry organizations where 20.45% confirm that an organization process centric causes clear responsibilities because of the process owner role. It reduces uncertainties caused by departmental fragmentation of responsibilities.

In sub-processes or individual processes management, [5] proposes to classify line managers according to the scope of their activities. Operational managers work with clearly defined processes and their goals, adjusting human resources and solving operational issues. Tactic managers focus on possible process improvement and strategic managers concern the business model and their associated processes.

The other category treats process improvement. They are responsible for the identification, development, and benefits from the process management. The responsibilities of these managers are related to the support given to organizational and business managers in improving their processes, with focus on modifications support to reach long term objectives. In this group, we can distinguish the manager of business process management project, the manager of business process management program, the chief of business process excellence, and the chief of process office.

In the first category the managers concern short term goals and the second category cares about improving processes with long term goals in mind. It causes some tension between both groups because any process modification can harm the ability of managers to reach their goals [5].

The main activities of the process owner, according to [5], are: to document the process and to warranty that it is according to the patterns and requirements established; to improve the process. The process owner is the responsible for the decisions, change management, and implementation of improvements; to manage the interface and limits and boundaries of the process; to automate the process; to manage the process performance; to promote the process.

Jeston and Nelis [5] consider the clear and adequate attribution of responsibility and accountability of a process as a challenge in Business Process Management. According to them, an organization can choose to make functional managers responsible for their part of the process, to make functional managers responsible for the end-to-end process and to make non-functional managers responsible for the end-to-end process. Each approach presents some risks as in the functional property of sub-process, the owner may care just about his process and make difficult to implement end-to-end improvements that can damage his part. Another important aspect is the respect of the process owner in the organization.

Larsen and Klischewski [8] confirm that most of the business process approaches consider the responsibility and design of a process as centralized in only one person – the process owner who may have sufficient power to organize and direct the way other actors participate and accept the process reengineering and IT support. The process owner must be responsible for the end-to-end process and must the pointed by the organization leader as he needs authority and personal influence to ensure that the involved make the necessary modification. In order to keep the process owner motivated, his performance must be directly related to the performance of the process.

Table 1 presents a summary of the above mentioned approaches regarding the process owner. As each one has a different objective, it is difficult to compare them; but we can clearly understand the activities and characteristics of the process owner. In [4], [6], and [8] there is no distinction regarding design and execution level. We included this information based on software development processes where the design, implementation and maintenance phases are distinguished. For the best of our knowledge this contribution is not found in literature.

In summary, the process owner's attributions include the design of the process and its operation, which is to ensure that it is followed. In order to make a process operational, a process owner has to obtain resources, to establish and to implement tools to facilitate the process execution, to ensure high performance and to interfere to improve the process always when needed. Another conclusion is the importance of a process owner establishment for the end-to-end processes with the needed authority.

3 Related Work

List and Korherr [9] present a framework to evaluate business process modeling languages. According to them, these languages represent some aspects of a business process and areas of application but there is not a comprehensive evaluation of them. To solve this issue they proposed a meta-model with a wide range of process concepts so it is possible to know the core concepts of business process modeling languages and to perform an evaluation of them.

Table 1. Summary of process owner activities classified into design and execution level

Author	Division	Process Owner Activities	Level
Hammer [4]	level 1	identify and document the process	design
		communicate the process to all the performers	execution
		sponsor small-scale change projects	execution
	level 2	articulate the goals performance of the process and a vision for its future	design
		sponsor redesign and improvement efforts, plan their implementation and ensures compliance with the design	execution
	level 3	work with other process owners to integrate the processes to achieve organizational goals	execution
	level 4	develop a strategic plan for the process	design
		participate in organizational strategic planning	design
		collaborate with his partner for customers and suppliers to sponsor inter-enterprise process redesign	design
Jeston & Nelis [6]	business process management integrated with the organizational management - short term	specify goals and metrics associated to the objectives	design
		communicate the objectives, goals, and initiatives to the executors	execution
		monitor and manage the progress of objectives	execution
		motivate the group to overpass the objectives	execution
		solve process disturbances	execution
		encourage the group in identifying process improvements	execution
	management of the processes improvement - long term	identification, development and introduction of benefits from the process management	design / execution
	main activities	document the process and to warranty that it is according to the patterns and requirements established	design
		improve the process, being responsible the decisions, change management, and improvements implementation	execution
		manage interface, limits and boundaries of the process	execution
		automate the process	execution
		manage the process performance	execution
		promote the process	execution
Larsen and Klischewski [8]		responsibility and design of process	design
		have power to organize and direct the way other actors participate and accept the reengineering and IT support	execution
		responsible for the end-to-end process	execution
		must be pointed leader by the organization as he needs authority and personal influence	execution
		keep the process owner motivated, his performance must be directly related to the process performance	execution

The languages evaluated were UML 2 Activity Diagram, Business Process Definition Metamodel, Business Process Modeling Notation, Event Driven Process Chain, IDEF3, Petri Net, and Role Activity Diagram[1].

[1] This classification was made by the authors; they included languages that were not designed for business processes, like Petri-Nets and IDEF3.

The meta-model is composed of 5 perspectives: organizational, functional, behavioral, informational, and business process context. The organizational perspective is about where and who performs the process elements. The functional perspective represents the process elements that are performed. The behavioral perspective represents when and how the process elements are performed. The informational perspective considers the informational entities consumed, produced or manipulated by the process. The business process context perspective describes characteristics such as goals, measures, deliverables, the process owner, the process type and the customer.

The conclusion of [9] is that behavioral and functional perspectives are well represented in all analyzed languages; organizational and informational perspectives are partly supported; and the business process context perspective is not supported. This is highly important regarding the requirements of our approach abovementioned in Section 1. The organizational perspective is present in almost all of the BPMLs (Business Process Management Languages) except in IDEF 3 and Petri Nets as they have their origin in software development and they do not present the concept of role. All others languages include this concept. None of the languages represent the software with an explicit concept. Just AD represents the boundary of the role as internal or external. The languages AD, RAD, and BPMN use the same concept to represent all types of process participants (internal, external, software, application, service, human, role, and organizational unit). According to List and Korherr [9] this absence of specification prejudices the process enactment. The process owner, part of the business process context perspective, is not represented in none of the 7 analyzed languages. This conclusion reinforces the existence of some gaps on business process modeling, especially this one regarding the process ownership.

Lamb [12] presents a model based on the concept of *social actor*. According to the author, this model should help researchers developing Information and Communication Technology (ICT) studies. This model provides a multi-dimensional view of the organization member and his use of information and communication technologies. It contextualizes the organization members, their informational environments, and ICT. According to the model, social actors are organizational entities, which have their interactions enabled and constrained by the socio-technical affiliations and organizational environment, their members and industry. The social actors have conflicting and ambiguous requirements regarding their activities and the ways they perform their work. This view considers that the world is constantly changing and that the globalization influences the organizational relationships. In this model, the unit of analysis is people and ICT, although the goals are represented in a simple way [12]. In [11] there is no representation of intentionality. As such, these proposals are not able to satisfy our requirements presented in Section 1.

The i* framework is perceived as a social modeling framework where the central concept is the actor. The actor modeling, as a modeling concept, was first used by Hewit [10] as a way to model the work done in organizations named office work. An Actor Architecture and method proposed in Artificial Intelligence [11] is conceptually based in the actor object. In this context, an actor is an active agent that plays a role according to a script. The actor metaphor was used to emphasize the absence of separation between control and data flow in the model of [11]. Considering the Strategic Actor Diagram, in [11] there is no intentionality representation.

Another approach to represent the way organizations operate is DEMO (Design and Engineering Methodology for Organizations) [23]. In this approach, an organization is based on the operational principle of enter into and comply with commitments through communication. The communication occurs between human beings who play actor roles. According to this methodology, it is possible to identify the essence of organizations represented in ontological transactions. The diagrams used to represent the conceptual models of this methodology have the actor role concept but the ownership of, in this case, the ontological transactions is not discussed or addressed. The author shows how data ownership is neglected as one can choose to set the data owner as the initiator or the executor of the transactions but this work does not mention any way to represent this ownership.

4 AO-BPM

AO-BPM (Aspect-Oriented Business Process Modeling) [1] was proposed to address the modularization of crosscutting concerns in business process models. This approach is based on the Aspect-Oriented paradigm [13] that proposes the modularization of crosscutting concerns at software code.

AOSD (Aspect-Oriented Software Development) proposes the following abstractions [13]. *Aspects* encapsulate crosscutting concerns and take them out of the core elements in a given specification or implementation. To specify the composition of an aspect and the core process flow, an aspect contains *pointcuts* and *advices*. *Pointcuts* are sets of *join points*. *Join Points* are the core description elements which an aspect intercepts. Thus join points allow aspect composition, they are core process flow elements where the aspect is applied. A *pointcut* defines an expression with quantification mechanisms to select the join points to be advised by the aspect. A *pointcut* language defines patterns to write *pointcut* expressions. Advices define the action to be taken when a join point is reached. They act on a *pointcut* and can be configured to do so before (before advice), after (after advice), or around (around advice) the joint point.

AO-BPM proposes the separation of the business process model into: (i) *core process* – where is represented the essence of the process and (ii) *aspect process* – where is captured the crosscutting concerns cut across the core process.

The crosscutting concerns [14] [13] appear in conventional models interlaced with other concepts of a process and scattered in many parts of the process. These characteristics lead to several business process elements representing the same concern as they are scattered and tangled all over the process model. The resultant model has reduced understandability and reuse capability and increased maintenance overhead. The conventional modularization of business process models and its abstractions are not able to effectively modularize the crosscutting concerns.

The identification of the aspectual elements in a process model is not a trivial task. To support this discovery, in [1] was suggested some heuristics for aspect identification, they are based in previous work on requirements engineering [15] and are presented as follows: "*(i) if the concept is repeated several times in different places, (ii) if the concept is used by different other concepts, (iii) if the concept* *reflects an integration of semantically distinct situations, (iv) if the concepts represents a decision situation from which different options may be taken, (v) if the concept's absence does not interfere with the global goals of the whole, (vi) if the concept can be reused in other domains and (vii) if the concept is very much independent of other concepts*".

A process model is composed of processes, sub-processes, activities, rules, events, data, actors, and connectors (sequence flow, message flow, and association). It was argued in [1] that all those elements may be identified and represented as a crosscutting concern. The crosscutting concerns can be identified in the context of the same process (intra-process) or among different processes (inter-process).

To support the separation between core and aspectual processes, AO-BPM proposes to represent the crosscutting concern in a specific swimlane to highlight that a crosscutting concern is orthogonal to the core process as well as to make the representation of the proposed crosscutting relationships comprehensible. In order to detail the crosscutting relationship, a quantification mechanism is used. It helps to make the several references to each join point explicit in a single statement. Fig. 1 illustrates the representation applied in the Change Management Process.

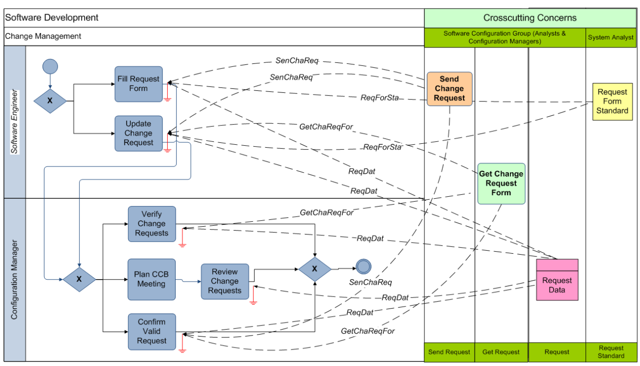


Fig. 1. Aspect-Oriented Change Management Process [1]

AO-BPM applies a symmetric strategy to represent aspects using the same concepts as the base description language. The graphical representation of the join points is a *ground element* near the core process element, allowing the source of the crosscutting relationships to be the crosscutting element and the target to be the ground element that represents the join point.

A *pointcut* language was also specified [1]. It allows stating the different types of join points which appear in a process model, the points where the aspect acts, and the moment this is being applied (before, after, during) at the core description, in a textual format. Basically, it expresses the inclusion of crosscutting concerns in a process. To do so, the *include* primitive is the main clause of the *pointcut* language, used in the advice part to specify the insertion of a crosscutting concern in a core process.

Cappelli et al. [1] argue that AO-BPM modifies the way that a business process is elaborated. In this new approach the process ownership, as well as the aspect ownership, must be established. We may have the following possibilities [16].

(i) To consider just one owner to the whole process (core and aspectual);
(ii) To divide the responsibilities between the process owner and the aspect owner, each one acting in his part with two possibilities:
   a. Each core process has an aspect owner associated
   b. Assign one aspect owner to each type of aspect process

The advantage in *a* is that the aspect owner is directly related with the core process and knows its details, as such he will have basement to request modifications necessary as he understands the aspectual needs of the process. The disadvantage in this case is the difficult to know the impact of modifications as the aspects may act in other processes. In this case, there might be different owners for the same kind of aspect. It requires constant alignment between both aspectual owners and between the aspect owner and the process owner.

The advantage in b is the case when a crosscutting concern, such as the transparency [17] may be incorporated to a process, if required. The set of aspectual elements that compose one concern should belong to the same owner since this concern is only complete when all related elements are considered. The disadvantage is related to the absence of a complete aspectual view and the ignorance of the details of the aspectual process where must act. When the aspectual process is composed of more than one type, there must be an agreement between the owners as it is imperative to have the clear management of the aspectual process.

Considering just one owner for both, core and aspectual processes, the advantage is the existence of only one owner who cares about the improvements of both processes. The disadvantage is the lack of aspectual end-to-end view as aspects are essentially different from process core because they act in different process.

5 An Approach to Represent Process and Aspect Owner

Our research confirms the results of [9], as we did not find any approach to represent the business process ownership and we understand the large benefits of process ownership representation as presented in Section 2. Thus, we propose to represent the process owner using the Strategic Actor [2] proposal because this model explicitly represents actors in the context of intentional modeling. We also include the aspect owner concept in this representation as it refers to a manager responsible for the crosscutting part of the processes. The aspect process has a different characteristic as it crosscuts other processes.

In a business process model, it is possible to represent the role or organizational unit responsible for the execution of the tasks at design level, however there is no way to represent the owner. Considering the ARIS Framework [18] in an EPC, for example, Aris tool provide an attribute to indicate the person responsible for the process as depicted in Fig. 2. There are also attributes to allow customizations to be done and attributes to register the management of changes in the processes. If we consider sub-processes, they also have an attribute for setting the owner but they do not provide a representation of the relationship between the owners involved in business process composition and a representation to clarify the scope of the owner's responsibility. This issue becomes especially important when using the AO-BPM [1] approach. As stated in Section 2, in the models generated with core process separated

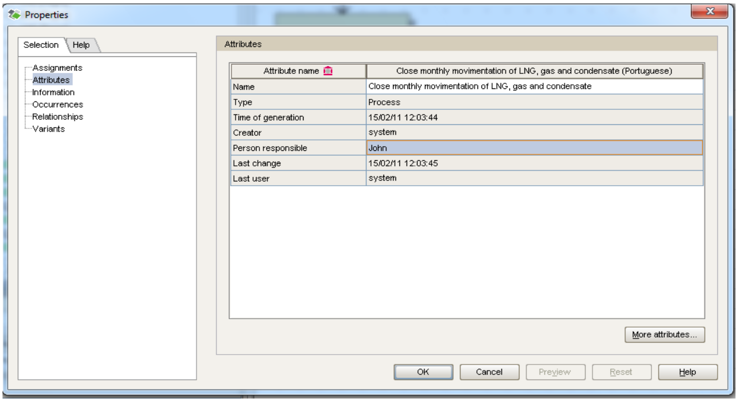


Fig. 2. Process attributes in Aris – process level

from the aspect process, we may have two kinds of ownerships: the aspect owner and the process owner.

Another important distinction regards the business process levels as design and execution. During design, when the process is thought either before it exist (to be) or through reverse engineering (as is), there is no way to represent the process owner. The same holds for the execution level, when the processes are performed on the day-to-day, even if they are not automated. As such, it is important to be able to represent the process and aspect owner and to distinguish between design and execution level.

With the goal to better understand the concept of actor in i* framework, [3] propose the model depicted in Fig. 3. Regarding the entities, the following clarifications apply: *real agent* is an instance of agent, it represents the concept of a specific person or software or hardware; *agent* is an actor with concrete, physical manifestation such as human individual; *role* is an abstract characterization of the behavior of a social actor within some specific context, its characteristics are easily transferable to other social actors; *position* is an intermediation between role and agent, it refers to the position an agent occupy in an organization, also a position covers a role; *actor* is an active entity that carries out actions to achieve goals by exercising his know-how, it is a super class of agent, position, and role.

To facilitate the understanding consider an agent with interviewer role and information engineer position. This distinction can help the mapping of organizational sectors with the modeling of the positions. In a well-structured organization we may find a equivalence between these two categories, so the position is exactly the role to be played. This differentiation provides flexibility to the modeler to distinguish between these two situations if it was the case in the organization.

To attend requirements a and d presented in Section 1, we propose an instantiation of the Strategic Actor Diagram as presented in Fig. 4. An ACTOR is a parent class that can be instantiated in PERSON (agent), OWNER (role) and MANAGER (position). OWNER can be further specified as PROCESSOWNER or ASPECTOWNER. We considered a PERSON as an agent because it indicates the physical manifestation of the ACTOR. OWNER is a specific role that can be covered by a position and characterize the actor’s behavior within the context of ownership, also it can be transferred to other social actors. The POSITION as MANAGER indicates the intermediary characteristic of this class between agent and role.

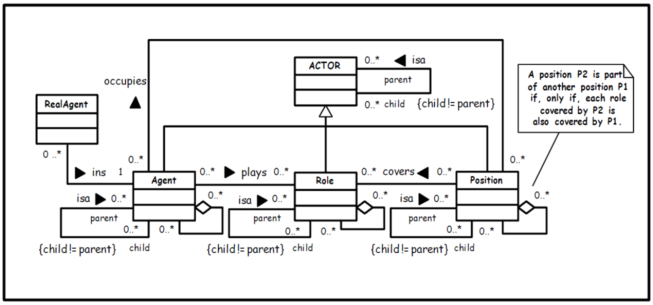


Fig. 3. Strategic Actor Diagram [3]

Regarding requirement a, the OWNER can be defined as an active entity that carries out actions to achieve the goals by exercising its know-how, and besides, it characterizes the behavior of a social actor within some context. In the case of the PROCESSOWNER, his goal is to make the process to be performed as expected and his context is the process he is responsible for. Regarding the d requirement, the crosscutting nature of the ownership responsible for the aspect process is represented by the ASPECTOWNER. His goal is to make the aspectual process act as expected in the core process and his context is the aspect he is responsible for.

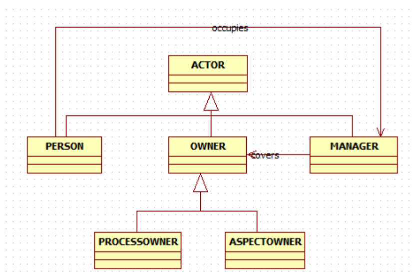


Fig. 4. Strategic Actor instantiation for the Ownership case

To attend requirements b and c we choose to incorporate our elements from Fig. 4 to the EPC meta-model [20]. It is possible to do the same using the BPMN meta-model [19] exploring the lane element but it was not included in this paper due to space reasons. Fig. 5 presents the ARIS EPC [18] meta-model using MOF (Meta-Object Facility) and OMG's meta-metamodel adapted from [20]. In grey are the elements we propose to extend ARIS EPC meta-model with process and aspect owner. The Execution Level and Design Level entities are specializations of the entity EPC. Owner and consequently Process Owner and Aspect Owner are specializations of the Organization Role and the Owner is responsible for an EPC, this relation was incorporated from [6]. The Organizational Unit can be refined by Organizational Role [7] and also refined by a Position as Manager. Person, Manager and Organizational Role are all specializations of Actor.

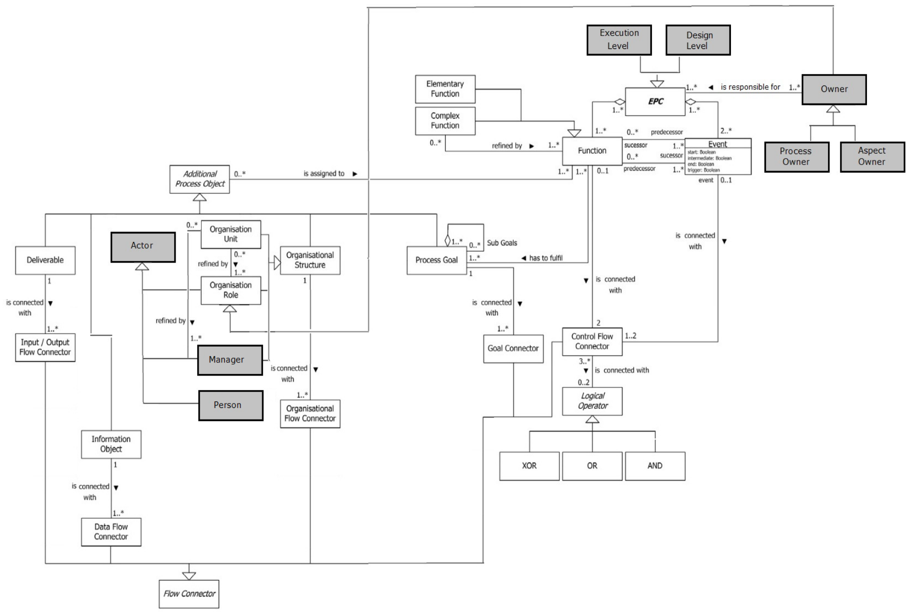


Fig. 5. EPC meta-model extended with process and aspect owner from [20]

5.1 Example

Consider one organization has an organizational structure like the one represented in Fig. 6. A view as depicted in Fig. 7 should be created to generate a representation for the process and aspect owner in the context of the business process Close Monthly Production Aspect-Oriented depicted in Fig. 8.

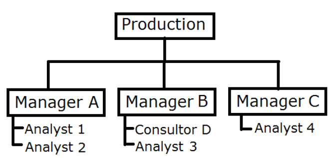


Fig. 6. Organizational Structure of Production in an hypothetical organization

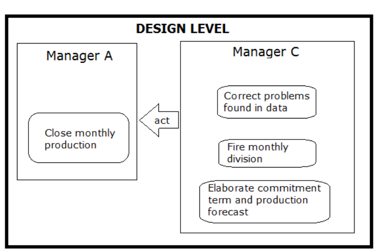


Fig. 7. Responsibility representation for core and aspectual elements of Close Monthly Production (Aspect-Oriented)

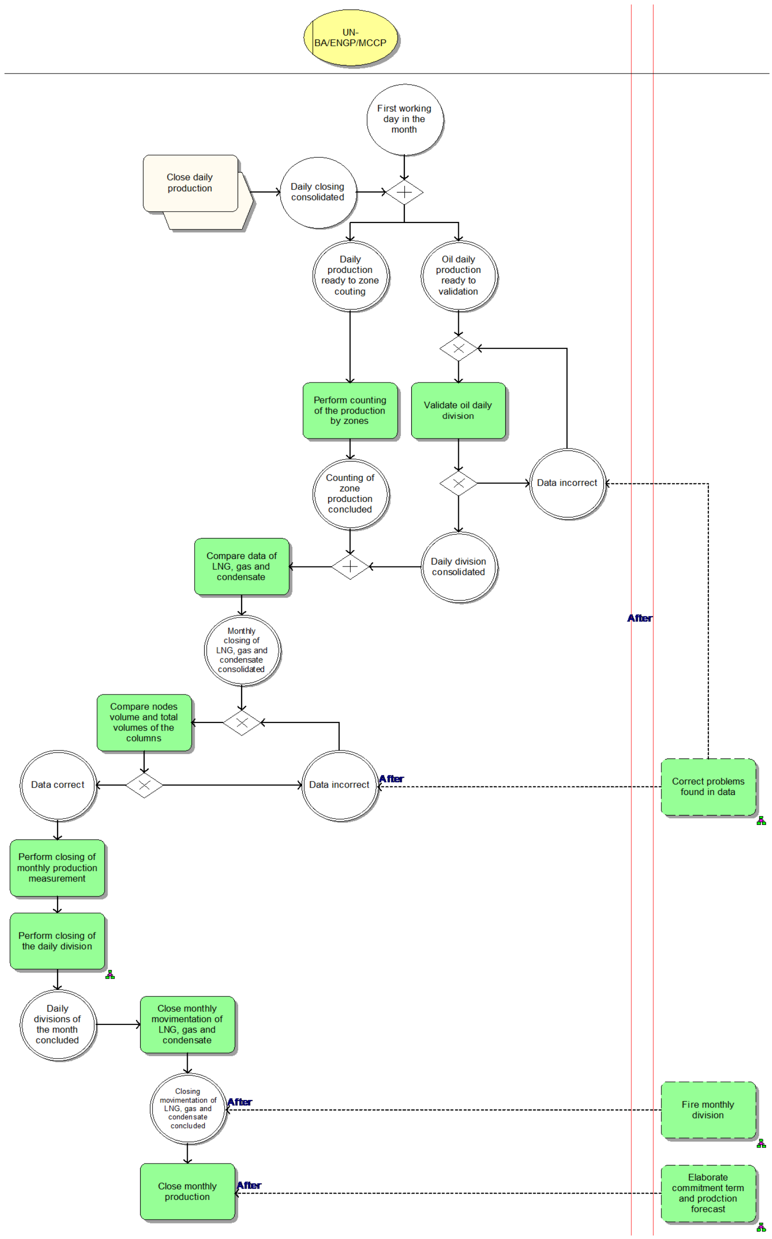


Fig. 8. Close monthly production (aspect-oriented) [22]

The Aspect-Oriented business process *Close Monthly Production* was adapted from an oil and gas organizational where the production of the month must be consolidated with the correct data. In this business process we identified 3 crosscutting concerns: (i) Fire monthly division, (ii) Elaborate commitment term and production forecast and (iii) Correct problems found in data.

The management approach of this organization may consider the options:

1) Assign a process owner and aspect owner being the same position or role (design level) or the same real agent Mary (execution level);
2) Assign a process owner being a real agent Mark or a role or a position and one aspect owner responsible for the 3 crosscutting concerns being another real agent or a role or a position;
3) Assign a process owner being a real agent Josh or a role or a position and three aspect owners, one for each crosscutting concerns, being the real agents or roles or positions.

Regarding the second case, a view of the Aspect-Oriented process *Close Monthly Production* as depicted in Fig. 7 should be created for the organizational structure in Fig. 6.

6 Conclusion and Future Work

In this paper we advocate the importance of clear establishment of the process owner. As we developed our work on AO-BPM, we concluded that it is also important to clear establish the aspect owner. There are some options to establish the aspect owner each one with vantages and disadvantages.

In order to attend the requirements a, b, c, and d, we presented an approach to represent the actor involved in the ownership of process and aspects as an instantiation of Strategic Actor from i*.Thus, it was possible to define the process owner in terms of actor instead of through the activities he performs. It also allowed to define the aspect owner and to include the aspectual process concept in the business process model. To be able to differentiate and represent the design and execution time of a process and integrate our proposal to conventional BPM models, we adapted and incorporated our model to the EPC meta-model.

By means of a hypothetical example we illustrated how to use our proposed approach and presented a model to represent the ownership view regarding a business process. Specifically, business processes composed of core and aspectual elements.

As future work we aim to include our model to the business process generic meta-model presented by [9]. By doing so, we complete the framework with detailed ownership information and enrich business process context perspective.

Regarding the hypothetical example presented in Sub-section 5.1, we should improve it by means of multi-processes as it is the reality in organizations and multiple departments. Another possibility is to use a concrete structural organization.

To facilitate the use of our approach in real organizations, we should develop a tool that allow the representation proposed in this paper - the view composed of aspect and process owner, as well as the elements they are responsible for and the view of the business process as it is modeled and the aspect and process owner. We also have to explore the representation of this second case at execution level.

References

1. Cappelli, C., Santoro, F., Leite, J.C.S.P., Batista, T., Medeiros, A., Romeiro, C.: Reflections on the Modularity of Business Process Models: the Case for Introducing the Aspect-Oriented Paradigm. BPM Journal (2010)

2. Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD Thesis, University of Toronto, Toronto, Canada (1995)
3. Leite, J.C.S.P., Werneck, V., Oliveira, A., Cappelli, C., Cerqueira, A., Cunha, H., Gonzalez-Baixauli, B.: Understanding the Strategic Actor Diagram: an Exercise of Meta Modeling. In: 10th Workshop on Requirements Engineering, Toronto, Canada (2007)
4. Hammer, M.: The Process Audit. Harvard Business Review (April 2007)
5. Jeston, J., Nelis, J.: Business Process Management – Practical Guidelines to Successful Implementations. Elsevier (2006) ISBN-10: 0-7506-6921-7
6. Hengst, M., Hlupic, V., Currie, W.: The Increasing Need for Integrating Simulation and Collaboration to Support Change Management Programs. In: 37th Hawaii International Conference on System Sciences (2004)
7. Kohlbacher, M.: The Perceived Effects of Business Process Management. In: IEEE Toronto International Conference, Toronto (2009)
8. Larsen, M., Klischewski, R.: Process Ownership Challenges in IT-Enabled Transformation of Interorganizational Business Process. In: 37th Annual Hawaii International Conference on System Sciences, Big Island, Hawaii (2004)
9. List, B., Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages. In: 21th Symposium on Applied Computing, Dijon, France (2006)
10. Hewitt, C.: Offices are open systems. ACM Trans. Inf. Syst. 4(3), 271–287 (1986)
11. Hewitt, C., Bishop, P., Steiger, R.: A Universal Modular ACTOR Formalism for Artificial Intelligence. In: International Joint Conferences on Artificial Intelligence (1973)
12. Lamb, R.: Modeling the Social Actor. North American Association for Computational Social and Organizational Systems (NAACSOS), Notre Dame, Indiana (2005)
13. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., e Irwin, J.: Aspect-oriented programming. In: 11th European Conference on Object-Oriented Programming (1997)
14. Filman, R., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley (2005)
15. Silva, L.: An Aspect-Oriented Strategy to Model Requirements. Rio de Janeiro. 220p. PhD Thesis on Software Engineering - PUC-Rio, (2006) (in Portuguese)
16. Santos, F., Cappelli, C., Santoro, F., Leite, J.C.S.P., Vasconcelos, T.: Identificação de Gestores em Processos de Negócio Transversais. In: Workshop de Business Process Management, Salvador, Brazil (2011) (in Portuguese)
17. Cappelli, C., Leite, J.C.S.P., Oliveira, A.P.A.: Exploring Business Process Transparency Concepts. In: RE 2007, pp. 389–390 (2007)
18. Scheer, A.W.: ARIS: Business Process Modeling, 3rd edn. Springer (2000)
19. Owen, M., Raj, J.: BPMN and Business Process Management – Introduction to the New Business Process Modeling Standard. Popkin Software (2003), `http://www.bpmn.org/`
20. OMG Inc., MOF Core Specification v2.0 (2006), `http://www.omg.org`
21. Korherr, B., List, B.: Extending the EPC and BPMN with Business Process Goals and Performance Measures. In: ICEIS (2007)
22. Santos, F., Leite, J.C.S.P., Cappelli, C., Santoro, F., Vasconcelos, T.: Aspect-Oriented Business Process Modeling: Analyzing Open Issues. Business Process Management Journal (to appear)
23. Dietz, J.: Enterprise Ontology – Theory and Methodology. Springer, Heidelberg (2006) ISBN-10 3-540-29169-5

Modeling Security Requirements in Service Based Business Processes

Sameh Hbaieb Turki[1], Farah Bellaaj[1], Anis Charfi[2], and Rafik Bouaziz[1]

[1] University of Sfax, MIRACL Laboratory, Tunisia
{turkisameh,bellaajfarah}@yahoo.fr,
raf.bouaziz@fsegs.rnu.tn
[2] SAP Research CEC, Darmstadt, Germany
firstname.lastname@sap.com

Abstract. Non-functional concerns such as security are essential in business process management and in service based realizations of business processes. Many works and efforts addressed these concerns on the service layer by developing a number of XML-based standards such as WS-Security and other WS-* standards. However, there are non-functional properties that are on the business process layer and need therefore to be specified in business process models. We notice nevertheless that current business process modeling languages lack appropriate means for specifying non-functional properties such as security for example. In this paper, we present a model driven approach for the development of service based business processes which supports both functional and non functional concerns. We also introduce the concept of profiles to BPMN in analogy to UML Profiles. Based on that, we present a BPMN profile to specify security properties in business process models and illustrate its usage through an example.

Keywords: Business Process Modeling, BPMN, Security, Non-functional properties.

1 Introduction

While functional concerns are very well supported in state of the art business process modelling languages such as the Business Process Modeling Notation (BPMN) [1] we observe that these languages do not provide appropriate support for expressing non-functional concerns such as security and quality of service. Non-functional concerns are very important in business process management. In addition, it is also important two distinguish between two levels with respect to non-functional concerns: the process level and the service level [2]. In fact, service level non-functional concerns pertain to the service realizing an activity in the business process. They can be specified and enforced at the service layer using for instance WS-SecurityPolicy [3] and WS-Security [4]. On the other hand, there are those non-functional concerns that are inherently related to the business process and therefore need to be specified on the process level. For example, security properties such as separation of duties [5]

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 76–90, 2012.

or binding of duties [5] need to be expressed on the business process because such properties require the existence of some kind of business process consisting of activities and they allow specifying that certain activities should or should not performed by the same users to avoid the risk of frauds.

However, we notice that most works focus more on the service level. In fact, several standards and specifications were proposed in the last years to address various security related aspects around service oriented architectures. Standards and specifications such as WS-Security [4] and WS-Security Policy [3] fall in this category. These rather technical standards are geared toward the specification and enforcement of security properties on the service level and they cannot be used to specify security properties in business process models.

While service level security is already well-supported we notice that there are only a few works on security in business process models and on non-functional properties in general in such models. A major limitation of most works on security in business process models (e.g. [6], [7], [8], [9], [10]) is that they come with a predefined set of security primitives which cannot be extended by the user of those primitives. There are extensions addressing different security related properties and they cannot be composed with each other as they are not based on a common meta-model. Further, many of these works propose extensions that are specific to security and are not applicable to other concerns such as quality of service for example. There is a need for more generic approaches to specify non-functional concerns in business process models, which can be used for different concerns and not only for security.

In this paper, we address the problems mentioned above by presenting a generic approach to specify non-functional properties in business process models. We also illustrate the usage of that approach in the security context and present an extension to BPMN for security. That security extension can be easily extended by the end users according to their needs. Further the security extension is easily composable with other extensions for expressing other non-functional properties such quality of service or business performance.

Our contributions in this paper are many-fold. First, we present an approach to specify non functional properties in business process models and introduce the concept of profiles based on a meta-model for non-functional concerns. Second, we present a toolset supporting this approach, which consists of a profile editor that can be imported in an enhanced BPMN editor. Third we present a security extension to BPMN, which illustrates our generic approach and allows expressing security properties in business process models. This extension will be applied to the example of a loan approval process.

The remainder of this paper is structured as follows. In Section 2, we use a motivating example to illustrate the limitations of BPMN in expressing non-functional properties in general and security properties in particular. Section 3 presents our solution for modeling non-functional properties and also the tooling that supports this approach. Section 4 presents a security profile that we defined and Section 5 illustrates the usage of the profile. Section 6 reports on related works. Section 7 concludes the paper and presents directions for future works.

2 Motivating Example: The Loan Approval Process

We present in Figure 1 a loan approval process (LAP for short) that we will use as running example throughout the paper. In this example, there are several security properties that need to be expressed such as confidentiality, integrity, privacy, authentication, authorization, etc.

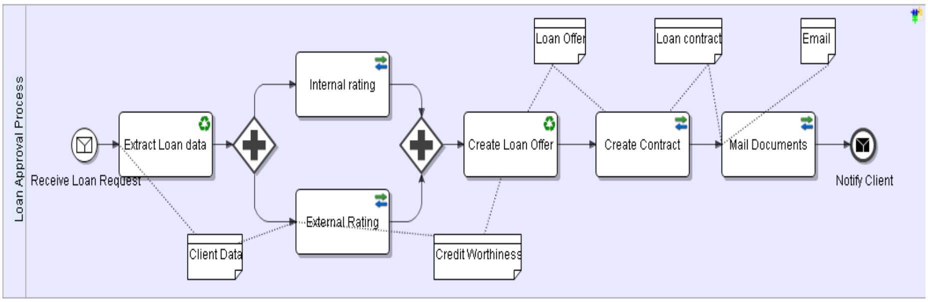


Fig. 1. The Loan approval process (LAP)

The LAP starts when the client asks for a loan from his bank. The bank will then perform in parallel two rating activities: one for the external rating using a credit reporting agency and one for the internal rating to verify the credibility of the client based on the provided documents. After that and upon positive evaluation of the customer creditworthiness an offer is created and subsequently the contract documents are generated using a contracting web service. Finally the offer and the contract are sent to the customer by email using an appropriate mailing web service.

In this business process privacy is essential as sensitive information about the customer is being used. This privacy has to be respected as regulated by laws and also by the internal policies of the bank. Furthermore, the communication with service providers has to be confidential to protect sensitive data from being disclosed by unauthorized parties. In addition, the service providers have to be authenticated. In this example, we have one external service provider (the credit reporting agency) and two internal web services (the contract generation service and the mailing service). Another important requirement in this example is separation of duties as the internal rating has to be done by a different employee than the one who does the external rating. This is necessary to avoid the risk that some employee does both steps and qualifies his friend for example as being creditworthy although that person has a bad credit report. Other security requirements in this example relate to the data objects of the business process such as the client data, the loan offer, the credit report, and the loan contract. Further, one also needs to specify which user roles are allowed to execute which activities and also to access which business objects in this business process model.

Several security properties have been identified so far. Many of these need to be specified in the business process model. However, process modeling languages and BPMN do not provide constructs to express non-functional properties in general and security properties in particular. There is a need to enrich business process modeling languages with generic means to express non-functional properties such as security.

3 Proposed Solution

In this section we first explain the overall context for the work presented in this paper. Then, we present our meta-model for non-functional profiles. After that we present the tooling we developed to allow the definition of non-functional profiles and the usage of these profiles when modeling business processes.

3.1 Context for This Work

The overall context for this work is a model driven approach for the design and implementation of service based business processes. This approach allows defining a platform independent model for service based process and a set of transformation rules that link the model with specific platforms following the Model Driven Architecture (MDA) [11]. On the one hand, such a model-driven approach raises the level of abstraction from low-level technical implementation languages such as WS-BPEL to design models. Therefore it allows the separation of concerns between the business layer and the technical layer. On the other hand this approach provides reusable models for service based business processes. The same model can be used for the implementation of a business process on different target platforms. The approach is illustrated in Figure 2 and it comprises four abstraction layers.

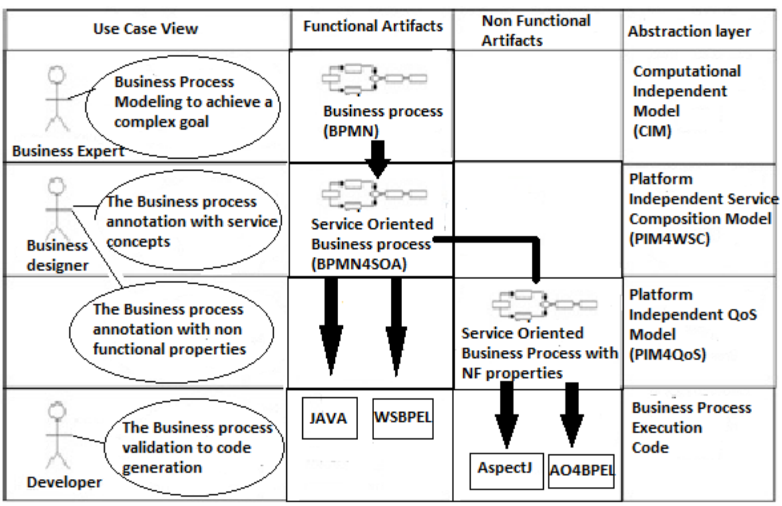


Fig. 2. A model-driven approach for the development of service based business processes

3.2 Meta-Model for Non-functional Profiles

In analogy to the profile concept in the Unified Modeling Language (UML) [12] we propose a profile concept for process modeling languages. In UML a profile consists of a set constraints expressed either in natural language of in a formal language such as OCL. Further, it comprises a set of stereotypes and tagged values. In process modeling languages such as BPMN, we define the profile to consist of a set of constraints expressed in natural language or in OCL and a set of extension definitions, which also may have attributes. Figure 3 shows a meta-model with these concepts:

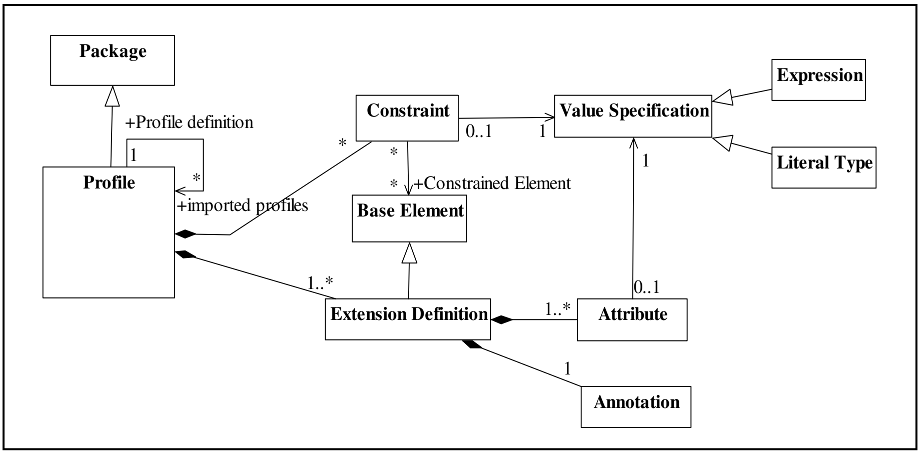


Fig. 3. Meta-model for non-functional profiles

Profile: The profile is an extensibility mechanism to adapt a business process modeling language, such as BPMN, to a specific platform, domain, etc. It can also represent a non-functional concern such as security and business performance. We defined in a previous work a BPMN profile called BPMN4SOA [13] for modeling service based business processes. It is possible to use several profiles at the same time and with the same business process model.

Extension Definition: represents a new property which can be expressed in business processes. Confidentiality and integrity are examples for such extension definition.

Attribute: An extension definition has zero or more attributes and each attribute has a name and a value. For example confidentiality may have an attribute called encryption algorithm and an attribute called encryption level, which indicates whether weak or strong encryption is required.

Constraints: A profile also includes a set of constraints that can be expressed either in natural language or in a formal language such as OCL. Each constraint has a value specification and may be an applicability constraint which specify for instance to which process elements a given property can be applied or a dependency constraint which specifies interdependencies between different concerns. Further types of constraints can also be expressed.

3.3 Application to BPMN

One advantage of BPMN is the capability to express additional information that does not influence directly on the sequence flows or messages flows of the process [1]. This capability is provided through the extensibility artifacts which are Association, Group and TextAnnotation.

For applying our approach to BPMN and defining a BPMN profile for modeling non functional concerns, we introduce a new type of artifact called QoSAnnotation which is an extension of the standard text annotation of BPMN. The QoSAnnotation artifact inherits TextAnnotation attributes and defines the icon attribute as well as other attributes. The attributes depend on the non-functional property represented by the annotation. For example, a confidentiality annotation has two attributes: encryption algorithm and encryption level. Figure 4 shows how we use BPMN extensibility to define the extended Annotation and its attributes.

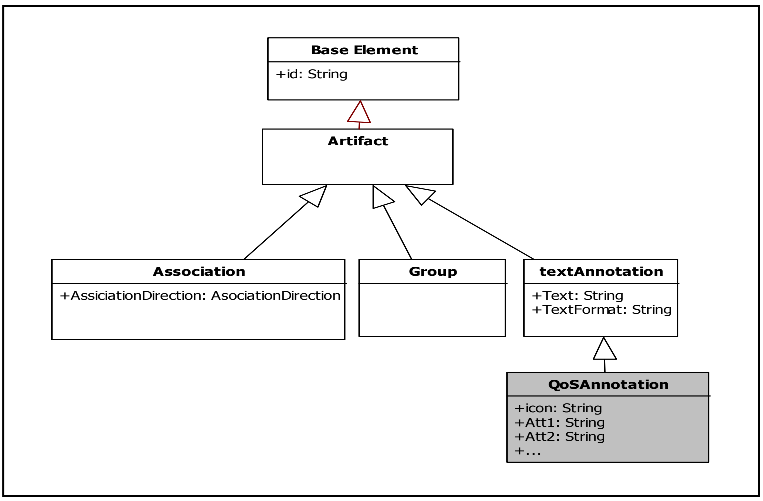


Fig. 4. Definition of the QoSAnnotation artifact according to BPMN extensibility

3.4 Tooling and Implementation

We have implemented a profile editor based on Eclipse Modeling Framework (EMF). The editor allows the definition and manipulation of profiles. In Figure 5, the profile editor is shown, which allows defining a profile for modeling non functional properties and their attributes. The icons can be associated with the properties and will be later displayed on the respective annotations in the business process model. The profile editor allows exporting the defined profiles in an archive file. To build a Profile, one first has to set the profile's name and then define the properties. Each Property is created as a child as of the Profile and can have its name and icon set. Attributes can be defined in a similar way using the appropriate contextual menu.

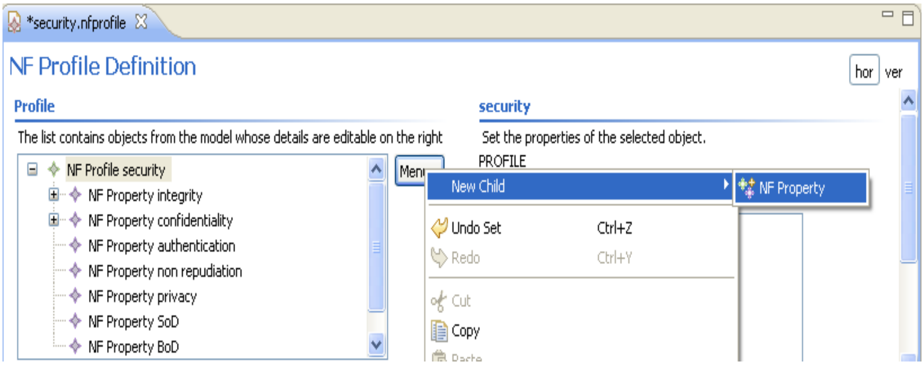


Fig. 5. Screenshot of the profile editor

In addition to the profile editor we also extended the BPMN editor STP BPMN [14] to support importing profiles. After importing one or a set of profiles, the palette of the BPMN editor is extended with new annotations as defined in imported profiles. The annotations of each profile are grouped in a group in the palette. For example Figure 6, shows the palette extension after we import the security and performance profiles. In this case, two groups are added in the palette. For each selected property, the respective attributes can be viewed and edited in the property sheet.

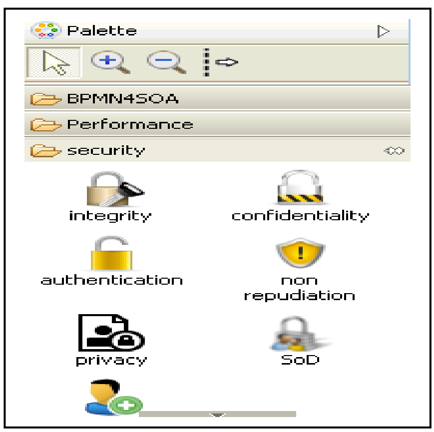


Fig. 6. Palette extension after importing the security and performance profiles

Whilst the profile concept is generic its application to BPMN was based on the annotation concepts and the extensibility mechanisms of BPMN 2.0 [1] which allows extending both the process model and graphical notations. A BPMN extension consists of four different elements: the Extension links an Extension Definition and its attributes to a BPMN model, the Extension Definition defines and groups additional attributes, the Extension Definition Attribute defines new attributes, and the Extension Attribute Value contains the attribute value [1].

4 Security Profile

In this section, we present a BPMN profile for modeling security properties in business processes. This BPMN profile has been specifically defined for providing a lightweight extension to the BPMN elements for expressing security properties. This security profile defines a set of annotations and their attributes, and a set of constraints which are classified in two categories: the application constraints which define the BPMN element to which we can apply the new annotation and the dependency constraints which define a set of dependencies between annotations. In the following, we describe the properties of our profile, their attributes, their annotations and the corresponding constraints.

4.1 Separation of Duties (SoD)

This property expresses that two tasks have to be done by two different users or user roles to avoid the risk of frauds.

Attributes

- **Type:** the value of this attribute is one of the following strings:
  - Static SoD (SSoD) specifies that two mutually exclusive roles must never be assigned to the same user simultaneously.
  - Simple Dynamic SoD (SDSoD) specifies that two mutually exclusive roles must never be activated by a user at the same time.
  - Object-Based SoD (ObjDSoD) specified that a user can activate two exclusive roles at the same time, but he cannot act upon the same object via both roles.
  - Operational SoD (OpDSoD) expresses that a user can activate two exclusive roles at the same time, but cannot have all required authorizations to execute all tasks in a business process.
  - Operational Object-Based SoD (OpObjDSoD) combines operational and object-based SoD; a user can activate two exclusive roles at the same time and can have the authorizations to execute all tasks in a workflow process, as long as these tasks do not act upon the same object [15].
- **GroupID:** This attribute allows defining activities which belong a same SoD constraint. If two annotations are applied to two different activities and they have the same value of GroupID then they express one SOD property.

Notation:

Constraints

- This property can be applied to tasks and activities.
- The SoD and BoD properties are semantically opposed. It is not possible to apply these two annotations to a same element.

4.2 Binding of Duties (BoD)

This property expresses that two activities must be performed by the same user or by the same user role.

Attributes

- **GroupID:** This attribute allows defining activities that belong to a same BoD constraint as explained for SoD.

Notation: 

Constraints

Similar to the SoD constraints.

4.3 Confidentiality

This property expresses that data is confidential and should be only accessible to users with appropriate security credentials. Confidentiality is generally ensured using an encryption algorithm.

Attributes

- **Operation:** This attribute allows defining the type of operation to ensure confidentiality. It can have one the two values encryption or decryption.
- **Encryption level:** The encryption level has a one of the following values:
  - **Middle:** an ordinary encryption algorithm is to be used.
  - **Strong:** an unbreakable encryption algorithm is to be used

Notation: 

Constraints

- This property can be applied to message flows or data objects in BPMN

4.4 Integrity

This property expresses that the data must not modified by a malicious party when transmitted.

Attributes

- **Mechanism:** This attribute specifies the mechanism to implement the integrity property, for example: Hash function, MAC or Digital Signature

Notation:

Constraints

Similar to the confidentiality constraints.

4.5 Authentication

This property expresses that authentication is required for accessing some data object or for performing some activity.

Attributes

- **Authentication method:** The possible values are Username/Password or Certificate.
- **Authentication type:** The possible values are Basic authentication or strong authentication.

Notation:

Constraints

- This property can be applied to pool, to an activity, or to a data object

4.6 Non repudiation

This property expresses that the sender or receiver should be not able to claim not having sent or received some data.

Attributes

- **Type:** the possible values are: sender or receiver. The value sender assures that the transmitter of the message cannot deny having emitted the message in the future. The value receiver assures that the receiver cannot deny having received the message.

Notation:

Constraints

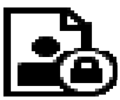
- This property can be applied to message flow

4.7 Privacy

This property expresses that some data is sensitive and has to be kept private.

Attributes

- **Type:** possible values are: *user privacy* (client pool), *service privacy* (partner pool) or *data privacy* (data object).
- **Privacy policy:** possible values are:
  - usage policy (the purposes for which the information collected can be used)
  - storage policy (specifies whether and until when the information collected can be stored by the service)
  - disclosure policy (states if and to whom the information collected from a given user can be revealed)

Notation: 

Constraints

- This property can be applied to a pool and a data object elements.

5 Applying the Security Profile to the Loan Approval Process

In the following, we explain how we used the security profile in the context of the loan approval process presented in Section 2 to address the different requirements. Figure 7 shows a screenshot of the process using the BPMN editor mentioned in Section 3.4 and after applying the security annotations to several elements of the business process model.

Req1: To express that the activities for "internal rating" and "external rating" should be performed by different users we used the annotation for separation of duties.

Req2: To protect the sensitive client data from any modifications we applied the privacy annotation to the data object "Client Data".

Req3: To ensure that the client does not deny having requested a loan. To express this we applied the non-repudiation annotation to message flow separating the LAP and the client pool (not shown in the figure for space limitation).

Req4: To keep confidential the loan contract we applied the confidentiality annotation to the message flow going from the activity "create contract" to the partner pool "contracting service".

Req5: As the partner services involved in this process require appropriate authentication we applied authentication annotations to the respective pools.

It is possible to combine the security profile with another profile such as the temporal profile which we defined to express temporal properties in business process models. To illustrate this we also used annotation from the temporal profile to express that the mailing service has to send the contract within 5 minutes. Also the external rating service should not exceed 3 days before responding to the request.

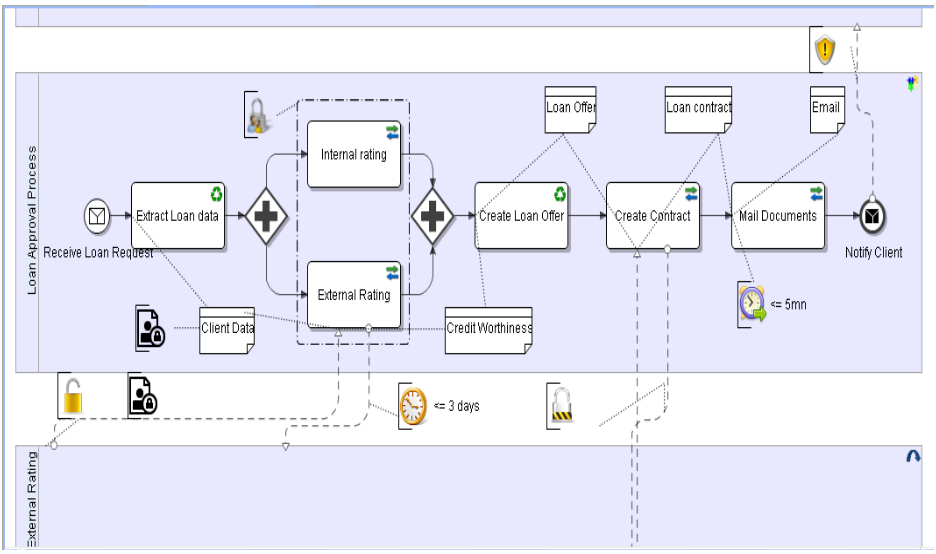


Fig. 7. The loan approval process using security editor

6 Related Work

Several works addressed the specification of security requirements in business process models. Many of them use BPMN and UML activity diagrams for business process modeling. We classify these works in two groups: the first group proposes UML security profiles and the second group uses BPMN extensions to integrate security requirements. Another group of related work focuses on model driven approaches for the security modeling and management.

As an example of related work from the first group, we mention the work presented in [16], in which the authors define a domain specific language to express security requirements in UML based business process models. They define a UML-profile with three types of security stereotypes for integrity, confidentiality and availability, respectively. This work has two main limitations. First, it supports only three security properties and cannot express others requirements such as privacy, non-repudiation, and separation of duties. Second, it is not clear when and to which process elements these security goals can be applied. In addition, the defined stereotypes do not allow providing parameters such as the attributes in our approach and therefore the specification remains at a very high level. A second work in this first group is presented in [17], which defines a UML profile for modeling secure business processes based on activity diagrams from the perspective of business analyst. This work allows expressing security properties in a platform independent manner according to the MDA approach. However, this work and the one presented in [16] have a limited set of security properties that can be expressed. In addition, these sets are not extensible, unlike in our approach. Further, both works do not support expressing constraints such as the applicability constraints in our approach. In [18] the authors propose *SecureUML*, a modeling language designed to integrate information relevant to access control into application models defined with UML. This extension combines graphical notations and logical constraints to express security requirements. In contrast to our work, this approach concentrates on access control requirements and does not cover the other security aspects.

The second group of related works proposes BPMN extensions for security. For example, the authors of [6] define a language for defining security constraints in business process models. They used the standard BPMN TextAnnotation to add security information to the process model. The main limitation of this work is that, using text annotation makes the business process model full of text and difficult to understand. Further, the text annotation does not provide attributes for expressing further parameters of the security property, unlike in our approach which uses the QoSAnnotation. In the same context, the work presented in [5] extends BPMN to express authorization constraints such as role-task assignments, role hierarchies, separation of duty and binding of duty constraints for manual tasks within BPMN. The major advantages of our approach over the one presented in [6] and [5] are extensibility and genericity. Our approach easily supports further security properties. The user just has to extend the profile definition. This is not feasible in the approach presented in [6] and [5]. Further, our approach works for various non-functional properties and not only for security. Different profiles for example for security and quality of service can be used at the same time.

The third group of related work focuses on model-driven approaches for the security modeling and management. In [19], the authors present an approach to describe security requirements at the business process layer and they also cover the translation of these requirements to concrete security configurations for service-based systems. In [20], the authors presented an approach to deal with security requirements of service composition at various levels of abstraction: from business process specification, to composite service design and development, and to business process execution. Both works cover the specification and realization of security properties. In our work, we also cover the realization of security properties but this was not presented in this paper for space limitations. In fact, we generate aspects in AO4BPEL and in AspectJ to enforce the properties expressed via annotations. The functional business processes are transformed to BPEL processes or Java process implementations as shown in [21]. Te major benefits of our approach over works such as those presented in [19] and [20] are genericity (not only applicable to security) and modularity as the functional code and the non-functional code are separated.

7 Conclusion

In this paper, we have presented a generic approach for specifying non-functional properties in business process models. The approach is based on the notion of a non-functional profile, which is inspired by UML profiles. We also presented a toolset supporting this approach, which consists of a profile editor and an extended business process editor that can import the profiles and enrich its palette according to the properties defined in the profile. We applied the approach and the tooling to security and defined a BPMN security profile that we used to specify several security related properties in a business process model for loan approval processing. Our future work will focus on defining a quality of service profile and validating it.

Acknowledgments. The third author of this paper was partly supported by the BMBF Software-Cluster project EMERGENT (grant no. 01IC10S01.)

We thank Dani Colomer for his contribution to the implementation of the profile editor and the profile importer.

References

1. Object Management Group.: Business Process Modeling Notation (BPMN) 2.0, http://www.omg.org/spec/BPMN/2.0
2. Charfi, A., Schmeling, B., Heizenreder, A., Mezini, M.: Reliable, Secure and Transacted Web Service Composition with AO4BPEL. In: 4th IEEE European Conference on Web Services (ECOWS), pp. 23–34. IEEE Computer Society (2006)
3. Chris, K., Anthony, N.: Web Services Security Policy Language (WS-SecurityPolicy) Version 1.1 (July 2005), http://www-128.ibm.com/developerworks/library/ws-secpol/
4. OASIS. Web Services Security: SOAP Message Security 1.0 (2004), http://docs.oasis-open.org/wss/2004/01/
5. Wolter, C., Schaad, A.: Modeling of Task-Based Authorization Constraints in BPMN. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 64–79. Springer, Heidelberg (2007)
6. Mulle, J., Von Stackelberg, S., Bohm, K.: A Security Language for BPMN Process Models. Karlsruhe Reports in Informatics, KIT, pp. 2190 – 4782 (2011)
7. Rodriguez, A., Piattini, E.F.-M.M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. J. IEICE - Transactions on Information and Systems E90-D(4), 745–752 (2007)
8. Rodríguez, A., Fernández-Medina, E., Piattini, M.: Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 51–61. Springer, Heidelberg (2006)
9. Wolter, C., Menzel, M., Meinel, C.: Modelling Security Goals in Business Processes. In: Proc. GI Modellierung. LNI, vol. 127, pp. 197–212. GI (2008)
10. Hafner, M., Berthold Agreiter, R.B.: SECTET: an extensible framework for the realization of secure inter-organizational workflows. Internet Research 16(5), 491–506 (2006)
11. OMG: MDA Guide Version 1.0.1 (2003), http://www.omg.org/mda/

12. OMG: Unified Modeling Language: Superstructure version 2.0 UML/2.0/ (2005), http://www.omg.org/spec/
13. Charfi, A., Turki, S.H., Chaâbane, A., Bouaziz, R.: A model-driven approach to developing web service compositions based on BPMN4SOA. J. Reasoning-based Intelligent Systems 3(3/4) (2011)
14. SOA Tools Plattform Project, http://www.eclipse.org/stp/
15. Kallel, S., Charfi, A., Mezini, M., Jmaiel, M., Klose, K.: From Formal Access Control Policies to Runtime Enforcement Aspects. In: Massacci, F., Redwine Jr., S.T., Zannone, N. (eds.) ESSoS 2009. LNCS, vol. 5429, pp. 16–31. Springer, Heidelberg (2009)
16. Saleem, M.Q., Jaafar, J., Hassan, M.F.: Model-based Security Engineering of SOA System Using Security Intent DSL. J. New Computer Architectures and Their Applications (IJNCAA), The Society of Digital Information and Wireless Communications 1(3), 565–580 (2011) ISSN: 2220-9085
17. Rodríguez, A., Fernández-Medina, E., Piattini, M.: Security Requirement with a UML 2.0 Profile. In: First International Conference on Availability, Reliability and Security, p. 8. IEEE Computer Society (2006)
18. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
19. Menzel, M., Thomas, I., Meinel, C.: Security Requirements Specification in Service-Oriented Business Process Management. In: 7th IEEE International Conference on Availability, Reliability and Security, pp. 41–48. IEEE Xplore, Prague (2009)
20. Souza, A.R.R., Silva, B.L.B., Lins, F.A.A., Damasceno, J.C., Rosa, N.S., Maciel, P.R.M., Medeiros, R.W.A., Stephenson, B., Motahari-Nezhad, H.R., Li, J., Northfleet, C.: Incorporating Security Requirements into Service Composition: From Modelling to Execution. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 373–388. Springer, Heidelberg (2009)
21. Chaâbane, A., Turki, S.H., Charfi, A., Bouaziz, R.: From Platform Independent Service Composition Models in BPMN4SOA to Executable Service Compositions. In: 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2010), pp. 653–656 (2010)

A Conceptual Foundation of Requirements Engineering for Business Information Systems

Sebastian Adam, Norman Riegel, Anne Gross, Oezguer Uenalan, and Simon Darting

Fraunhofer IESE, Fraunhofer Platz 1, 67663 Kaiserslautern
{sebastian.adam,norman.riegel,anne.gross,oezguer.uenalan,
simon.darting}@iese.fraunhofer.de

Abstract. Requirements engineering (RE) is a crucial discipline when developing software systems. Applying RE activities successfully in the domain of business information systems (BIS) requires a deep and common understanding on how concepts of RE and business analysis are related. We consider this fact as being a challenge as currently no commonly accepted RE process exits that bridges the gap between these two disciplines. This results in unclear mappings and finally makes it difficult to align methods that exist in both areas. To tackle this challenge, we propose a reference issue model that aims to capture definitions and relations of the issues that are typically relevant in BIS development. In this context, we describe our followed research approach, an underlying meta-model as well as an exemplary instantiation and usage of the reference issue model. This contribution shall serve as a foundation for the integration of RE and business analysis as well as for the development of corresponding analysis approaches.

1 Motivation

Requirements engineering (RE) is a discipline that is indispensable for assuring the success of a development project right from its beginning [1]. When developing business information systems (BIS), requirements must therefore be handled systematically in order to avoid costly problems and rework in later phases. We define a BIS as a software system that "integrates and streamlines business processes across organizational and geographical borders" [2], and that automates the input, storage, transmission, and transformation of business data.

However, during the early conception of BIS, bridging the gap between the business analysis (focusing on business processes, interactions and structures), and actual RE (mainly focusing on the supporting software systems) is not always easy. The different terminology used in these two areas leads to misunderstandings and makes it difficult to continuously and seamlessly apply methods for the elicitation, analysis, validation and management of requirements. In particular, even though it has been recognized that business and software must be closely aligned [3], many existing RE approaches tend to neglect the integration with business analysis and rather focus on the software side. Thus, when both disciplines are needed in a project, it depends

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 91–106, 2012.

on the individual skills of an engineer to close this gap. Approaches used in the business process management community, such as [4], also share this basic problem. In particular, many of them tend to neglect issues beyond the scope of the actual business process, even though the process is often addressed exhaustively both from a business perspective and from a technical perspective. Thus, even if traceability has been recognized as an important concept [5], traces to influencing or influenced elements such as goals, regulations, implementations, etc. can therefore not be maintained explicitly. This results in the problem that early impact analysis of upcoming changes becomes hard. Furthermore, it is widely acknowledged that a sound engineering approach always requires a stable foundation in order to explain how to proceed with an appropriate method [6]. Since a common foundation is missing, existing approaches in RE typically do not fit together well when one tries to integrate them into value-added approaches. To enable the creation of holistic methods that address both the business and the system view in an integrated manner, a solid foundation is therefore needed that can address the problems stated above.

However, as existing work is not sufficient to provide a holistic foundation for RE in BIS, the overall aim of this paper is to introduce a reference issue model that consolidates different best practices[1]. The purpose of this model is to define and clarify the relationships between typical and well established issues that are relevant in BIS development projects, and to make this (typically tacit) knowledge explicit. We define an *Issue* here as an “inherent element that is either part of a system or part of the system’s environment” (see Section 3). By using the reference issue model, we are then able to address the business side as well as the IT side in an integrated manner. This helps clarifying the traceability between both layers and therefore closes the gap that often exists between both worlds (see Figure 1). Thus, a RE method that is based on the reference issue model is then able to provide a seamless procedure on how to elicit, analyze and refine requirements based on a given business context. In this regard, we explicitly distinguish the concept of a requirement from the real world element (i.e., the issue) it concerns.

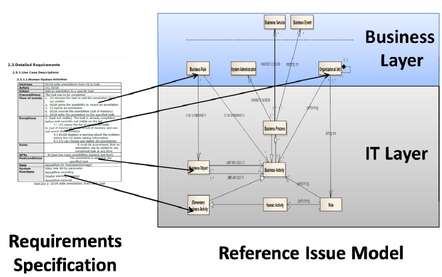


Fig. 1. Requirements Traceability based on the Reference Issue Model

[1] The work presented in this paper was performed in the context of the Software-Cluster project (`www.software-cluster.org`). It was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant no. "01IC10S01". The authors assume responsibility for the content.

The remainder of this paper is structured as follows: In Section 2, we describe the research approach that was followed to develop the model. In Section 3, a meta-model is introduced that clarifies basic RE related terms. Based on this, a reference issue model for RE in BIS is described in Section 4. Section 5 deals with the instantiation and usage of the model, while Section 6 provides an overview of related work. The paper closes with a conclusion and outlook on future work in Section 7.

2 Research Approach

As a first step of our research approach (see Figure 2), we consolidated existing work in the area of RE for BIS. For this purpose, we analyzed prominent specification standards like IEEE 830 [7], IEEE 1233 [8], IEEE 1362 [9], V-Model XT [10], and the Volere Template [5] and extracted all issues included in them. Furthermore, several industrial requirements specifications as well as the following methodological frameworks were also considered in order to enhance and challenge the initial set of issues: ARIS [11], Zachman [12], Rational Unified Process [13], TORE [14], and Fraunhofer IESE's SOE Model [15]. The reason for choosing exactly these frameworks was that they are prominent and aim at conceptually aligning business and IT explicitly.

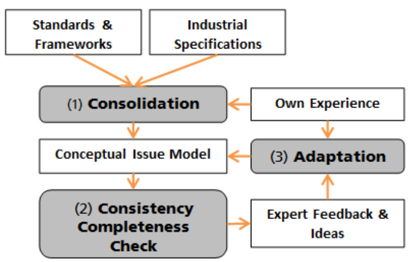


Fig. 2. Research Approach for the Development of the Model

In the more than 190 (partially redundant) issues obtained from these resources, synonyms and redundancies were sorted out by using the open card sorting technique [16]. Furthermore, issues that were not expected to be important for specifying a BIS or that were already part of the meta-model (see Figure 3) were removed. Thus, we deleted motivations, background, justifications, goals, visions, expected impacts and effects, project-related issues, support and accompanying services, or aspects of the solution space. The reason was that these "issues" are either not issues in the sense of our definition (see Section 3) or that they are not needed to elaborate the required capabilities of a BIS. Based on additional RE literature, as well as on our own experience gathered during elicitation in industry, relationships between the 40 remaining issues were then determined. For being able to detect all existing relationships, we considered each pair of issues and checked whether there are relationships between them. Finally, we assigned the issues to four views according to the circles of Alexander's onion model [17]. As a result, a first version of the reference issue model was developed.

In several subsequent iterations, this model was then checked for completeness and consistency. This was done in two different ways. On the one hand, the model was discussed with three experts in order to take their knowledge and experience also into consideration. Their feedback and ideas were carefully analyzed and incorporated into a new version of the model. On the other hand, completeness and consistency were checked by assessing concrete elicitation instructions that were systematically derived from the reference issue model using the template described in [18]. We chose this indirect validation, because it was easier for us to check whether the issues and relationships reflected in the instructions actually covered a realistic elicitation process than when just reviewing the underlying abstract model. Thus, besides direct feedback on the model, ideas that emerged during the assessment of the derived elicitation instructions were also incorporated into an adapted version of the reference issue model. The research here can thus be considered as an iterative approach, in which the model was built in short iteration cycles and applied directly to show its usefulness and applicability.

For the near future, we plan to iterate the model further with external experts from the RE and BPM community in order to get broader commitment of the ideas.

3 Basic RE Concepts and Their Relationships

In each software development project, requirements form the basis for communication, contracting, development, integration, maintenance, and "are the things you should discover before starting to build your product" [6]. Even if there is no universal definition of requirements [20], "requirements (basically) express the needs and constraints placed on a product that contributes to the solution of some real world problem" [21]. Hence, requirements are descriptions of what a "product must do or a quality that a product must have" [6] in order to achieve a certain goal. However, requirements do not exist in isolation, but are strongly related with other concepts. Thus, the reference issue model described in this paper (see Section 4) cannot be defined without providing an underlying meta-model explaining the meaning and relationships of basic RE terms. For this purpose, we have defined an RE meta-model (illustrated in Figure 3) that captures basic RE terms such as *Goal, Requirement, Stakeholder, Artifact*, etc. and their relations.

In the following, the concepts captured in the RE meta-model are described in more detail. Even though the aforementioned explanation of the term *Requirement* may probably fit the reader's implicit understanding of what a requirement is, we would like to provide a more precise definition of this concept. Thus, a *Requirement* can be considered as "(1) information about a characteristic or capability a system must have, or (2) information about a characteristic of the usage environment a system must consider to satisfy a stakeholder goal". Examples include "The system should provide a print function", or "The system should support mobile end users". While the description of a system's capability is usually indeed more precise, the description of environmental objects that must be supported is also widely used and typically helpful.

The concept of a *System* in this context is considered as “a set of components interacting with each other to satisfy some global objectives” [22]. Hence, an entire organization can also be a system for which requirements may exist. The purpose of each requirement is therefore the specification of any kind of system from an external point of view, i.e., from the perspective of a stakeholder who has certain goals that should be satisfied by the system. While a *Stakeholder* is “a person or organization who will be affected by the system or who has a direct or indirect influence on the system’s requirements” [23], we define a *Goal* as “a target state in the future that is worthwhile to be achieved or kept and whose satisfaction requires the cooperation of the system and its environment”. Therefore, goals are not explicitly considered as requirements in this paper; they rather define a state in the future that is to be reached / fulfilled when implementing requirements.

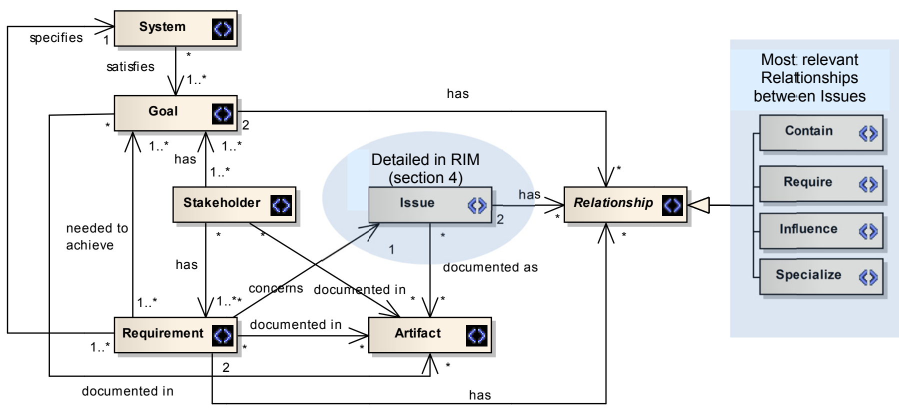


Fig. 3. Basic RE Concepts on a Meta Level (RIM = Reference Issue Model)

Regarding its content, a requirement is always concerned with an *Issue*, respectively an issue instance. We define an *Issue* as “an inherent element that is either part of a system or part of the system’s environment”. Issues therefore cover both functional and non-functional system aspects (e.g., system functions), as well as elements of the usage environment for which a system must be designed in order to provide appropriate support (e.g., user roles, workplaces, interaction data, business processes, etc.). The purpose of the reference issue model introduced in Section 4 is therefore to define and clarify the typical issues that are relevant for discussions in the context of BIS projects.

However, as an issue is just a class, each requirement does not provide information about the issue itself, but about a concrete instance that is either to be realized or supported by a system. An example of a requirement concerned with the issue “human system activity” could be: “The system must support the purchase of a ticket that should include the following steps: […]”. Hence, a requirement determines the issue instances that have to be supported or implemented by the system; in this example the concrete human system activity “Purchase Ticket”.

As issues classify conceptual elements of the real world, there exist *Relationships* between them. While there also exist several relationships between goals and requirements (e.g., conflicts), we just identified those related to issues, as issues are in the focus of this paper. In our current model, we distinguish four relations here: *influence*, *contain*, *require,* and *specialize*. That is, an instance of an issue may be influenced, contained, required, or specialized by an instance of another issue. As an example, a certain organizational unit may be *contained* in another organizational unit while being *required* by a certain business process in order to execute this process.

Typically, stakeholders, goals, requirements, and issues often do not remain in tacit knowledge, but are documented explicitly in a project. Thus, all of them can be described with tangible *Artifacts* that we consider as important concepts. We define an *Artifact* as “an object produced or shaped by human conception or agency [24] in order to externalize the information about the aforementioned concepts”. For instance, the issue “human system activity” is often described by means of textual use case descriptions or UML activity diagrams.

In order to address the challenge of a missing best-practice foundation, a reference issue model for BIS is proposed in the subsequent section. However, similar to other reference models, this reference issue model also does not claim to be perfectly applicable in each concrete setting. Indeed, it addresses typical settings when a BIS is introduced, but needs to be tailored to the specific development needs as addressed in our ongoing research (see [25]).

From a theoretical point of view, the reference issue model bridges the gap between the stereotype “issue” introduced by the RE meta-model (see Figure 3) and concrete real world objects (e.g., “business travel process”) with which a specific project may deal. Thus, the reference issue model contains a concrete set of issues for which requirements typically have to be elicited (and described) in the business process domain.

The oval in Figure 4 visualizes the position of issues within the MOF stack [26] on the example of the issues “Business Process” and “Business Object”. Both business process and business object are here a class of the stereotype “Issue”, while “Business Stakeholder”, “Business Requirements”, “Business Specification”, “BPMN Model”, and “UML Class Model” instantiate other stereotypes of the meta-model.

For each issue of the reference issue model, concrete instances exist in the real world with which requirements can be concerned. In the given example, a “Business Travel Process” is an instance of the issue “Business Process” on which a “Business Stakeholder (Institute Director)” may state the requirement that a certain BIS should implement this concrete business process. While this requirement may be documented in a concrete “Business Specification” used for development, the “Business Travel Process” may be documented in a “Travel Process Model” based on BPMN, as it can exist independently of system requirements concerned with it.

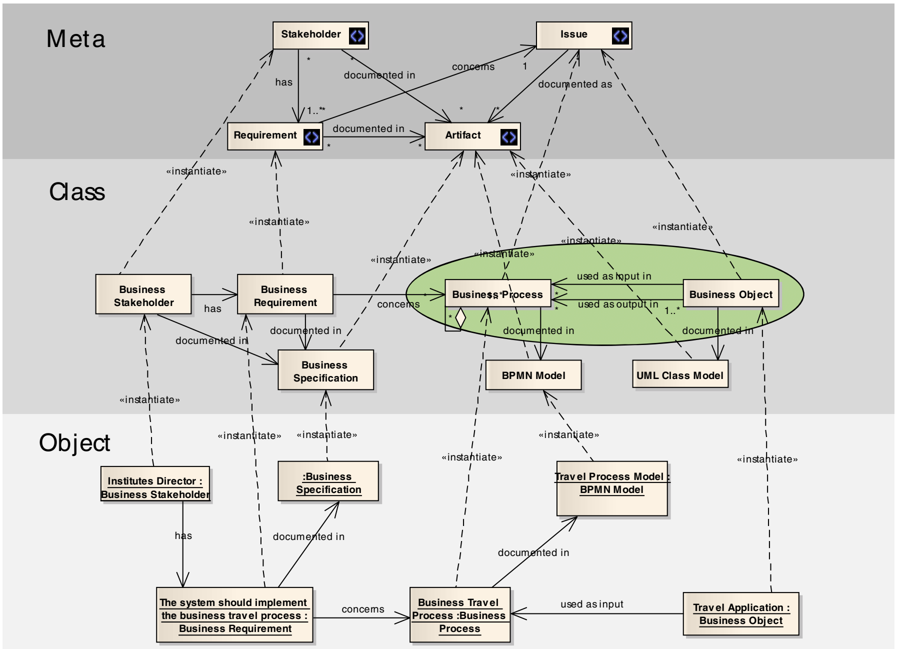


Fig. 4. Example to Illustrate Issues (oval) within the MOF Stack

4 Reference Issue Model

This section introduces the reference issue model that further details the *Issue* element identified in the RE meta-model (see Section 3, Figure 3). This reference issue model defines and clarifies the relationships between typical issues that are relevant in the context of BIS development projects and is described incrementally according to the circles of Alexander's onion model (Wider Environment, Containing System, System, Kit) [17]. However, the views according to which we introduce the model are just simplifications, as they do not show all relationships that may exist between issues of different views (in particular, we abstract from influence-relationships between issues in different views).

4.1 The Business Environment View

The Business Environment View (see Figure 5) basically reflects the "wider environment" of the onion model, and thus contains those issues necessary for scoping the actual business context to be supported by a BIS.

The anchor of this view is a *Project* in which a certain application should be developed or introduced. A *Project* is "a planned and managed action to solve one or more problems". In such a project, at least one *Business Area* is considered (e.g., travel management) for which this application may be relevant. We define a *Business Area* as "a logical part of an organization responsible for a certain market segment, respectively for a certain kind of services and goods, or locations, domains, etc."

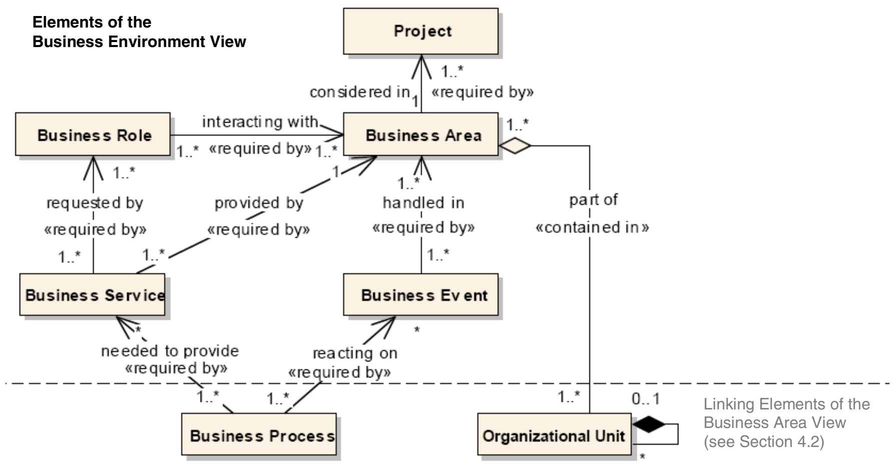


Fig. 5. Business Environment View

Business areas have in common that they provide different *Business Services* (e.g., travel booking, travel accounting) requested by external *Business Roles* (e.g., sales employees) with which the business area interacts. Thus, a *Business Service* is “a useful labor performed to produce value for a third party”. Furthermore, the business area may handle (additional) *Business Events* (e.g., requests for reporting) and may also consider given *Regulations* from outside. In this regard, a *Business Event* is “an external trigger that requires the business area to react”, while a *Regulation* is “a law or standard that can have an impact on the organization and work of a business area”.

4.2 The Business Area View

The Business Area View (see Figure 6) reflects the ”containing system” of the onion model, and describes the issues necessary for defining the actual business context to be supported by a BIS. However, the Business Area View just describes the broader organizational context of an information system, but not the actual environment in which this system will be operated and used.

The anchors of this view are all important issues that require the business area to do something internally: the Business Services (e.g., travel booking) provided by the business area, and the Business Events to be handled in the business area (e.g., request for reporting). However, besides the pure business-related triggers, the introduction of the planned BIS itself also requires internal reactions. This holds especially true for applications that require certain governance or administration processes for their management and evolution. Thus, the planned *System Administration* is also considered as an anchor here, which we define as “the whole set of tasks to be done for administering the users, assets, and data of an information system”.

For each of these anchor issues, at least one *Business Process* shall exist that is needed internally for reacting (e.g., travel application process, monthly report generation process, etc.). A *Business Process* is “a specific ordering of *Business Activities* across time, people, and places, with a beginning, an end, and clearly

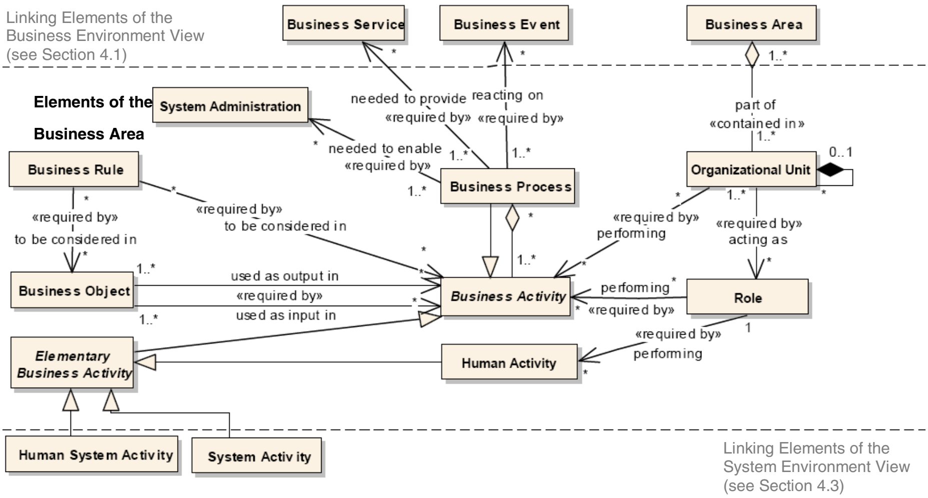


Fig. 6. Business Area View

identified inputs and outputs". As a business process is a specific type of business activity, each business process can either be decomposed recursively into further (sub-) business processes (e.g., travel booking) or just comprise *Elementary Business Activities* (e.g., approve travel application) depending on its level of abstraction. In this regard, an *Elementary Business Activity* is "an atomic step in a business process that results in a valuable, stable state and that is performed by a single role or system". Business activities can be performed either by *Roles* (e.g., project managers) or *Organizational Units* (e.g., sales departments), where Organizational Units can also be decomposed recursively. While an *Organizational Unit* is "a structural part of an organization responsible for a certain area of tasks and topics", a *Role* is "a group of human persons based on a logical set of their responsibilities, rights, and tasks".

Furthermore, business activities use *Business Objects* (e.g., travel applications, tickets, etc.) as input and output, while considering *Business Rules* that may govern and control their execution. While a *Business Object* is "an entity that is handled in or affected by business processes", a *Business Rule* is "a rule that guides the behavior of an organization in order to operationalize its business strategy". Business rules can either be facts, restrictions (rights and duties), enablers (conditional actions), conclusions (conditional facts), or (conditional) calculations. [20]. Specific kinds of elementary business activities are *Human Activities*, which "are performed by a role without any system support". Even though these activities are typically not of interest for application development, they may help to identify novel automation possibilities.

4.3 The System Environment View

The System Environment View (see Figure 7) reflects the "system" in the onion model and therefore describes the immediate context in which a BIS is operated and used, both from a technical and from a work perspective. It therefore clarifies the issues that are needed for describing an application's interactions with its outside.

With regard to the business area view, this view introduces two further sub-types of the elementary business activity, namely *System Activities* (e.g., auto-reply to incoming email) and *Human System Activities* (e.g., book hotel). While *System Activities* “are performed by the planned application without any human intervention”, *Human System Activities* “are performed by *User Roles* from different *Workplaces* (e.g., office) by using a system via certain *UI areas* the application provides (e.g., travel application form)”. In this regard, we define a *User Role* as “a role that interacts with an application”, while the *Workplace* is “a location from where a user role works with the application”. A *UI area* is “the logical part of a system’s user interface that allows users to interact with the system in order to carry out certain human system activities”.

However, besides user roles, *Partner Systems,* i.e., external systems already available or to be introduced in a parallel project (e.g., SAP), can also interact with an application depending on the application’s *Operation Mode*. An *Operation Mode* is “a specific state of a system in which a certain (sub)set of capabilities (system functions, quality characteristics) is available (e.g., normal mode, recovery mode, maintenance mode, offline mode)”.

Each partner system performs one or more *System-System Interactions* with the application to be introduced, i.e., “an interaction sequence, in order to automatically exchange data (e.g., synchronization of employee data)”. This involves the *System Interfaces* of the process application as well as the *Partner System Interfaces* of its partner systems. In this context, an *Interface* is understood as “an endpoint provided by a system through which another system can interact”. Via these interfaces as well as the UI areas, *Interaction Data*, which are parts of business objects, are then exchanged (e.g., certain employee data).

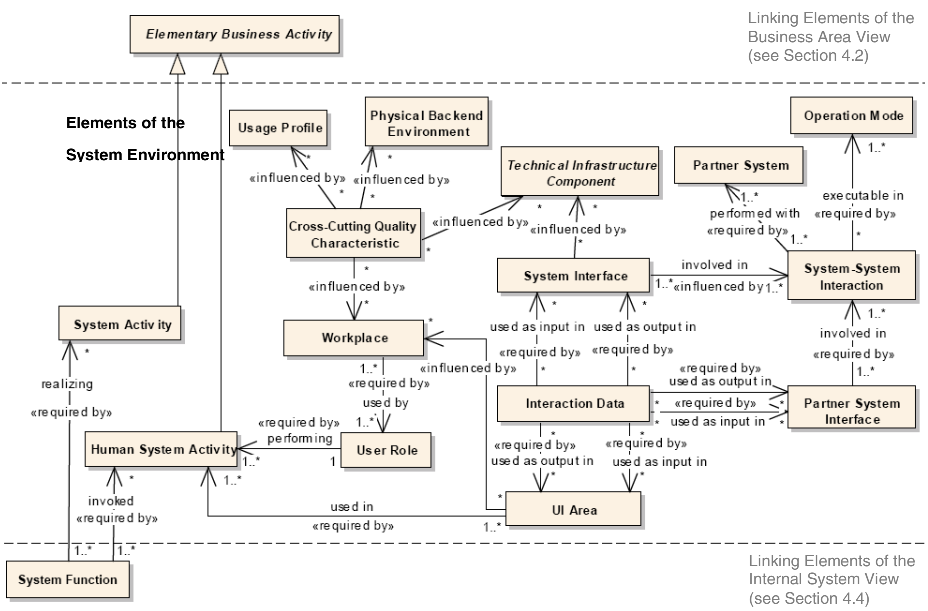


Fig. 7. System Environment View

However, as the system environment does not only have a functional component, non-functional issues also have to be considered here. In this regard, the *Cross-Cutting Quality Characteristics* of the system, i.e., the non-functional properties that affect a system as a whole (e.g., security, reliability, usability, etc.) are important. These characteristics are influenced by the *Technical Infrastructure Components,* which are already given in the usage environment (e.g., existing server hardware, etc.), the *Physical Backend Environment*, (e.g., climate or risk of natural disasters, etc.), the intended *Usage Profile* (e.g., 10.000 users between 9 a.m. and 5 p.m.), and the *Workplaces* from which the system will be invoked. While a *Technical Infrastructure Component* is "the underlying technology (e.g., hardware, operation system, middleware, network, etc.) whose services are used by a system to run", the *Physical Environment* is "the natural environment in which a system's components are deployed". A *Usage Profile* is finally "a quantitative description of how a system will be used".

4.4 The Internal System View

The Internal System View (see Figure 8) reflects the "kit" in the onion model and thus the issues to be considered or designed during the actual development of a system. The Internal System View addresses the internal parts of a system, however, only to the extent to which these parts are already decided during RE. Thus, we abstract from concrete system components and rather deal with logical issues such as an application's functionality or the policies that are defined with regard to its implementation.

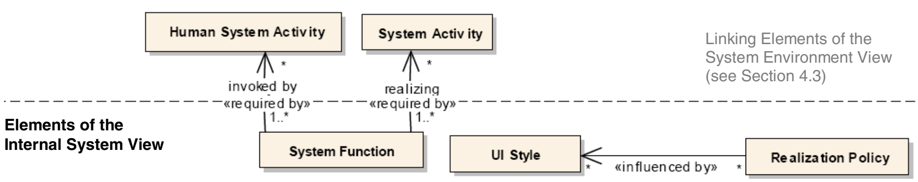


Fig. 8. Internal System View

The anchors of this view are the Human System Activities and System Activities introduced before. For both activities, *System Functions* must exist that realize (parts) of these activities. A *System Function* is "a reaction (i.e., state change or response) of a system that is triggered by an external stimulus, e.g., an environmental change, or an explicit request of a user or an external system". During the development of these functions, *Realization Policies* have to be considered. A *Realization Policy* is here seen as "a constraint for the development of the system under development, including security policies, desired architecture styles, COTS or open source to be used, development activities, and development technology". Realization policies may be influenced by the *UI Style* according to which the user interface components of a system have to be designed. Thus, a *UI Style* "defines the look and feel (i.e., the appearance of the user interface), respectively the representation rules to be considered during UI development". All the introduced issues related to the Internal System View will serve as input for subsequent development phases where they will be aligned with other elements that are of specific importance there (e.g., system component).

5 Application and Adaptation of the Model

The reference issue model introduced in this paper is no end in itself. Rather, the aim of this model is to clarify the issues that are to be discussed with stakeholders in order to derive requirements for a BIS. In particular, the introduced model can act as a foundation for systematic RE and be used for an algorithmic derivation of corresponding elicitation instructions. In our previous work [18], we have presented an approach that explains the reflection of issues in a requirements elicitation guideline.

According to this approach, the described model leads to the following elicitation sequence (exemplary for the Business Environment View introduced in Section 4.1): In a first step, basic information about the project is elicited. Then, the business areas that are in the scope of the project are elaborated, before the interacting business roles, handled business events, and relevant regulations can be identified. Based on these issues, business services that are provided by the business areas can then be defined, and so forth.

In this regard, knowledge of the issues with which a certain requirement is concerned can also help to prioritize requirements [29] on a more objective basis. As an issue represents (more or less) an element of the real world, there are clear criteria according to which its value can be determined. Building an aligned value model and using it for requirements prioritization can improve the reliability of prioritization decisions [30]. Besides prioritization with regard to the implementation order, the usage of such prioritization can also help to guide the entire RE process. If, for instance, 20 business services have been identified, but time and budget are limited, early prioritization can focus the subsequent elicitation steps on only those business services that promise the best cost-value-ratio.

However, besides a foundation for elicitation and prioritization, the reference issue model introduced in this paper can also be used to better align other software engineering methodologies with an RE approach. For instance, models that guide architecture design can be aligned with the reference issue model in order to explain how corresponding decision can be made based on the supported real world. This helps to elaborate method interfaces that must exist in order to make different development methods more interoperable. Finally, the reference issue model can also be used as a foundation for change management. In particular, the relationships between different issues allow adopting traceability concepts that are indispensable for implementing impact analysis. For instance, when a certain regulation changes, the issues that are affected by this change (e.g., business processes) can be determined much easier.

However, even though we carefully elaborated the model using a systematic manner, we are aware that the issues that are actually relevant in a certain development context may vary. Thus, like every reference model, the model described in this paper also needs to be adapted before being used as a basis for RE in a certain project. In particular, the reference issue model should be tailored to reflect the actual information needs that actually exist. This may include deletion, extension, or modification of issues. For instance, when no one cares about a certain issue (e.g., information about this issue does not influence any development decision), this issue can be deleted from the model. In contrast, when very specific information is needed, concrete issue attributes (e.g., the education of a user role) or even completely new issues (e.g., "physical engine") can be added individually. The approach of [27] can be used to systematically identify the information needs of different engineering roles.

In this regard, tailoring of the model also includes determining how certain issues have to be documented in a certain project or development context. For this purpose, artifact models have to be defined that prescribe the actual documents, templates, diagrams, figures, etc. according to which certain issues and corresponding requirements have to be described. This separation of issues from their representation is very important, as there does not need to be a one-to-one relationship. In agile development, for instance, many issues are discussed, but not explicitly specified, while in other contexts, different issues may be represented in one common artifact.

Furthermore, the point in time at which a certain issue must be discussed is also not fixed. For instance, it may be possible that very technical issues such as the existing middleware must be known earlier than the business services. The overall processing order of issues can therefore differ significantly once the model has been tailored. In this case, only the relationship stereotypes still have to be considered, as, for instance, a contained issue cannot be discussed before a containing issue has not been discussed.

However, besides tailoring the model based on information needs and desired representations, also characteristics of the development environment are important drivers [28]. This holds especially true as the multitude of BIS is built in a reuse-based manner today. The incorporation of corresponding reuse constraints in a RE process is therefore needed to make a requirements engineer aware of what is economically feasible and what is not. During an elicitation session, this knowledge can then be used to better discuss and negotiate requirements with the stakeholders [18]. For this purpose, the issues in the (tailored) model can be enriched with reuse constraints that limit their instantiation. For instance, the issue "technical infrastructure component" can be restricted by a constraint that states that the BIS to be developed can only run on Linux, as there are core components that are only intended for this operating system.

6 Related Work

The idea of using conceptual models to clarify elements to be addressed by an engineering method is not new and rather the state of the art in method engineering

[6]. However, to our knowledge, no model exists so far that provides such a comprehensive foundation of RE for BIS as described by our model in this paper.

Models such as those described in the BPMN specification [31] or in [32], for instance, focus on very specific aspects (e.g., business processes) and elaborate them in detail. However, these models typically neglect all other issues that are necessary in BIS development. Thus, our model is different, as it covers the issues that are relevant here more completely. Indeed, there are also some broader models that also cover additional aspects, but these models are either typically very specific for a certain application domain (e.g., the logistics domain, e.g., [33]) or focused on the business side only [34]. In this regard, our model is both broader and more generic, and covers business- and IT-relevant issues holistically.

However, being too generic such as [3], is also no option. As such models rather deal with meta- or even meta-meta concepts, for instance, objects, associations, and states (e.g. [32]) instead of describing real issues, they do not allow conceptualizing RE for BIS sufficiently. To avoid this problem, our model addresses two different levels of the MOF [26] stack and therefore addresses both concrete issues and also underlying meta-concepts. Nevertheless, there already exist many (meta-)models that aim at formalizing RE (e.g., [35] and [37]). However, it can often be observed that such models (e.g. [35]) do not clearly distinguish requirements and other meta-concepts (e.g., rationales, priorities, sources, etc.) from artifacts (e.g., use cases) or contents (e.g., users), which makes it hard to consider them as a formalization of the issues to be addressed. Our model therefore explicitly separates them by using different MOF levels.

Existing approaches that explicitly distinguish conceptual models and artifact models are rare. So far, only [36] and [38] have proposed such a distinction. However, as [36] is from the area of embedded systems, it is not applicable in the BIS domain and [38] does not explicitly discuss the relations between the issues.

7 Conclusion

We have identified that there are problems when trying to use methods from the RE and business analysis in an integrated manner. This gap leads to unclear mappings and makes it difficult to perform methods and approaches that exist in both areas. We therefore proposed a reference issue model for BIS that should serve as a foundation for the integration of RE and business analysis as well as corresponding tasks.

The issues described in this model reflect inherent elements that are either part of an application or part of an application's environment. In RE, issues determine the elements for which requirements have to be elicited in order to specify a system. However, as requirements processes for BIS are basically different from requirements processes for other kinds of systems (e.g., embedded systems) [19], the issues to be discussed in this domain are specific and not generalizable to terms such as "requirements", "use cases", etc. only. The model introduced in this paper therefore describes the issues that are typically relevant in BIS projects, as well as the relationships between them. The model therefore externalizes best practice and tacit knowledge by formalizing the conceptual world to be processed during RE in this context. In this regard, we are aware that the set of issues introduced in this work is not new. The novelty of this model lies in the more precise and more complete clarification how BPM concepts are related with concepts of (traditional) RE.

However, like any other reference model, the reference issue model cannot be applied without adaptation either. Rather, it must be tailored based on the information needs of the people involved in downstream development activities, as well as based on required representation forms and existing constraints. Based on a tailored version of this model, RE processes and corresponding instructions can then be automatically generated using an approach such as described in [18]. Thus, even though the reference issue model is not intended to be used out-of-the-box, it is an indispensable means for tailoring, as it reflects established elicitation procedures in the information system domain. Such a knowledge prevents method engineers from violating best practices when defining own requirements processes. For the near future, we plan to iterate the model further with external experts from the RE and BPM community in order to get a broader commitment and validation of the ideas. Furthermore, the development of other reference models such as an artifact reference model is to be done. Finally, we plan to develop analysis and modeling methods to increase the effectiveness and efficiency of RE in BIS projects.

References

1. The Standish Group International: CHAOS Manifesto (2011), `http://blog.standishgroup.com` (last visited: January 18, 2011)
2. Gulla, J.A.: Understanding Requirements in Enterprise System Projects. In: RE 2004. IEEE (2004)
3. Etien, A., Rolland, C.: Measuring the fitness relationship. Requirements Engineering Journal, 184–197 (2005)
4. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Berlin (2007)
5. Robertson, S., Robertson, J.: Mastering the Requirements Process. Addison-Wesley, Reading (1999)
6. Jeusfeld, M.A., Jarke, M., Mylopoulos, J.: Metamodeling for Method Engineering. The MIT Press, Cambridge (2009)
7. IEEE Computer Society: IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. IEEE (1998)
8. IEEE Computer Society: IEEE 1233-1998 - IEEE Guide for Developing System Requirements Specifications Description. IEEE (1998)
9. IEEE Computer Society: IEEE 1362-1998 - IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document Description. IEEE (1998)
10. BMI: Das V-Modell XT, `http://www.v-modell-xt.de/` (last visited: February 14, 2012)
11. Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. Springer, Berlin (1998)
12. Hay, D.C.: Requirements Analysis: From Business Views to Architecture. Prentice Hall Professional, Upper Saddle River (2003)
13. Kruchten, P.: The Rational Unified Process. Addison-Wesley, Boston (2000)
14. Adam, S., Doerr, J., Eisenbarth, M., Gross, A.: Using Task-oriented Requirements Engineering in Different Domains. In: RE 2009, pp. 267–272. IEEE (2009)

15. Adam, S., Naab, M., Trapp, M.: A Service-Oriented View on Business Processes and Supporting Applications. In: CAiSE 2010. Springer, Berlin (2010)
16. Card Sorting, http://www.usability.gov/methods/design_site/cardsort.html (January 29 2012)
17. Alexander, I.F.: A Taxonomy of Stakeholders. Int. Journal of Technology and Human Interaction, IJTHI 1, 23–59 (2005)
18. Adam, S.: Providing Software Product Line Knowledge to Requirements Engineers – A Template for Elicitation Instructions. In: Regnell, B., Damian, D. (eds.) REFSQ 2012. LNCS, vol. 7195, pp. 147–164. Springer, Heidelberg (2012)
19. Nuseibeh, B., Easterbrook, S.: The Future of Software Engineering. In: ICSE 2000, pp. 35–46. ACM Press, New York (2000)
20. Wiegers, K.E.: Software Requirements. Microsoft Press (2005)
21. IEEE: Computer Society. Software Engineering Body of Knowledge. IEEE (2004)
22. Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons, Chichester (2009)
23. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley, New York (1998)
24. The Free Dictionary, http://www.thefreedictionary.com/ (last visited: January 01, 2012)
25. Riegel, N., Adam, S., Gross, A.: Addressing Requirements Engineering Challenges in the Context of Emergent Systems. In: RESS 2011. IEEE (2011)
26. Object Management Group: OMG's Meta Object Facility (MOF), http://www.omg.org/mof/ (last visited: January 18, 2012)
27. Gross, A.: Perspective-based Specification of Efficiently and Effectively Usable Requirements Documents. In: Doctoral Symposium RE 2010, Sydney (2010)
28. Adam, S., Doerr, J., Ehresmann, M., Wenzel, P.: Incorporating SPL Knowledge into a Requirements Process for Information Systems. In: PLREQ 2010, Essen, pp. 54–66 (2010)
29. Wiegers, K.E.: First Things First: Prioritizing Requirements. Software Development 7, 48–53 (1999)
30. Riegel, N., Adam, S., Gross, A.: Addressing requirements engineering challenges in the context of Emergent Systems. In: RESS, pp. 6–9. IEEE, Trento (2011)
31. Object Management Group: Business Process Model and Notation (BPMN), http://www.omg.org/spec/BPMN/2.0/ (last visited January 18, 2012)
32. Soffer, P., Wand, Y.: Goal-Driven Analysis of Process Model Validity. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 521–535. Springer, Heidelberg (2004)
33. Scheer, A.-W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Springer, Berlin (1994)
34. Champion, R.E.M., Moores, T.T.: Exploiting an enterprise model during systems' requirements capture and analysis. In: ICRE 1996, Colorado Springs (1996)
35. Goknil, A., Kurtev, I., van den Berg, K.: A Metamodeling Approach for Reasoning about Requirements. In: Schieferdecker, I., Hartman, A. (eds.) ECMDA-FA 2008. LNCS, vol. 5095, pp. 310–325. Springer, Heidelberg (2008)
36. von Knethen, A.: Trace model for system requirements changes on embedded systems. In: IWPSE 2001, New York (2001)
37. Bubenko, J.A.J., Persson, A., Stirna, J.: User Guide of Knowledge Management Approach Using Enterprise Knowledge Patterns, deliverable D3, IST Programme project IST-2000-28401 HyperKnowledge, KTH, Schweden (2001)
38. Méndez-Fernández, D., Lochmann, K., Penzenstadler, B., Wagner, S.: A Case Study on the Application of an Artefact-Based Requirements Engineering Approach. In: 15th EASE, IET (2011)

Local Behavior Similarity

Matthias Kunze and Mathias Weske

Hasso Plattner Institute at the University of Potsdam,
Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam, Germany
{matthias.kunze,mathias.weske}@hpi.uni-potsdam.de

Abstract. Business process models explicitly capture an organization's operations and thus are essential to a process oriented organization. Typically, hundreds or thousands of models are stored in business process repositories. Effective capabilities to manage and, in particular, search are required to leverage stored business process models.

Yet, search remains a challenge, because business processes cannot easily be compared. Existing approaches to process similarity do not support queries that are significantly smaller than sought models and contain only few, yet important, aspects.

In this paper, we introduce a novel approach to behavioral similarity search that is sensitive to local behavior inclusion, i.e., it will feature models that contain the behavior of a query. This is achieved by comparing local behavioral relationships of a query model with global relationships of candidate models. We present the formal foundation of this approach, derive a similarity measure, and illustrate the applicability of our approach, also with respect to complexity.

Keywords: process model search, behavioral similarity, weak order, precedence order.

1 Introduction

In recent years, BPM attained a considerable stance in large organizations. Business process models are the central asset of a process-oriented organization, as they capture the knowledge to carry out an organization's operations and are used for many purposes, including, but not limited to, documentation, analysis, automation, and certification. Since business process management has increasingly gained impact, organizations maintain hundreds or thousands of process models in so-called business process repositories [1]. Such repositories bear large benefits to an organization, for instance, reusing existing models leads to more efficient and consistent process design as repeated work can be avoided or best practices can be referenced. Duplicate or superseding models that coexist undetected due to mergers and acquisitions could be discovered and consolidated. Changes to one process model shall be propagated to any other model that holds similar conditions for the change. Hence, effective use of process model repositories requires, in particular, meaningful techniques to search within them.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 107–120, 2012.

In practice, simple folder navigation and keyword text search are established tools to search for processes, which only offer very limited capabilities to search for specific peculiarities of business processes and may only be of limited value in scenarios as the ones outlined above. In academia, the topic of process model search received considerable interest and many researchers addressed it by means of process model similarity, i.e., they provide measures that compare process models by structural or behavioral aspects. Unfortunately, most approaches do not address subsumption of these aspects and will only identify processes that share a large ratio of aspects compared to their size. This does not align well with the use case of process model search, where a person only expresses very few yet relevant aspects in a query, e.g., activities and the order in which they shall be executed. This person then expects to find those models that contain the aspects of the query, no matter whether they are significantly larger or allow for more versatile behavior.

In this paper, we address this issue and propose a solution to the following search problem. *Given a query model q and a set of candidate process models P, find all process models from the candidate set that allow to execute all or some activities of q in the requested order in at least one process instance.* For our approach, we assume query and candidate models to be of the same domain, i.e., there is no dedicated query language required. At the same time, query and candidate models may be expressed in different modeling languages, e.g., BPMN, EPC, or UML activity diagrams, as our approach is independent from a modeling language. To provide a continuous similarity function, our approach also accepts candidate models that do not completely match the query but are quite similar to it. On the one hand, this enables finding top k candidates, even if no complete match can be found and, on the other hand, it provides an implicit ranking among the found process models.

Comparing behavioral similarity of process models is typically computationally expensive, as it involves comparing sets of traces or the state space that two models can produce. Both can grow exponentially and even be infinite. Our approach mitigates such cost, as it can be computed efficiently and comparison time is cubic to the number of activities of both process models in the worst case and quadratic in most cases.

The remainder of this paper is structured as follows. Section 2 illustrates a practical scenario to discuss the above search problem. Section 3 introduces formal preliminaries our work builds upon, before we present the concept of local behavior inclusion that matches local behavioral relations of a query against global behavioral relations of a candidate model in Section 4. There, we further derive and discuss a continuous similarity measure. We position our approach in the current state of research and discuss related work in Section 5, before we conclude the paper with a short summary and an outlook on future work.

2 Motivation

Assume a company that manufactures and sells bicycles. The company has a large product portfolio and customers can order a set of usual bikes from stock, construct a custom bike from a set of available configuration options (build-to-order), or request offers for individual makings, such as foldable off-road tandems. Accordingly, the company maintains several sales processes among their large collection of business processes. A sales person might search such processes that comprise receiving an order and shipping goods. Also reception of payment is required.

With existing technologies, the sales person would be limited to browsing the process repository or searching for keywords in the inscription of process activities. However, they may want to impose ordering constraints, i.e., the goods shall be shipped before a payment is received. Naturally, the sales person would create a very simple process query, similar to the one illustrated in Fig. 1.

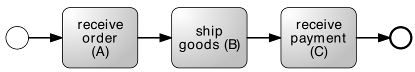
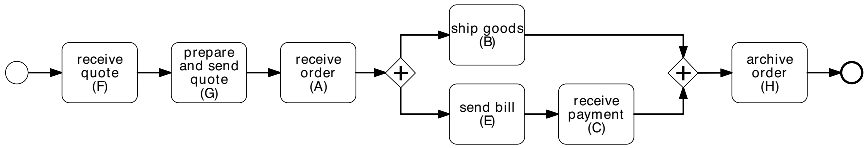


Fig. 1. Example query q that consists of required activities and their desired execution order

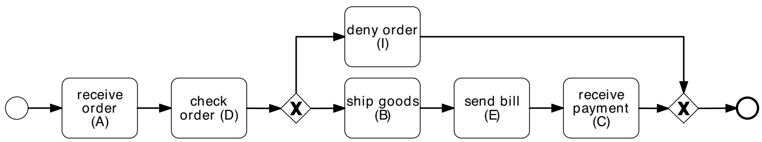
This query requires that any matching process shall execute the activities *receive order*, *ship goods*, and *receive payment*. However, it is obvious that no process would consist of exactly these activities, since payment requires a bill sent to the customer. Hence, a sequence flow edge in such a query expresses that the execution order of the activities shall be preserved, while other activities may be carried out between them. Compared to existing process querying approaches, cf. Section 5, the sales person does not need a specific query language. They simply formulate their requirements as a process model.

Consider the sales processes of this company, depicted in Fig. 2. All of them contain the activities requested in the query, cf. Fig. 1. Process model m_1 in Fig. 2a expresses an individual order process, i.e., a customer first requests a quote on a bicycle to be manufactured individually. Fig. 2b (m_2) and 2c (m_3) are processes supporting build-to-order and order-from-stock. Please note that, in m_3, the order of *ship goods* and *receive payments* is reversed, as this is a product from an online shop, where orders require upfront payments.

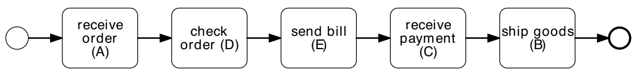
According to the sales person's search query q, the processes m_1 and m_2 both match the query. However, the build-to-order process, m_2, has a higher resemblance with the query, as it contains less additional activities than the individual order process, m_1. If these were the only search results that match the query, the sales person would miss processes that are quite similar to the query, yet violate it to a certain extent. Hence, an effective search solution should also suggest the order-from-stock process m_3. Nevertheless, it should be ranked lower than the other processes of Fig. 2.



(a) Individual-order process model m_1, where a customer requests a quote before ordering the product.



(b) Build-to-order process model m_2, where a customer can configure a product according to their needs. If a configuration can temporarily not be provided, the order will be denied.



(c) Order-from-stock process model m_3, where a customer orders a product and has to pay before the product will be shipped, e.g., in an online shop.

Fig. 2. Matching candidates for the query in Fig. 1

3 Preliminaries

Before we present a solution to the above scenario, we need to provide the basis our approach builds upon. We will refer to activities in the process models by their label annotation, e.g., (A) for *receive quote*, hereafter.

Nowadays, we observe various graph-based business process modeling languages, such as BPMN, EPC, YAWL, and UML Activity Diagrams, which vary in the vocabulary they provide and thus their expressiveness. However, they share common concepts with regards to execution semantics, i.e., activities to express a work to be conducted and gateways to determine control flow routing [2]. Without loss of generality, we resort to the essential behavioral concepts for our notion of a process model to keep the paper concise.

Definition 1 (Process Model). *A* process model *is a tuple* $P = (A, C, N, F, T)$ *where:*

- *A is a finite non-empty set of activity nodes, C is a finite set of control nodes, and $N = A \cup C$ is a set of nodes with $A \cap C = \emptyset$,*
- *$F \subset N \times N$ is the flow relation*
- *$\bullet y = \{x \in N | (x, y) \in F\}$ and $x\bullet = \{y \in N | (x, y) \in F\}$ denote direct predecessors and successors, we require $\forall\ a \in A : |\bullet a| \leq 1 \wedge |a \bullet| \leq 1$,*
- *there exists exactly one start node $s \in A$, such that $\bullet s = \emptyset$, and exactly one end node $e \in A$, such that $e\bullet = \emptyset$,*
- *$(N, F \cup \{(e, s)\})$ is a strongly connected graph,*
- *$T : C \rightarrow \{and, xor\}$ associates each control node with a type.*

We require a dedicated start node s to indicate initialization, and a dedicated end node e to indicate termination of a process. Process models that do not expose this requirement may be refactored using methods presented in [3]. We rely on BPMN to illustrate examples, hereafter.

For our notion of a process model, we assume trace semantics, which follow from common Petri net-based formalizations [4], i.e., the behavior of a process model $P = (A, C, N, F, T)$ is expressed by a set of traces $\mathcal{T}_P$ that can be produced by P. A trace is a sequence of activities $\sigma = \langle a_1, ..., a_n \rangle$ with $n > 0, n \in \mathbb{N}, a_i \in A$ for all $0 < i \leq n$, which represents their allowed execution order in P. Note that a process model may have an infinite number of traces and traces can be infinitely long, e.g., due to loops. As a trace and the set of traces may be infinite for certain processes, e.g., through loops, we leverage a behavioral abstraction that looks at behavioral relations in terms of execution order between any pair of activities in a process model.

Definition 2 (Behavioral Relations). *Let $P = (A, C, N, F, T)$ be a process model and $\mathcal{T}_P$ its set of traces. We define two* behavioral relations*:*

- *The* precedence relation*, $>_P \subseteq (A \times A)$ contains all pairs (x, y), such that there exists a trace $\mathcal{T} = \langle a_1, a_2, \ldots, a_n \rangle$ in $\mathcal{T}_P$ and an index $j \in \{1, 2, \ldots, n\}$ for which holds $a_j = x$ and $a_{j+1} = y$ (local relations).*
- *The* weak order relation *$\succ_P \subseteq (A \times A)$ contains all pairs (x, y), such that there exists a trace $\mathcal{T} = \langle a_1, a_2, \ldots, a_n \rangle$ in $\mathcal{T}_P$ and two indices $j, k \in \{1, 2, \ldots, n\}$ with $j < k$ for which holds $a_j = x$ and $a_k = y$ (global relations).*

Table 1 depicts these relations for the example models of Fig. 1 and 2 in the form of a matrix, where the row of a matrix cell represents activity x and the column denotes activity y of the respective behavioral relation that (x, y) belongs to.

The *precedence relation* emerged in the field of process mining and is used as the base relation for the α-algorithm [5]. In practice, the precedence relation of a process comprises all 2-grams that may occur in any trace of this process. As this addresses only pairs of activities that may be executed directly one after another in a process instance, we refer to this as *local relations* of a process model. The precedence relation $>_{m1}$ of the process model in Fig. 2a, contains the pairs (F, G) and (G, A). Since F and A cannot be executed directly after one another, the pair (F, A) is not in the precedence relation of this process. Due to the interleaving execution characteristics of parallel paths, activity B can be

Table 1. Precedence ($>$) and weak order ($\succ$) relations of process models from Fig. 1 and 2

	A	B	C
A		$>$	
B			$>$
C			

(a) $>_q$

	F	G	A	B	E	C	H
F		$>$					
G			$>$				
A				$>$	$>$		
B					$>$	$>$	$>$
E				$>$		$>$	
C				$>$			$>$
H							

(b) $>_{m1}$

	F	G	A	B	E	C	H
F		$\succ$	$\succ$	$\succ$	$\succ$	$\succ$	$\succ$
G			$\succ$	$\succ$	$\succ$	$\succ$	$\succ$
A				$\succ$	$\succ$	$\succ$	$\succ$
B					$\succ$	$\succ$	$\succ$
E				$\succ$		$\succ$	$\succ$
C				$\succ$			$\succ$
H							

(c) $\succ_{m1}$

	A	D	I	B	E	C
A		$\succ$	$\succ$	$\succ$	$\succ$	$\succ$
D			$\succ$	$\succ$	$\succ$	$\succ$
I						
B					$\succ$	$\succ$
E						$\succ$
C						

(d) $\succ_{m2}$

	A	D	E	C	B
A		$\succ$	$\succ$	$\succ$	$\succ$
D			$\succ$	$\succ$	$\succ$
E				$\succ$	$\succ$
C					$\succ$
B					

(e) $\succ_{m3}$

executed before, between, and after E or C, which leads to the according pairs (B, C), (C, B), (B, E), and (E, B) of $>_{m1}$

In contrast, the *weak order relation* captures ordering relations between any pair of activities, i.e., also transitive relations if both activities can occur in one process instance. Here, the ordering relation of each activity to every other activity in that process is determined, and therefore, we refer to this as *global relationships*. If a pair of activities is not in weak order, these activities cannot be executed in the same process instance. Weak order relations build the basis for behavioral profiles [6] that characterize the behavior of a process by enriched behavioral semantics, i.e., strict order, exclusion, or interleaving order. In previous work [7], we showed that comparing behavioral profiles of a pair of process models provides a good approximation of process model similarity. The weak order relation $\succ_{m1}$ of the example process model in Fig. 2a contains the pair (F, A) as, from a global point of view, execution of F will eventually be followed by execution of A.

Business process modeling languages may differ in their concepts to express execution semantics, but agree on the same concept of activities as central work items and some restrictions on the order of their execution, e.g., sequence flow in BPMN and EPC. Hence, traces can be generated independently of a modeling language that captures the behavior of these processes. As both order relations presented above are based on trace semantics, search techniques based on them are also independent of a modeling language.

An essential aspect to comparing process models is the identification of corresponding activities [8]. Features, such as structure and behavior of two process models can only be compared, if they exist between corresponding activities. For instance, the common precedence relation between the tasks A and B in the process model q in Fig. 1 and m_1 in Fig. 2a can only be identified, if we know that the respective activities, e.g., A, are identical in the application of the process. To capture corresponding activities, we define an activity mapping for every pair of process models.

Definition 3 (Activity Mapping). *An* activity mapping *of two process models Q and P with their sets of activities A_Q and A_P, respectively, is a bijective function $\mathcal{M} : A_Q^* \rightarrow A_P^*$ that maps a set of activity nodes from Q, i.e., $A_Q^* \subseteq A_Q$, to a set of corresponding activities in P, i.e., $A_P^* \subseteq A_P$.*

Such a mapping can be created by different means based on the activities' labels. This is based on the assumption that similar or equal activity labels in distinct process models have the same semantics and granularity. As this is not the focus of this paper, we will briefly discuss related work on activity mappings in Section 5. Here, we rely on equal activity labels in our example. This is supported through equal character annotations in the process models, i.e., task *receive order* has the annotation A, among all models.

4 Local Behavior Similarity

As introduced in Section 1, we are interested in finding all models that are similar or complementary to a given query. In this section, we will first introduce our notion of behavioral subsumption, i.e., a method to determine whether a process model subsumes the behavior of a given query. Based on this, we then derive a similarity function that further allows finding such process models, which do not exactly match a query, and that enables ranking of process models returned as search result.

4.1 Matching Local Behavior with Global Behavior

Table 1 illustrates precedence and weak order relations of the example process models. Examining both behavioral relations of the process m_1 reveals intriguing links between the precedence relation, cf. Table 2b, and the weak order relation, cf. Table 2c: If a pair of activities is not contained in the weak order relation, it will also not be contained in the precedence relation, e.g., (E, A). However, pairs that are not in a precedence relation may occur in a weak order relation, e.g., (E, H). The reason for these properties lies in the peculiarities of the respective behavioral relations, i.e., precedence looks at local relations and weak order at global relations, which includes local relations.

We exploit these properties to match a query process model with a candidate process model. If and only if every pair of activities in the precedence order relation of the query model is contained in the weak order relation of the candidate model, the behavior of the query is subsumed by the candidate and it is considered a complete match, otherwise, if at least one activity pair of the query can be matched with a candidate model, we refer to this as a partial map.

Definition 4 (Local Behavior Inclusion). *Given two process models Q and P, the local behavior of a pair of activities $x, y \in A_Q^*$ is included in the global behavior of P, denoted as $(x, y) \in (>_Q \sqcap \succ_P)$, iff $(x, y) \in >_Q \wedge (x', y') \in \succ_P$, with $\mathcal{M}(x) = x'$ and $\mathcal{M}(y) = y'$.*

Here, we ignore those pairs from evaluation that are not in precedence relation, as this only expresses that these activities may not be executed in a direct sequence. In contrast, if a pair of activities is not in the weak order relation, both activities can never occur in one process instance.

The above definition leads to the following properties of matching a query against a candidate model. A pair (x, y) in the precedence relation of a query model is matched with any sequence of activities in a candidate model, where activity x can be executed before activity y. It is not required, that both or even any activity appear in all traces of that process model. However, at least one such trace must exist. This can be seen in the example process model m_2 in Fig. 2b, which, according to Definition 4, also completely matches the query model q: Every pair of activities in Table 2a is contained in the matrix of Table 2d. Yet, the process model allows *denying the order (A)*, which bypasses activities *ship goods (B)* and *receive payment (C)*.

Parallel paths in a process model allow executing activities in interleaving order, such that activities x and y appear twice in a behavioral relation in inverse order, i.e., (x, y) and (y, x). Hence, if a query requires a pair of activities to be executed in a sequence, i.e., $(x, y) \in >_Q$, and a candidate model allows to execute corresponding activities in interleaving order, i.e., $\{(x', y'), (y', x')\} \subseteq \succ_P$, this includes the sequential execution imposed by the query and $(x, y) \in (>_Q \sqcap \succ_P)$. For an example, refer to activities B and C that are in sequence in the query q, cf. Table 2a, and in interleaving order in process model m_1, cf. Table 2c. Conversely, if a query contains two activities in interleaving order, any candidate model that completely matches the query must also provide these activities in interleaving order.

Mutual exclusion of activities cannot be expressed in a query, because a query is considered as an underspecified process that expresses desired behavior—a behavioral relation absent in the query that is present in a candidate model does not affect matching. Given activities x and y that are mutually exclusive in a query, i.e., $\{(x, y), (y, x)\} \nsubseteq >_Q$, and thus $\{(x, y), (y, x)\} \nsubseteq \succ_Q$, their corresponding activities x' and y' may occur together in traces of a completely matching model, i.e., $|\{(x', y'), (y', x')\} \cap \succ_P| \geq 0$. This is intended, as such a fragment of the query may be contained in a loop of a candidate model, which then would allow to execute x and y in one process instance.

If a query contains an activity that has no corresponding activity in a candidate model, cf. Definition 3, this activity cannot participate in matched weak order relations, and the candidate cannot be a complete match.

It is important to mention that the above definition of local behavioral inclusion is not equivalent to the notion of trace equivalence [9] or bisimulation [10]. This is, because behavioral relations are an abstraction of the actual traces a process can produce, and hide, for example, cardinalities of repeatedly executed activities.

4.2 Similarity Based on Local Behavioral Inclusion

The local behavior inclusion $>_Q \sqcap \succ_P$, cf. Definition 4, enables matchmaking of candidate process models with a query. However, for process model search it is desirable to obtain a ranked list of process models, with the best match in the first position. This requires a notion of similarity, to find models that are good candidates even if no complete match can be found and to rank process models of the result. Hence, we derived a similarity measure, which yields 0 if no precedence relations of a query are matched with weak order relations in a candidate process model and 1 if all are, i.e., the behavior of a query Q is completely included in the behavior of the candidate process model P.

$$sim_{\sqcap}(Q, P) = \frac{|>_Q \sqcap \succ_P|}{|>_Q|}$$

This similarity measure is asymmetric and quantifies the ratio of order relations of a query shared by a candidate process model. If two different candidate process models share the same number of order relations with the query Q, they will be equally similar to Q by $sim_{\sqcap}(Q, P)$. In the above example, this is the case for process models m_1 and m_2 in Figures 2a and 2b, as can be seen in the behavioral relations in Table 1. Here, both process models are a complete match against the query q in Fig. 1, and thus their similarity to the query is 1.

In such a case, the process model that shares more nodes with the query should be preferred, because it reveals fewer additional activities and thus is considered behaviorally more similar to the query. We account for shared activities by the well established Jaccard coefficient and assess the relative size of the activity mapping of two process models P and Q,

$$sim_A(Q, P) = \frac{|A_Q \cap' A_P|}{|A_Q| + |A_P| - |A_Q \cap' A_P|}$$

where $|A_Q \cap' A_P|$ denotes the size of the activity mapping, i.e., $a \in (A_Q \cap' A_P) \Leftrightarrow a \in A_Q^* \wedge a' \in A_P^* \wedge \mathcal{M}(a) = a'$, according to Definition 3. The final similarity function that is used for searching and ranking results is constructed by a weighted sum of the elementary similarity functions above.

Definition 5 (Local Behavior Similarity). *Given two process models Q, P, the local behavior similarity is defined as follows:*

$$sim(Q, P) = \alpha \cdot sim_{\sqcap}(Q, P) + (1 - \alpha) \cdot sim_A(Q, P)$$

where $\alpha \in \mathbb{R}, 0 < \alpha < 1$ denotes a weighting factor to account for matched activities.

The weighting factor α is used to adjust the impact of corresponding nodes on the similarity and shall be rather large. However, it may be set according to a searcher's requirements. If α falls below 0.5, the query will increasingly prefer candidate models that share more nodes rather than matching behavioral relations, which contradicts the use case illustrated in Section 2.

It is worth mentioning that models with identical weak order relations will receive the highest similarity, i.e., $\succ_Q = \succ_P \Rightarrow sim(Q, P) = 1$, even if Q and P have different precedence relations. This is due to the property that any pair of activities of the precedence relation of a process model is also contained in its weak order relations, cf. Definition 2. This leads to $sim(P, P) = 1$.

4.3 Discussion and Evaluation

For our example, we assume $\alpha = 0.9$. Hence, the amount of non-shared activities shall only have a small impact. The most similar process model to the query model q in Fig. 1 is model m_2 in Fig. 2b. As mentioned above, $sim_{\sqcap}(q, m_2) = 1$ and both models share three out of six activities, i.e., $sim_A(q, m_1) = 1/2$, which yields a local behavioral similarity of $sim(q, m_1) = 0.95$. Candidate model m_1, cf. Fig. 2a has more activities that q does not share and thus receives a slightly lower similarity, $sim(q, m_2) = 0.94$. Still, it shall be considered a very good match, as it completely subsumes the behavioral relations required in q. Process model m_3

in Fig. 2c matches the behavioral relations of q only partially, as the pair (B, C) is not contained in $\succ_{m3}$, hence $sim_{\sqcap}(q, m_3) = 1/2$. Nevertheless, it may be a good result candidate, because it represents an arguably relevant case that the searcher may not have foreseen. As m_3 shares more than half of its activities with q, it receives $sim(q, m_3) = 0.51$.

Behavioral relations of a process model can be computed in cubic time to the size of the process model, for sound process models, and in exponential time for bounded models [6]. However, in practice, computation proves to be quite fast, and can be done before search is performed, e.g., every time a process model is updated. The precomputed relations are then stored for search that can be performed in cubic time to the size of query and process model. In most cases, however, the precedence relation contains only few pairs, which leads to quadratic time complexity for such queries.

To validate the practical applicability of our approach, we implemented the local behavior similarity and tested it against a set of 765 pairs of process models from the SAP reference model collection [11]. In our experiment, we precomputed the behavioral relations of all models and queries, and kept them in memory to exclude time for I/O operations. We ran above comparisons on a 2.8GHz processor in a Java Virtual machine that had 100MB memory assigned. Computation turned out to be quite fast, with 1.67ms median time (2.78ms in average) to compare two process models. Even for exhaustive search, i.e., comparing the query with every process model in a repository, this provides acceptable search times.

5 Related Work

At the basis of comparing process models stands the identification of an *activity mapping*, cf. Definition 3. For process models, this is typically based on the labels of activities using syntactic or semantic similarity measures. An example of the former is the string edit distance [12], which counts the minimal number of operations (insert, remove, or replace characters) to transform one label into another. Two activities are considered similar, if their string edit distance is below a certain threshold, and the best matches provide the activity mapping. This approach has been applied in [13,14,15,7]. Further approaches to syntactic label similarity generate n-grams of labels and compare them by means of a vector-space model, used in [16,17]. Semantic approaches have been employed in [16,18,19] and typically leverage natural language processing [20], such as elimination of stop words, word stemming, and utilize dictionaries, e.g., WordNet [21], to account for homonyms or synonyms , etc. Still, automatic identification of corresponding activities is far from trivial, due to heterogeneity in the used terminology and granularity of labels. Dijkman et al. [19] addressed this issue by means of a human assessment of similarity.

Once an activity mapping has been determined, process models can be compared by structural or behavioral aspects [8]. Structural approaches largely resort to a cost-based similarity model, i.e., graph edit distances [16,22,19] or a common share of features [18,23]. Yan et al. [14] derive small, characteristic fragments, e.g., split, merge, and sequence structures, from a query and search for their appearance in stored process models. Behavioral similarities consider execution semantics of a process model by means of the reachable state space or the traces a model can produce [24,17,15]. However, both, the state space and the set of traces, can grow exponentially, and thus, make comparison of process models computationally hard. Behavioral relations build an abstraction over execution traces, as they capture ordering semantics of pairs of activities of a process model. Hence, they provide a compact and finite representation of a process model's behavior, even if the process may produce infinitely many and infinitely long traces. We have formally introduced the basics of these relations and their origin in Section 3. Existing work [24,25,7] leverages behavioral relations to evaluate behavioral similarity, resorting to either precedence or weak order relations, whereas, in this work, we combine both.

All of the above approaches are insensitive to containment, i.e., two models will be highly dissimilar if a large number of non-shared features prevails. While this is well suited to identify duplicates among a large collection of process models, it is inconvenient for process model search, where a query contains only few relevant aspects that desired models shall embrace. The similarity measure presented in this paper compensates for this issue, as it is based on local behavioral inclusion. At the same time, it provides a good assessment of similarity for the intersection of two process models of comparable sizes.

Structured querying approaches the same problem with a different method, where a query contains specific constructs to match certain structural or behavioral properties of a process model, e.g., [26,27,28]. However, either of these approaches requires a custom query language and is restricted to one process model language—BPMN-Q [27] applies to BPMN, BPQL [26] to BPEL. While these two approaches provide a visual query language, IPM-QL [28] requires a custom XML representation for query and stored process models. Behavioral relations are independent from structural aspects or specific characteristics of process modeling languages as they capture ordering semantics based on traces processes can produce, cf. Section 3. Hence, a query can be modeled in one process modeling language, e.g., EPC, and can nevertheless be compared with models captured in different languages, e.g., BPMN, which allows search in heterogeneous process model repositories with regards to the used modeling language.

Above structured querying approaches expose the capacity to express wildcards. However, they will only return models that completely match the query as a search result. Slight deviations that the searcher could not foresee will be disregarded. Our approach provides a continuous notion of similarity among matched process models and thus will also propose process models that do not match the query completely, if not sufficient complete matches can be found.

In this paper, we provide only an intrinsic ranking of the search result, derived from the similarity based on behavioral inclusion and shared activities, cf. Section 4.2. Consequently, matched models may be ranked equally, even if they differ in aspects not covered by this similarity. In practice, more sophisticated ranking algorithms consider external information to increase the relevance of search results. These include, but are not limited to, a searcher's personal history, similarity in meta data, e.g., process model author and their organization, or popularity of a candidate model in terms of reuse. While this is a general topic addressed in the field of information retrieval, it has also been approached with a business process background, recently [29,30].

6 Conclusion

In this paper, we introduced a novel similarity measure for process models, which, in particular, features search scenarios, where a query contains only few, yet important aspects of the desired search result. The similarity measure presented in this paper does not only address the behavioral resemblance of two process models but is especially sensitive to behavioral aspects of the query that are contained in a candidate model. After motivating the target use case, we introduced formal preliminaries and presented our solution that is based on the notion of local behavior inclusion: To discover whether candidate process models contain behavioral aspects of a query model, we compare the local behavioral relations of the query with global behavioral relations of the candidate model. In a brief evaluation we illustrated its technical applicability in practical scenarios.

However, this is only a first step towards a readily usable solution. A qualitative evaluation of our approach is required to tell whether it can provide search results that humans consider relevant. The SAP reference model has been used for similarity evaluation before, cf. [19,7], and can be used to assess the general similarity capabilities of the proposed measure. However, the human assessment addresses traditional similarity, i.e., how much two models resemble each other, and does not cover the sensitivity to contained behavior stressed in this work. Hence, we need to artificially generate a test set for this particular use case and evaluate it with process modeling experts.

The proposed relations, i.e., precedence relation and weak order relation, are two ends of a spectrum of behavioral relations: The precedence relation considers only activities that can be executed directly after one another, i.e., looks ahead only one step, whereas the weak order relation looks at global relations, i.e., has an infinite lookahead. Specific properties of a query could be exploited, if one constructed behavioral profiles with a lookahead of n steps, for both query and search model collection, where $n_{query} < n_{model}$ to ensure that the query behavior can be contained in the model behavior.

Although we showed that comparison of two process models by the presented similarity is quite fast, cf. Section 4.3, further work shall examine the applicability of our work to indexing data structures and algorithms that avoid sequential search, i.e., comparison of the query with each stored process model. Note that the local behavior similarity cannot be translated into a proper metric, and thus, cannot easily be used for metric space similarity search that has been proposed for process model repositories before [31,7]. However, existing search algorithms may be adapted to enable efficient search in the partial absence of a proper metric.

References

1. Akkiraju, R., Ivan, A.: Discovering Business Process Similarities: An Empirical Study with SAP Best Practice Business Processes. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 515–526. Springer, Heidelberg (2010)
2. La Rosa, M., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R.M., Mendling, J., Dumas, M., Garcia-Banuelos, L.: Apromore: An advanced process model repository (2009), http://eprints.qut.edu.au/27448/
3. Vanhatalo, J., Völzer, H., Leymann, F., Moser, S.: Automatic Workflow Graph Refactoring and Completion. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) ICSOC 2008. LNCS, vol. 5364, pp. 100–115. Springer, Heidelberg (2008)
4. Lohmann, N., Verbeek, E., Dijkman, R.: Petri Net Transformations for Business Processes – A Survey. In: Jensen, K., van der Aalst, W.M.P. (eds.) ToPNoC II. LNCS, vol. 5460, pp. 46–63. Springer, Heidelberg (2009)
5. van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. on Knowl. and Data Eng. 16, 1128–1142 (2004)
6. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioural profiles of process models. IEEE Trans. Softw. Eng. 37(3), 410–429 (2011)
7. Kunze, M., Weidlich, M., Weske, M.: Behavioral Similarity – A Proper Metric. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer, Heidelberg (2011)
8. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity Search of Business Process Models. IEEE Data Eng. Bull. 32(3), 23–28 (2009)
9. Hidders, J., Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M., Verelst, J.: When are two workflows the same? In: CATS 2005, pp. 3–11. Australian Computer Society, Inc. (2005)
10. van Glabbeek, R.J., Peter Weijland, W.: Branching Time and Abstraction in Bisimulation Semantics. J. ACM 43(3), 555–600 (1996)
11. Curran, T., Keller, G., Ladd, A.: SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Prentice-Hall, Inc., Upper Saddle River (1997)
12. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10, 707 (1966)
13. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
14. Yan, Z., Dijkman, R., Grefen, P.: Fast Business Process Similarity Search with Feature-Based Similarity Estimation. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 60–77. Springer, Heidelberg (2010)
15. Wang, J., He, T., Wen, L., Wu, N., ter Hofstede, A.H.M., Su, J.: A Behavioral Similarity Measure between Labeled Petri Nets Based on Principal Transition Sequences. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 394–401. Springer, Heidelberg (2010)
16. Grigori, D., Corrales, J.C., Bouzeghoub, M.: Behavioral Matchmaking for Service Retrieval. In: ICWS 2006, pp. 145–152. IEEE Computer Society (2006)
17. Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., Zave, P.: Matching and Merging of Statecharts Specifications. In: ICSE 2007, pp. 54–64. IEEE Computer Society (2007)

18. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: APCCM 2007, pp. 71–80. Australian Computer Society, Inc. (2007)
19. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (2011); Special Issue: Semantic Integration of Data, Multimedia, and Services
20. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge (1999)
21. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM 38, 39–41 (1995)
22. Minor, M., Tartakovski, A., Bergmann, R.: Representation and Structure-Based Similarity Assessment for Agile Workflows. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 224–238. Springer, Heidelberg (2007)
23. Lu, R., Sadiq, S.K.: On the Discovery of Preferred Work Practice Through Business Process Variants. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 165–180. Springer, Heidelberg (2007)
24. van der Aalst, W.M.P., Alves de Medeiros, A.K., Weijters, A.J.M.M.T.: Process Equivalence: Comparing Two Process Models Based on Observed Behavior. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 129–144. Springer, Heidelberg (2006)
25. Eshuis, R., Grefen, P.: Structural Matching of BPEL Processes. In: ECOWS 2007, pp. 171–180. IEEE Computer Society (2007)
26. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying Business Processes with BP-QL. In: VLDB, pp. 1255–1258 (2005)
27. Awad, A.: BPMN-Q: A Language to Query Business Processes. In: EMISA, pp. 115–128 (2007)
28. Choi, I., Kim, K., Jang, M.: An XML-based Process Repository and Process Query Language for Integrated Process Management. Knowledge and Process Management 14(4), 303–316 (2007)
29. Awad, A., Sakr, S., Kunze, M., Weske, M.: Design by Selection: A Reuse-Based Approach for Business Process Modeling. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 332–345. Springer, Heidelberg (2011)
30. Burkhart, T., Werth, D., Loos, P.: Flexible Process Support by Automatic Aggregation of Implicit and Explicit User Behavior. In: ICADIWT 2011. IEEE Computer Society (2011)
31. Kunze, M., Weske, M.: Metric trees for efficient similarity search in large process model repositories. In: Muehlen, M.z., Su, J. (eds.) BPM 2010 Workshops. LNBIP, vol. 66, pp. 535–546. Springer, Heidelberg (2011)

Towards Customer-Individual Configurations of Business Process Models

Michael Becker and Stephan Klingner

Department of Business Information Systems, University of Leipzig, Germany
{mbecker,klingner}@informatik.uni-leipzig.de

Abstract. Nowadays business process models are a common approach to describe and analyse existing business processes and to create new processes in a structured way. However, with growing complexity of process models there is a lack of comprehensibility. Using existing notations, it is challenging or even impossible to define temporal and logical constraints between process steps that are not directly connected. We demonstrate a declarative approach for representing business processes that allows for configuration, i.e. selection of process steps, based on a component representation. In addition, we present ways to transform a configuration into a procedural process model using BPMN.

Keywords: Business Process Configuration, Service Modelling, Modularisation.

1 Introduction

The growing economical importance of the service sector is associated with an increasing complexity of services. One example of this trend are offers comprising products and services in so called product-service-systems. These developments are accompanied by a growing demand of customer-individual offers. To achieve fundamental economical aims – despite those challenges – an efficient provision of those services is a necessary precondition. To support a productive and standardised service provision, the modelling of services in terms of service engineering is a widely implemented approach. Various IT-based modelling languages allow for a precise description of the process-related aspects of services. But to widen the focus in terms of individualisation, the consideration of configuration-related requirements is also necessary. Hence, this paper proposes a modelling method aiming to fulfill the specific needs of the configuration of services, as presented in various papers before. This encloses the segregation of semantically related process parts in so called modules [1] as well as the description of dependencies between those modules [2]. Therefore, this paper gives a brief introduction of the concept of modelling service modules and their configuration. According to [1], we define that a service module offers a well-defined functionality via precisely described interfaces. Furthermore, a service module can be used for composition and can, therefore, itself be part of a more coarse-grained service module.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 121–135, 2012.

To support a seamless integration of the proposed modelling method in the overall process of service engineering, two additional steps besides segregation and description of dependencies have to be considered. It is possible to extract service components from existing business process models (*extraction*). These components can then be used as a basis for configuration. Furthermore, business process models can be generated based on customer-individual configurations of services (*generation*).

The main benefit resulting from extraction is the reuse of existing business process models as basis for the creation of configurable models. In doing so, the step towards a configurable service portfolio can be simplified. On the other hand, the generation of business process models based on configurations extends the advantages of process models on the level of individualisation. These business process models are defining the specific process according to the customer-individual offer and therefore can be used as the basis for a workflow description. This paper focuses generation of process models.

In summary, the whole course of action to create configurable service models as proposed in this paper consists of four steps. First, it is necessary to specify the unique service components and establish hierarchical dependencies between these components. This can be done either manual or by extracting these modules from existing business process models. This results in the existence of a component model. Second, it is necessary to declare logical (i.e. non-hierarchical) and temporal dependencies between components based on this component model. Temporal dependencies are evaluated to specify the order of activities and their parallelisation potential. These steps are described in the following section 2. As a third step, the configuration of components conforming to their structure and dependencies is conducted. This configuration is usually established in collaboration with customers, resulting in a customer-individual offer. The configuration is similar to variants in Product Line Engineering according to [3], i.e. a complete configuration can be understood as a specific variant.

Fourth and finally, this customer specific configuration is transformed into a workflow representation. This workflow model can be imported into a workflow management system to guide the process. Configuration and transformation are described in section 3. Following the explanation of the component definition and configuration we depict related work in section 4. This paper concludes with future research directions and an evaluation of the approach in section 5.

2 Defining Component Models

In this section we formalise the component model representing hierarchical process elements. Based on this formalisation, it is possible to derive actual configurations of a process and verify this configuration. The component model is specified using first-order logic. Though this formalisation requires additional initial effort to specify component models, it provides two fundamental benefits. First, it allows to define the used concepts in unambiguous way due to the formal defined semantics of first-order logic. Second, and even more important for practical applications, it allows for easy extension and adaption to domain specific facts. Due to space limitation, we present only the most relevant concepts of our component model. Further details and formalisations are specified in [4].

A component model is represented as a 6-tuple $M = (C, K, G, card, L, T)$:

- C is a finite, non-empty set of *components*,
- K is a finite, non-empty set of *connectors*,
- $G \subseteq (C \times K) \cup (K \times C) \cup (K \times K)$ is a set of arcs constituting an acyclic *configuration graph* representing *hierarchical dependencies*,
- $card : K \to \mathcal{P}(\mathbb{N} \times \mathbb{N})$ is a mapping from connectors to *cardinalities*,
- L is a finite set of *logical dependencies*,
- T is a finite set of *temporal dependencies.*

In the following subsections, we provide further details for the individual constituents of the component model based on a simplified example inspired by a real world application. The example describes services around assembly and maintenance of photovoltaics installations. An organisation provides installation services where customers can choose between delivery with self-assembly and delivery with assembly services. However, if users chose assembly services there are two constraints to satisfy. First, customers must obtain delivery service, too. Second, assembly can only be executed after delivery. For existing photovoltaics installations, customers can select maintenance services consisting of on-site maintenance, remote maintenance, and cleaning. Finally, it is possible to obtain monitoring services for evaluating the performance of the photovoltaics installation. Monitoring consists of recording, customer-specific performance analysis, and comparison with other installations. Due to hardware requirements, customers choosing performance analysis have to chose remote maintenance, too. Finally, comparison services needs recording services.

Based on the description of the example, it is possible to identify process components. To shorten formulae presented in the following, we use component identifiers. The process consists of the following components: *Photovoltaics* (C_1) representing the complete process, *Installation* (C_2), *Maintenance* (C_3), and *Monitoring* (C_4) to represent the three main services. *Delivery* (C_5) and *Assembling* (C_6) are installation services. *On-Site Maintenance* (C_7), *Cleaning* (C_8), and *Remote Maintenance* (C_9) are maintenance services. *Comparison* (C_{10}), *Analysis* (C_{11}), and *Recording* (C_{12}) are monitoring services. Thus, we have the set of components $C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}\}$. Fig. 1 on page 127 gives a visualisation of the component model containing all details described in the following sections.

2.1 Hierarchical Dependencies

The main objective of the component model is to describe complex processes using single, less complex subcomponents. This is achieved by using the configuration graph that contains hierarchical dependencies between components.

Decomposing components into more fine grained subcomponents represents process refinement. We use connectors to specify the type of hierarchic relation between components. For configuration reasons, we prohibit direct linking of components, i.e. components can only be linked with each other using connectors. Therefore, we have the set of connectors $K = \{K_1, K_2, K_3, K_4\}$ with K_1 linking the overall component C_1 with the main services C_2, C_3, and C_4. K_2, K_3, and K_4 link the respective subcomponents with their children. This results in the following configuration graph G.

$$G = \{(C_1, K_1), (K_1, C_2), (K_1, C_3), (K_1, C_4), (C_2, K_2), (C_3, K_3), (C_4, K_4), \\ (K_2, C_5), (K_2, C_6), (K_3, C_7), (K_3, C_8), (K_3, C_9), (K_4, C_{10}), (K_4, C_{11}), (K_4, C_{12})\}.$$

A connector can be assigned with an arbitrary amount of cardinalities specifying valid configuration choices using the mapping *card*. Each cardinality specifies the minimal and maximum number of subnodes that needs to be selected during configuration. If a connector is assigned with more than one cardinality, the cardinalities are linked with each other using logical ORs. Therefore, only one of the cardinalities must be satisfied during configuration. In our example, we have to choose at least one subcomponent of every component (e.g. if monitoring is chosen, at least one of comparison, analysis, and recording must be chosen as well). Therefore, we have the cardinalities $card(K_1) = \{(1, 3)\}$, $card(K_2) = \{(1, 2)\}$, $card(K_3) = \{(1, 3)\}$, and $card(K_4) = \{(1, 3)\}$.

2.2 Logical Dependencies

The specified graph with different types of connecting nodes and cardinalities defines the dependencies between components that are directly interrelated by the given graph. Additionally, dependencies have to be specified for components that are neither children nor parents of other components. These dependencies are necessary to make statements like *when choosing component A during a configuration, component X has to be chosen as well* or *when choosing component B during a configuration, component Y must not be chosen.* As these dependencies represent logical restrictions for configuration, they are called *logical dependencies* . Such logical dependencies are necessary to support error-free service configurations [5].

For specifying these dependencies, expressive methods like first-order logic are best suited. As the application of first-order logic is quite complex, typical rules can be specified that are based on first-order logic, but are at the same time applicable for users that are unfamiliar with first-order logic. The following rules in Table 1 are examples that can be used when specifying logical dependencies between service modules.

The dependencies given in Table 1 are *hard* dependencies, i.e. they must be satisfied in valid configurations. Besides this, it is also possible to specify *soft* dependencies, e.g. component A is an *alternative* of component B. For establishing new logical dependencies, it is possible to combine existing ones, e.g. components A and B are *exclusive alternatives* of each other can be defined as

Table 1. Logical dependencies between components

Rule	Formalisation	Explanation
Requirement	$requires : C \to C$ $requires(A) = B$	Component A requires component B: if component A is chosen during configuration, component B must be chosen as well.
Dependency	$depends : C \to C$ $depends(A) = B$	Component A depends on component B: if component B is not chosen during configuration, component A must not be chosen as well.
Prohibition	$prohibits : C \to C$ $prohibits(A) = B$	Component A prohibits component B: if component A is chosen during configuration, component B must not be chosen and vice versa.

$prohibits(A) = B$ and $alternative(A) = B$. A selection of additional dependencies in the domain of Product-Service-Systems is presented in [2]. To validate the applicability, we have implemented the dependencies as Prolog rules[1]. Based on this representation, it is possible to verify if a configuration (i.e. a set of selected components) satisfies the given hard dependencies of a component model and whether additional soft dependencies are available. The semantics of the given logical dependencies are defined during configuration (see section 3.2).

Three hard logical dependencies exist in the photovoltaics example. If customers choose assembly service (C_6), they have to choose delivery (C_5), too. Comparison (C_{10}) needs recording (C_{12}). Finally, performance analysis (C_{11}) requires remote maintenance (C_9). This results in logical dependencies $L = \{requires(C_6) = C_5, requires(C_{10}) = C_{12}, requires(C_{11}) = C_9\}$. Especially the requirements relation between performance analysis and remote maintenance is notable. In traditional process models it is often only possible to declare relations in one branch, e.g. relations between the subcomponents of maintenance.

Though we provide the opportunity to define logical dependencies this feature should be used sparsely. It both impacts the readability and comprehensibility of component models and adds to the complexity of configuration. As Thum et al. have shown for feature models these dependencies are especially hard to understand in editing models [6]. Some ideas how to eliminate non-hierarchic constraints in the domain of feature models are given in [7,8]. These concepts should be applicable in the domain of process components, too.

2.3 Temporal Dependencies

Since the components in our model represent process activities, it is necessary to specify the possible execution order of these activities, e.g. sequential execution, parallelisation, and synchronisation. Concerning the specified graph, it is, therefore, not enough to display only logical dependencies between components, but also to display temporal dependencies. These temporal dependencies define whether a component has to be performed before or after another component.

[1] https://sourceforge.net/projects/kpstools/

Using this information, it will be easier to implement finally the whole process out of the chosen component. The instantiation of a process (specifying which service module has to be executed at which time) has to take into account the specified temporal dependencies.

For keeping the graph as flexible as possible, temporal dependencies can be specified by using a declarative approach (as opposed to a procedural approach). Such approach has been proposed by Aalst and Pesic [9]. The application of the linear temporal logic (LTL) [10] would offer the most flexible and expressive way of specifying the temporal dependencies. Nevertheless, this approach is not applicable for users that are unfamiliar with LTL. Therefore, a set of rules can be specified that covers most of the temporal dependencies. These rules are, on the one hand, understandable for non-professionals and, on the other hand, based on the LTL which allows functionalities like model checking or simulation. Table 2 shows a selection of possible temporal dependencies. It is necessary to note that the exact semantics of temporal dependencies is formalised during configuration (see section 3.3). Furthermore, the examples shown in Table 2 are not independent of each other, i.e. precedence can be specified in terms of succession and vice versa.

Table 2. Temporal dependencies between components

Rule	Formalisation	Explanation
Precedence	$before(A) = B$	In all configurations containing components A and B, it is necessary to execute component B before component A.
Direct Precedence	$iBefore(A) = B$	In all configurations containing components A and B, it is necessary to execute component B directly before component A.
Succession	$after(A) = B$	In all configurations containing components A and B, it is necessary to execute component B after component A.
Direct Succession	$iAfter(A) = B$	In all configurations containing components A and B, it is necessary to execute component B immediately after component A.

In the photovoltaics example one temporal dependency occurs, i.e. before assembling (C_6) the installation has to be delivered (C_5). Therefore, the set T of temporal dependencies is defined as follows: $T = \{before(C_6) = C_5\}$. It is necessary to mention that all components that are not linked with temporal dependencies are independent from each other. That means, they can be executed in parallel, e.g. while performance analysis it is possible to clean the photovoltaics installation. In a complex real world example there would be much more temporal dependencies. For example, cleaning and on-site maintenance may be performed by the same individuals. Therefore, only one of the activities can be executed at one time. Furthermore, installation certainly needs to be completed before maintenance.

2.4 Graphical Representation

For comprehensibility reasons, we provide a set of notational elements for the graphical representation of a component model. Components are depicted as rectangles and connectors as circles. Hierarchical dependencies between these elements are represented using directed arrows. To represent logical dependencies, we use directed, dotted arrows where $A \rightarrow B$ means that selecting component A also requires selecting component B. Temporal dependencies are represented using dashed lines, where $A \rightarrow B$ means that component A must be performed before component B. Fig. 1 shows the photovoltaics example using the defined notational elements.

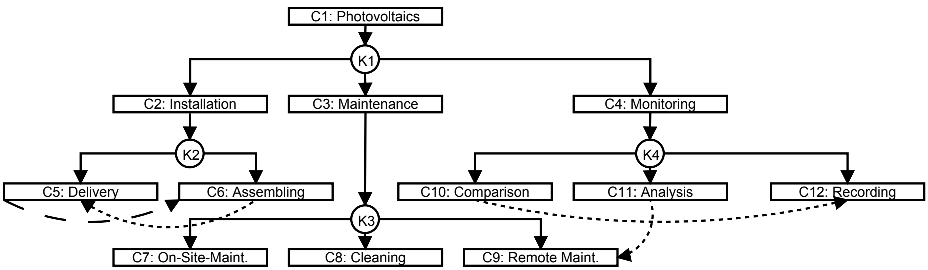


Fig. 1. Component model for a photovoltaics installation

3 Configuration

Using the hierarchical, logical, and temporal dependencies between components, it is possible to generate a customer specific configuration. A configuration is a set of components selected from the given portfolio defined by the component model. Due to the formalisation of the model, it is possible to validate whether selected components fulfill the established dependencies. In this section we use the photovoltaics example introduced in the last section and show how to create configurations and how to establish process models based on a given configuration.

Each configuration consists of three distinct steps. First, components are selected based on customer-specific requirements. The selection is restricted by hierarchical dependencies between components, cardinalities of connectors, and hard logical dependencies between components. Second, soft logical dependencies are evaluated and presented as configuration alternatives. Third, an actual configuration is transformed into a procedural process representation using the temporal dependencies. This representation can for example be imported in workflow management systems to guide through the process.

3.1 Component Selection and Cardinality Evaluation

During configuration, the mapping $s : C \rightarrow \{0, 1\}$ represents whether a component is selected or not. A configuration is defined as the set of selected components, i.e. the set $Configuration = \{c|c \in C \wedge s(c) = 1\}$ contains all selected components. At the beginning of the configuration process, none of the nodes is selected, i.e. $\forall c \in C : s(c) = 0$.

To define selection and connector semantics during configuration we need to introduce the mapping $p : C \cup K \to \mathcal{P}(C \cup K)$ to identify postnodes (i.e. succeeding nodes) of a node in the configuration graph. This mapping is defined as $p(n_1) = \{n_2 \in C \cup K : \exists e \in G : e = (n_1, n_2)\}$.

Now we define that succeeding nodes of an unselected node are not selected, too. Thus, we prohibit to select subcomponents without selecting the respective superior component: $\forall n_1 \in C \cup K, \forall n_2 \in p(n_1) : s(n_1) = 0 \to s(n_2) = 0$.

On the opposite, all succeeding nodes of a component are selected. Since components can only be followed by connectors, we include these connectors in the configuration: $\forall n_1 \in C, \forall n_2 \in p(n_1) : s(n_1) = 1 \to s_(n_2) = 1$.

Finally, we have to define connector semantics during configuration. A connector is satisfied if there is a number of succeeding nodes selected fulfilling the interval defined by one of the cardinalities. Therefore, we define the set sp that contains all selected succeeding nodes of a connector k.

$$\forall k \in K : sp(k) \subseteq p(k)$$
$$\forall k \in K, \forall n \in sp(k) : s(n) = 1 \wedge n \in p(k)$$
$$\forall k \in K : \exists (m, n) \in card(k) : m \leq |sp(k)| \leq n.$$

3.2 Evaluate Logical Dependencies

As stated in section 2.3, logical dependencies restrict possible configurations. Therefore, it is necessary to assign formal semantics to given dependencies. In Table 1, we defined the dependencies *requirement*, *dependency*, and *prohibition*.

A requirement $requires(c_1) = c_2$ between component c_1 and c_2 states selecting component c_1 leads to the selection of component c_2. This can be formalised as follows: $\forall c_1, c_2 \in C : s(c_1) = 1 \to s(c_2) = 1$.

If component c_1 depends on component c_2 ($depends(c_1) = c_2$), it is not possible that component c_2 is not selected while component c_1 is selected: $\forall c_1, c_2 \in C : s(c_2) = 0 \to s(c_1) = 0$.

Finally, prohibition of components c_1 and c_2 ($prohibits(c_1) = c_2$) permits both components being selected at the same time: $\forall c_1, c_2 \in C : (s(c_1) = 1 \to s(c_2) = 0) \wedge (s(c_2) = 1 \to s(c_1) = 0)$.

Based on the semantics of the logical dependencies, it is possible to establish valid configurations. However, as can be seen from the specification, it is also possible to establish models that are not satisfiable. For example, the logical dependencies $requires(A) = B$ and $prohibits(A) = B$ must not occur in the same model. However, satisfiability of models is not in the focus of this work. The interested reader is referred to [2] for a detailed overview about interactions between different logical dependencies.

3.3 Evaluate Temporal Dependencies

After a configuration that satisfied the given cardinalities and logical dependencies has been established, it is possible to represent the components as a procedural process. This representation can be used as input for workflow management systems that guide through a configured process. Therefore, it is necessary to arrange the components according to the given temporal dependencies.

As stated in section 2.3, we specify temporal dependencies using LTL. In the following, we first show how the temporal dependencies of Table 2 are enriched with formal semantics. Based on this semantics, we show an example configuration in its process representation in the next section.

The precedence dependency $before(c_1) = c_2$ states that in a configuration containing both components c_1 and c_2, component c_2 must be executed before component c_2. In LTL terms, this can be represented as the constraint that c_1 cannot be executed until c_2 was executed: $\forall c_1, c_2 \in C : (s(c_1) = 1 \wedge s(c_2) = 1) \rightarrow (\neg c_1 \mathcal{U} c_2)$.

The direct precedence dependency $iBefore(c_1) = c_2$ states that in a configuration containing both components c_1 and c_2, component c_2 must be executed immediately before c_1. This is an extension of the precedence dependency. In addition, it is necessary that the execution of c_1 follows immediately after the execution of c_2: $\forall c_1, c_2 \in C : (s(c_1) = 1 \wedge s(c_2) = 1) \rightarrow (\neg c_1 \mathcal{U} c_2 \wedge c_2 \rightarrow \bigcirc c_1)$.

The succession dependency $after(c_1) = c_2$ states that in every configuration containing both component c_1 and c_2, component c_2 must be executed after component c_1. In LTL terms, this can be represented as the constraint that after the execution of c_1 eventually c_2 must be executed in the future: $\forall c_1, c_2 \in C : (s(c_1) = 1 \wedge s(c_2) = 1) \rightarrow (c_1 \rightarrow \Diamond c_2)$.

Finally, direct succession $iAfter(c_1) = c_2$ implies that immediately after the execution of component c_1, component c_2 must be executed. This can be formalised similar to direct precedence. However, in this case it is not necessary that c_1 is executed before c_2 can be executed: $\forall c_1, c_2 \in C : (s(c_1) = 1 \wedge s(c_2) = 1) \rightarrow (c_1 \rightarrow \bigcirc c_2)$.

3.4 Procedural Process Transformation

With the semantics of the temporal dependencies at hand, it is possible to establish a procedural process model based on a given configuration. For comprehensibility, we show the transformation of a configured component model into a process model using the photovoltaics example. A configuration is established based on customer requirements where different ways of asking for these requirements are possible. For example, [11] shows a dialogue-driven approach to establish configurations. We support configuration decisions by our proposed hierarchical, logical, and temporal dependencies between components. In a typical example, a customer asks for a photovoltaics installation that is constructed by the service provider. In addition, the customer wants to buy the cleaning service and does not want to maintain the installation on her own. Therefore, remote maintenance is necessary. Due to hierarchical (e.g. cleaning is a child component of maintenance) and logical dependencies (e.g. remote maintenance needs the analysis component), we have the following configuration including all necessary components.

$$Configuration = \{photovoltaics, installation, delivery, assembling, \\ maintenance, cleaning, remote - maintenance, monitoring, analysis\}$$

Based on the hierarchic dependencies between components this configuration can be represented as a configured graph shown in Fig. 2. This graph does not contain any connectors because all components are mandatory. Since the graph should only act as a helpful visualisation, we do not formalise its constituents.

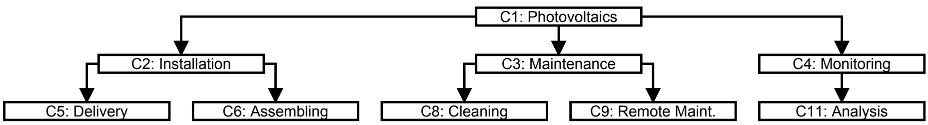


Fig. 2. Configured graph for selected components

In the following, the configured graph is transformed into a procedural process representation using Business Process Model and Notation (BPMN [12]) as the process notation. In doing so, every selected component can be represented as a (possible expandable) activity in the process model. The transformation results in a complete process model according to the logical and temporal dependencies between selected components. By now, there does not exist a complete formalisation of the transformation process. Thus, we give a step-by-step instruction.

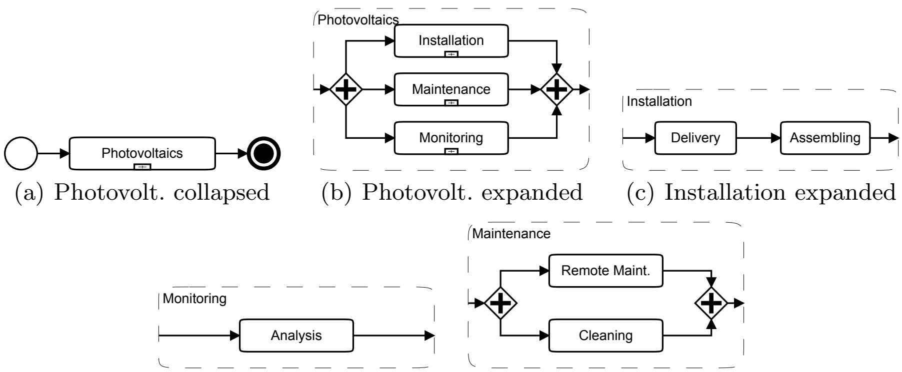
First, a collapsed activity for the complete photovoltaics process is created (Fig. 3(a)). This is to show that we use components as refineable activities in the process model. Thus, it is possible to represent hierarchic process dependencies.

Going down one level in the configured graph, we have the components installation, maintenance, and monitoring. As stated above, all of these components are necessary. Furthermore, there are no temporal dependencies between these components. Thus, we can create an activity for each component and connect them using a parallel gateway when expanding the photovoltaics activity (Fig. 3(b)). Every components of this level is again represented as an activity that can be expanded.

When expanding the installation activity, we have to satisfy the temporal constraint $before(assembling) = delivery$. Therefore, it is necessary that the activity delivery is executed before the activity assembling, i.e. both activities must be in sequential order. Since we only have these two activities in the respective subprocess, we can directly connect both activities with each other (Fig. 3(c)).

The remaining two activities (monitoring and maintenance) are expanded in similar way. In monitoring there is only one activity analysis. Therefore, we do not have to consider any temporal constraints (Fig. 3(d)). The two maintenance activities can be executed in parallel and are thus connected using a parallel gateway (Fig. 3(e)).

The activities in Fig. 3(c) - 3(e) cannot be expanded anymore. Therefore, the transformation is finished resulting in a complete procedural process model.



(a) Photovolt. collapsed (b) Photovolt. expanded (c) Installation expanded

(d) Monitoring expanded (e) Maint. expanded

Fig. 3. Transformation of configuration into procedural process representation

4 Related Work

In our approach we use a component model representation to allow for configuration of complex business processes. Other approaches extend existing process notations with explicit representation of variabilities. For example, Rosemann and van der Aalst extend event-driven process chains (EPCs) to so-called configurable EPCs (C-EPCs) to represent configurable reference models [13]. Therefore, they give a formal definition of C-EPCs and describe how configuration decisions effect resulting process models. Another streamline of research focuses BPMN extensions to display variability, e.g. [14,15,16]. However, a big differences between these approaches and our presented approach lies in the fact that we analyse the whole portfolio of process models of an organisation. Opposing to this, C-EPCS and BPMN extension are on the level of one specific process model.

Feature models are well-known in the domain of software engineering and another feasible approach to represent variability [17]. By establishing a distinct feature model, it is possible to extract variability from the model and to define clearly distinctable feature decision points. In doing so, existing process models can be reused without any changes. It is just necessary to group them according to included features and map the features in the feature model to the respective processes. La Rosa et al. use this approach for reference model configuration by describing variability of a domain and of a process in separate models [18].

Since we also use declarative elements in our component model, approaches for specifying declarative process models are of interest, too. Aalst and Pesic have proposed a comprehensive overview about declarative modelling in [19]. Their work can be used as a source for additional temporal dependencies between components. Furthermore, Soffer and Yehezkel have proposed declarative modelling focusing expression of variability [20]. To broaden the view on temporal dependencies, the work of Lanz et al. can be used as an additional source [21]. They present time patterns that occur in workflow systems, e.g. lags between activities and durations of activities. It is an interesting approach to analyse how these time patterns are related with temporal dependencies.

5 Evaluation and Conclusions

In this work we presented an approach to represent business processes in a hierarchic way. The proposed component model focuses configuration of processes. Therefore, it uses hierarchically structured components that are connected by distinct nodes allowing for specifying semantics of the structure. Additionally, it is possible to assign logical and temporal dependencies between components. By comparing our approach with requirements for configurable reference modelling techniques mentioned in academic literature [13], it is possible to establish a first evaluation and future research directions.

1. Differentiate between run-time and build-time decisions. Our model uses connector nodes to represent build-time decisions, i.e. configuration points of the model that need to be decided before a model is executed. By assigning leafs of the tree with process models, it is possible to allow for run-time decisions, too. Thus, we support both decision possibilities with a clear distinction between them.
2. Support configurations regarding entire processes, functions, control flow, resources, and data. In the current state we only support configuration based on process level. Resources and data are out of focus. However, enriching component descriptions with data and resource information should be possible in future developments.
3. Differentiate between mandatory and optional decisions. Connectors can be initialised with default cardinalities. However, in the current state it is still necessary to select specific components during configurations. In future extensions of the model, default cardinalities can be enriched with the specification of default components that are selected when no explicit decision was made.
4. Differentiate between global and local decisions. Global decisions are based on specific context factors (e.g. country, domain etc.). Currently, it is not possible to map these context factors on decisions. However, an extension of the model includes so-called external variables [22]. Based on these variables it may be possible to define configuration decisions.
5. Differentiate between critical and non-critical decisions. In the current state, we support only non-critical decisions, i.e. every decisions can be re-done and can be changed over time. Thus, it is not possible to distinguish between these two decision types.

6. Depict interrelationships between configuration decisions. Due to the hierarchic representation of the component model, there is a natural configuration order, i.e. if a superior component is not selected, the child components cannot be selected, too. However, there exists no such order for logical dependencies. This is a current weakness of our component model and needs to be overcome in future research.
7. Differentiate between configuration decisions on different levels. Since we do not cover organisational details in our models, this differentiation is not contained in the model. Nonetheless, it is possible that configuration is conducted step-wise. In doing so, different levels can refine a configuration.
8. Relate variation points with additional information. Additional information are not formally defined in the model. However, the definition of connectors may be enriched with an additional information, e.g. an URL. At this URL, configuration information can be placed.
9. Guide the configuration by recommendations and guidelines. It is possible to assign key performance indicators (KPIs) to components [23]. Based on these KPIs, the productivity impacts of configuration decisions can be assessed. Organisations can further use these information to develop configuration guidelines. Additionally, the configuration process can be supported by defining recommendations for components using logical dependencies. Other research approaches promote using questionnaire-based configuration, e.g. [18]. This is possible using our approach, too. In future research we will analyse ways to establish a guided configuration.
10. Make complexity manageable. Due to the modularised, hierarchic structure of our component model, it is possible to separate process modelling from configuration. For example, executive management of an organisation uses component models on a very abstract level to decide about the overall organisation strategy. Functional departments can build on this configuration with their own, refined models.

In future, the consequences of our approach for modelling practice can be evaluated based on two approaches. First, it is possible to establish reference process models based on existing reference processes for existing domains, e.g. SAP R/3 [24]. According to the representation of these reference processes, we will analyse the complexity differences in configuring an existing reference model in comparison to the configuration using our presented approach. Second, we will analyse processes of our industry partners in more details and evaluate them according to their configuration potential.

Since the first-order and linear temporal logic formalisation is not easy to use (especially for non-professionals) we have developed a tool conforming to the notation. The practical applicability of this tool is shown in [25]. However, in its current state the tool cannot transform component into process models. To further enhance practical applicability, we have to analyse how existing process models can be reused (extraction). This is necessary, since organisations often own process repositories consisting of hundreds or even thousands of models [26]. A valid starting point for extraction is to identify mappings between workflow patterns according to [27] and specific component hierarchies.

References

1. Böttcher, M., Klingner, S.: The basics and applications of service modeling. In: SRII Global Conference 2011 (2011)
2. Becker, M., Klingner, S.: Formale Modellierung von Komponenten und Abhängigkeiten zur Konfiguration von Product-Service-Systems. To Appear in: Diensleistungsmodellierung 2012 (March 2012)
3. Pohl, K., Böckle, G., Linden, F., Lauenroth, K., Pohl, K.: Principles of Variability. In: Software Product Line Engineering, pp. 57–88. Springer, Heidelberg (2005)
4. Becker, M.: Formales Metamodell für Dienstleistungskomponenten. Technical report, Universität Leipzig (2011)
5. Heiskala, M., Paloheimo, K.S., Tiihonen, J.: Mass Customization of Services: Benefits and Challenges of Configurable Services. In: Frontiers of e-Business Research (FeBR 2005), Tampere, Finland, pp. 206–221 (September 2005)
6. Thum, T., Batory, D., Kastner, C.: Reasoning about edits to feature models. In: Proceedings of the 31st International Conference on Software Engineering, ICSE 2009, pp. 254–264. IEEE Computer Society, Washington, DC (2009)
7. van den Broek, P., Galvão, I., Noppen, J.: Elimination of constraints from feature trees. In: Thiel, S., Pohl, K. (eds.) Proceedings of 12th International Conference on Software Product Lines, SPLC 2008, Limerick, Ireland, September 8-12, vol. 2(Workshops), pp. 227–232 (2008)
8. Gil, Y., Kremer-Davidson, S., Maman, I.: Sans Constraints? Feature Diagrams vs. Feature Models. In: Bosch, J., Lee, J. (eds.) SPLC 2010. LNCS, vol. 6287, pp. 271–285. Springer, Heidelberg (2010)
9. van der Aalst, W., Pesic, M.: Specifying, Discovering, and Monitoring Service Flows: Making Web Services Process-Aware. BPM Center Report BPM-06-09, BPMcenter.org (2006)
10. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM 32, 733–749 (1985)
11. Tiihonen, J., Soininen, T., Niemelä, Sulonen, R.: A practical tool for mass-customising configurable products. In: Proceedings of the 14th International Conference on Engineering Design, pp. 1290–1299 (2003)
12. Grosskopf, A., Decker, G., Weske, M.: The Process: Business Process Modeling Using BPMN, 1 edn. Meghan Kiffer Pr, Tampa (March 2009)
13. Rosemann, M., van der Aalst, W.: A configurable reference modelling language. Information Systems 32(1), 1–23 (2007)
14. Montero, I., Pena, J., Ruiz-Cortes, A.: Representing runtime variability in business-driven development systems. In: International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, vol. 1, p. 241 (2008)
15. Rogge-Solti, A., Kunze, M., Awad, A., Weske, M.: Business process configuration wizard and consistency checker for bpmn 2.0. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 231–245. Springer, Heidelberg (2011)

16. Santos, E., Pimentel, J., Castro, J., Sánchez, J., Pastor, O.: Configuring the Variability of Business Process Models Using Non-Functional Requirements. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 274–286. Springer, Heidelberg (2010)
17. Batory, D.: Feature Models, Grammars, and Propositional Formulas. In: Obbink, H., Pohl, K. (eds.) SPLC 2005. LNCS, vol. 3714, pp. 7–20. Springer, Heidelberg (2005)
18. La Rosa, M., Gottschalk, F., Dumas, M., van der Aalst, W.M.P.: Linking Domain Models and Process Models for Reference Model Configuration. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 417–430. Springer, Heidelberg (2008)
19. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 1–23. Springer, Heidelberg (2006)
20. Soffer, P., Yehezkel, T.: A State-Based Context-Aware Declarative Process Model. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 148–162. Springer, Heidelberg (2011)
21. Lanz, A., Weber, B., Reichert, M.: Workflow Time Patterns for Process-Aware Information Systems. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 94–107. Springer, Heidelberg (2010)
22. Becker, M., Klingner, S., Böttcher, M.: Configuring services regarding service environment and productivity indicators. In: 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 505–512 (September 2011)
23. Böttcher, M., Klingner, S.: Providing a Method for Composing Modular B2B-Services. Journal of Business and Industrial Marketing 26(5), 320–331 (2011)
24. Curran, T., Keller, G., Ladd, A.: SAP R/3 business blueprint: understanding the business process reference model. Prentice-Hall, Inc., Upper Saddle River (1998)
25. Klingner, S., Böttcher, M., Becker, M., Döhler, A.: Managing complex service portfolios. In: Ganz, W., Kicherer, F., Schletz, A. (eds.) RESER 2011 Productivity of Services NextGen - Beyond Output/Input (September 2011)
26. Dumas, M., García-Bañuelos, L., Dijkman, R.: Similarity Search of Business Process Models. IEEE Data Engineering Bulletin 32(3), 23–28 (2009)
27. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow Patterns. Distributed and Parallel Databases 14, 5–51 (2003)

Identifying and Classifying Variations in Business Processes

Fredrik Milani, Marlon Dumas, and Raimundas Matulevičius

Institute of Computer Science, University of Tartu, Estonia, J. Liivi 2, 50409 Tartu, Estonia
{milani,marlon.dumas,rma}@ut.ee

Abstract. Many business processes exist not as singular entities but rather as a plurality of variants that need to be collectively managed. The spectrum of approaches for managing collections of process variants range from capturing all variants in a large consolidated model, down to capturing each variant as a separate model. Most of these approaches are built on the assumption that the variation points and variation drivers are given as input. The question of how process variation is elicited and conceptualized in the first place has received relatively little attention. As a step to filling this gap, this paper puts forward a framework for identifying and classifying variation drivers in business processes. We apply the framework on two collections of process models: one consisting of a collection of process models implicitly clustered along product type and the other one along market type. In both cases, the framework allowed us to identify and to classify additional variation drivers that were not evident from the initial clustering.

Keywords: Business Process Variant, Variation Driver, Variation Point.

1 Introduction

Every organization, be it non-profit, governmental or industrial, has sets of business processes through which value is produced. Many of these business processes will have variations [4, 8]. One way of managing variations is to treat each process variant as a distinct process, and to model each variant separately. Such a *fragmented-model approach* creates redundancy and inconsistency [8]. On the other hand, modeling and managing multiple variants together in a *consolidated-model approach* leads to complex processes that may prove difficult to understand, analyze and evolve [8]. Striking a tradeoff between maintaining each process variant separately versus collectively in a consolidated manner is still an open research question [4].

To address this tradeoff, various approaches have emerged [8, 11, 17] for annotating variations in a business process models with meta-data so as to facilitate efficient management of the process variations from different perspectives. For example, Rosemann *et al.* [17] present an extended EPC language (namely c-EPC) for managing variations in reference models. Building on the c-EPC, La Rosa *et al.* [11] have developed a method for merging multiple models of process variants into a consolidated process model. Hallerbach *et al.* [8] propose an alternative approach to managing

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 136–150, 2012.

large numbers of variations in process models, namely PROVOP (PROcess Variants by OPtions). Berger *et al.* [16] proposes an approach whereby an organization can create a generic organizational reference model that then is specialized and customized by different units of the organization as a method of managing variations.

The above studies work under the assumption that the variation points and the drivers (or "root causes") of variation are given as input. More generally, while the question of representing variations in business process models has been extensively studied [13], the question of how variation can be elicited and conceptualized in the first place has received little attention.

In this setting, this paper addresses the following research question: *How can variation points and their drivers be identified from a given collection of process models?* In order to address this question, we propose a framework to systematically identify both explicit and implicit variation in a collection of process models. The framework is built on a classification of variation drivers that allows analysts to ask a series of questions that lead to the identification of variation drivers.

The rest of this paper is structured as follows. In Section 2, we develop a framework for variation drivers followed by examples that show their existence in practice. Then, in Section 3, we illustrate the potential usefulness of our framework by applying it on two collections of process models. In Section 4, we review related works and finally we present our conclusions in Section 5.

2 Framework for Variation Drivers

2.1 Definitions

A *business process*, as defined by Weske [25], is a coordinated performance of a set of activities that aim at fulfilling a certain predefined *outcome* or business goal. A *business process model* is a representation of a particular business process, which expresses the relationships and restrictions of the activities of the process, using a set of notational techniques [25]. A business process, captured as a business process model, may have multiple possible inputs or multiple possible outcomes that are perceived to be similar (but not identical) by a business analyst and that have a visible effect on the way the process is performed. For example, an insurance company would typically perform the process for managing claims differently depending on whether it concerns a personal, vehicle or property claim [4]. These different processes that have similar inputs and outcomes, can be seen as variations of a single process. In this case, these processes are referred to as *variants* [7].

When analyzing a collection of process models, different analysts might choose to focus on different aspects or levels of granularity of the process and thus recognize different variants in the process. Our framework does not provide the analyst with a prescriptive definition of what constitutes a process variant. It will be the choice of the analyst to determine what constitutes a process variant. For example, the analyzer will choose whether to treat the processes for handling personal, vehicle and property claims as three different variants or a single process. Our framework does not prescribe this choice but builds on top of the set of variants chosen by the analysts.

Given a process model or a collection of process models capturing a family of process variants, there will necessarily be points at which a choice is made between multiple branches. For example, when using the Business Process Model and Notation (BPMN), such choices may appear in the form of exclusive gateways (a.k.a. XOR-splits), inclusive gateways (a.k.a. OR-splits) or other types of split gateways. Such points are hereby called explicit *branching points*. Another (implicit) type of branching point occurs when a choice is made between instantiating one process model versus another one – for example instantiating a process model for handling personal claims versus instantiating a process model for handling vehicle claims.

In the proposed framework, each branching point corresponds either to a *variation point*[1] or a *decision point*. A branching point is a variation point if different branches of this point can be attributed to different process variants. A branching point is not a variation point if its outgoing alternatives all belong and are confined within the same variant or if the branches lead to different processes that are not considered to be variants of one another. In such cases, the branching point is labeled as a decision point.

A *variation driver* (henceforth *driver* for short) is a parameter or criterion that is used at a variation point to distinguish between its branches. A *variation option* is a possible option that exists at a variation point. Concretely, a variation option is a value or range of values of the variation driver associated to a variation point.

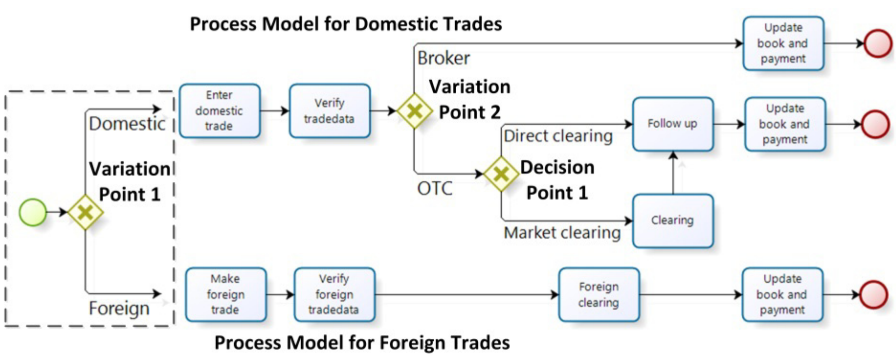


Fig. 1. Examples of definitions

Fig.1 shows two models for the processes of equity trading (domestic or foreign equity). The process models for domestic and foreign equity capture two variants of the equity trading process. The process model for domestic equity covers two variants: one for trades via a broker and another for trades made over-the-counter (OTC).

Seen collectively, these two process models contain three branching points. The first branching point (variation point 1) is the one where a choice is made between instantiating the "domestic equity" process model or the "foreign equity" model. This branching point is an example of an implicit *variation point* as its branches lead to

[1] Our definition of variation point is not to be confused with variation points in the context of configurable process models [17, 21].

different but similar outcomes (trading an equity). In Fig.1, this variation point is represented inside a dotted rectangle. Importantly, the XOR-split gateway inside this rectangle does not exist in the process models. We added it to the figure for the sake of making the branching point visible. In reality, the branching point exists merely by virtue of a choice being made between instantiating two alternative process models.

Within the process model for domestic equity trading, there is an *explicit variation point* (variation point 2) where a choice is made between trading using a broker or OTC. Within the process model for clearing a domestic equity, a third branching point can be found with two branches ("direct" vs. "market clearing"). This is a *decision point* as both alternatives belong to the same process variant, that is, they both lead to the same outcome from the perspective of the equity trading process.

The *variation driver* at variation point 1 is "domestic vs. foreign", and at variation point 2 it is "broker vs. OTC". At variation point 1, "domestic" is a *variation option* and "foreign" is the second variation option. Similarly, at variation point 2, broker is the first variation option and OTC is the second.

The meta-model of our framework, shown in Fig. 2, gives an overview of the above presented definitions. The top-level concept in this meta-model is that of a process, a process being that of a collection of logically related activities. It should not to be confused with a process instance which is one specific execution of a process, nor should it be confused with a process model, which is a specific way of describing a process or part of a process.

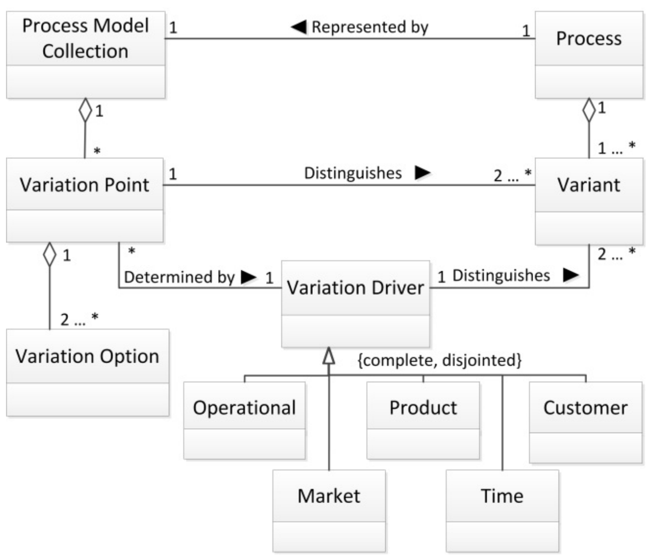


Fig. 2. Meta-model of our framework

A given process, constituting of one to many variants, is represented by a collection of process models. Within a collection of process models, there are variation points, each of which will have at least two variation options. The variation point has one variation driver. It may happen that one can identify multiple variation drivers in a variation point but if so, these variation points could be split into two or more consecutive variation points so that each of them will have only one variation driver. We therefore assume that each variation point has a single variation driver. It should be noted that in the meta-model, the concept of variation drivers has several sub-classes (complete and disjointed) that will be explained in the following section.

2.2 Theoretical Foundation of Variation Drivers Framework

The theoretical base of our framework is built on the framework for business architecture layer of enterprise architecture presented by Rummler and Ramais [18]. In their framework, organizations are viewed as systems whose purpose is to produce value and these systems exist within a larger "Super-System". This super system is the context within which an organization operates and the reality to which it must adapt itself to in order to survive. According to this framework, the environment, resources, stakeholders, markets, customers and competitors influence organizations. Within the context of these external variables, all organizations create an output by procuring resources in order to manufacture a product or a service. These products and/or services are then brought to a market place where the customers, those who need or wish to consume the outputs manufactured, can buy the products and/or services. In some situations, an organization might wish to adapt its processes depending on certain parameters in its external environment such as for example tourist season.

Rummler and Ramais framework can also been viewed as a map of factors that an organization needs to relate to in conducting its business. An organization interprets its business environment and chooses to respond to it in ways that they perceive to ensure competitive advantage. Therefore, these factors have an impact on their business processes. As such, these factors, combined with how they decides to manage them, are causes of variations in business processes. The premise of our framework is that these decisions will manifest themselves in business processes as variation points.

Rummler and Ramais framework, on its own, does not include an explicit classification of variation drivers occurring in business processes. But by overlaying the W-questions (how, what, where, who and when) on Rummler and Ramais framework (Fig. 3), we obtain a system for assessing and orthogonally classifying variation drivers.

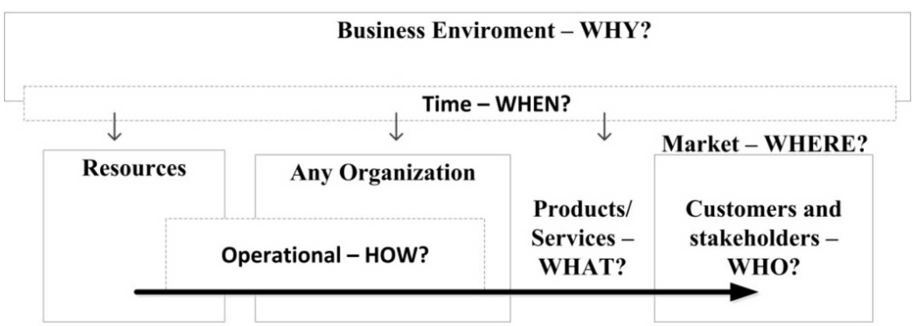


Fig. 3. A framework for business variation drivers

The overlay between Rummler and Ramais framework and the W-questions is as follows. Organizations have a set of processes to procure resources and manufacture (*how*) output (*what*) that they bring to a marketplace (*where*) for customers (*who*) to buy. Finally, organizations sometimes (*when*) adapt their processes to a specific external situation in order to remain efficient throughout the value chain.

Rummler and Ramais framework include “competitors” as a factor but we have excluded it in our analysis since an organization will in principle not design processes that are dependent or driven by competitors – although design choices made by an organization might be driven by competitors.

2.3 Driver Elicitation Method

The driver elicitation method (as depicted in Fig. 4) begins with identifying all branching points of a given process model. Once a branching point has been identified, the outgoing alternatives are examined to assess if they lead to different but similar outcomes, that is, classification of the branching point as either decision or variation point. Once a variation point has been identified, its variation options are identified from which we can identify its variation driver. Continuing the analysis, we identify which W-question corresponds best to the variation driver and then orthogonally classify them accordingly. The task beginning from classify branching points to classify variation driver, are repeated for each branching point in the collection of process models. It should be noted that in some cases, certain variants might be known before the analysis start, and in other cases the variants are discovered during the variation elicitation analysis. It might even be a combination of these two, that is, some variants are known at the start and some are discovered during the analysis.

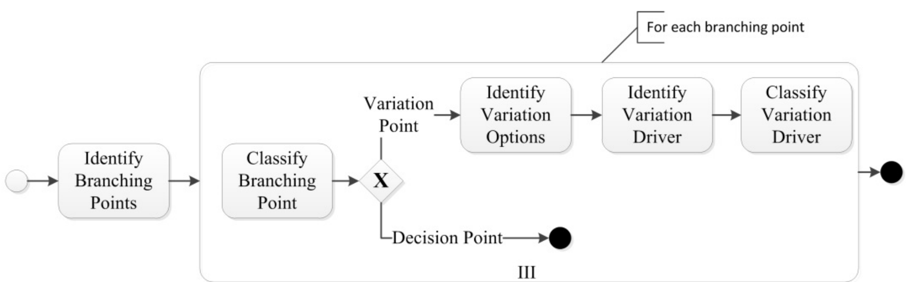


Fig. 4. Driver elicitation method process

2.4 Classes of Drivers

The above analysis leads us to recognize five orthogonal categories of variation drivers, namely: operational (*how*), product/service (*what*), market (*where*), customer (*who*) and time (*when*).

Operational Variations

Every organization has designed processes to manufacture what will bring value to its customers. Although traditionally manufacturing processes has been referring to the production of physical products, we consider manufacturing to cover services as well in accordance with the broader definition proposed by Dalek & Carlsson [3].

Examples: the processes of Dutch municipalities has been investigated by Buijs *et al.* [1] who compared the processes for building permit and housing tax in four different municipalities. Gottschalk *et al.* [6], using the same data set, compared the process of acknowledgement of an unborn child. Buijs *et al.* chose those municipalities that had the same type of information system and yet, each of them had different processes for building permit and housing tax. Gottschalk *et al.* chose municipalities that varied from each other in regards to information systems used. In these cases [1, 6] the municipalities are offering the same service but have chosen to manufacture them differently. These variations exist as the municipalities have a certain degree of autonomy and are free to choose how to design these processes and what system solutions to use. The variations in this example are manufacturing driven variations as in choosing between two variants, the answer to the W-question "*how*" provides guidance as to which variant to follow.

Product/Service Variations

The primary purpose of any given organization is to produce value in the form of products and/or services. As firms offering multiple products/services are ubiquitous, the field of multi-product competition and product differentiation strategies has been and is being studied extensively as Manez and Waterson show [14]. Offering several products or a set of products with differing features is therefore a driver of variations in business process models.

Examples: La Rosa *et al.* [12] presents an example from the film industry. In this example, there are two variants of the post-production process of a film. The first variant is for when shooting the film "on tape" and the second for when the film is shot "on film". These two variants follow the same path until a certain point where the variation occurs. When the case of "on tape" is relevant, there occurs "online editing" and when the film is "on film", "negmatching" takes place. This variation point is driven by product/service as the product, in this case "*what*" kind of film (tape or film), determines which path the next step will follow. Van der Aalst *et al.* [22] uses an example of travel requisition. This process covers two variants, one for international and one for domestic travel. If it concerns an international travel, the process involves requesting quote, preparing travel requisition form, submitting for approval, approval or rejection of the request, possible modifications or updates of the request, and re-submission or cancellation. For domestic travels, the process includes asking for quote and reporting the request to the administration. This variation is driven by product/service as the question *"what"* kind of travel suggest which of the two variants is relevant.

Market Variations

The concept of dividing a market that an organization targets with its products/services (market segmentation) has been studied extensively [24]. Market segmentation can be defined [2] as dividing a heterogeneous market into relative homogenous segments. Organizations can and do segment their markets differently in accordance with their own needs and preferences [5]. The many different methods and the various basis for market segmentation studied [24], illustrates the variety of organizational flexibility in market segmentation strategy implementation. As organizations can divide their markets into different segments and approach them differently, their business process models will have market driven variations. In these variation points, the W-question that is most relevant is "*where*".

Examples: Hallerbach *et al.* [8] describe the variations of a process for vehicle repair. One of the variations in this process depends on the country. If it is in country 1, the process is described as reception, diagnosis, repair and hand over. The same process in country 2, has a "final check" before the vehicle is handed over to the owner. This variation, as explained in the article, is due to a legal requirement in country 2 stating that the vehicle must be checked before handed over to the owner. This regulation does not exist in country 1 and therefore there is a variation. This is a market driven variation as, the answer to the W-question "*where*" provides the answer as to which variant is relevant.

Customer Variations

Organizations produce products/services that bring value to customers but not all customers are the same. Customers can therefore be segmented, that is, divided into various subgroups based on certain attributes and characteristics, and subsequently treated or managed differently [21]. An example of customer segmentation taken from the airline industry is that first class customers are treated differently (have different processes for) compared to economy class customers [20]. Organizations have different processes in offering the same product, to different types of customers. Due to this, the customer is a driver of variation in business processes.

Examples: Kleijn & Dekker [9] writes about the inventory of rotables (aircraft part that can be repaired if it breaks down) where a major airline has founded a company to service them with the inventory of such aircraft parts. This company also provides other customers (airlines) with the same service. There is however variation in the process depending on *"who"* the company is dealing with. If it is the major airline that founded the company, there is an agreement that parts are to be supplied within 24 hours in 95% of the times. Similar, but not identical procedure, exist with other airlines that has an agreement with the company. Airlines without an agreement can also use their services. In such cases, the decision is made, depending on various reasons, to sell, loan or exchange the part. The variations in this process are caused by *"who"* the customer is and therefore it is a customer driven variation.

Time Variations

The above presented variation drivers share the commonality of being independent of differing requirements that may occur in the environment of the process. These process variations do not include the possibility of different execution paths depending on extrinsic events or requirements. If, at a variation point, the path of execution is determined by an external factor, we define it as a time driven variation. At such variation points, the relevant W-question to determine the next step in the path of execution is "*when*". We make no distinction between variations whose execution is predefined according to a set of conditions (design time) and variations that has execution alternatives, dependent on situations occurring at runtime [10].

Examples: An Australian insurance company [23] has call-centers to manage incoming claim calls that are then routed to the back-office that manages the claims. The call-centers have an even flow of calls coming except for during the Australian storm season. During the storm season, the number of calls increases from average 9000 to as much as 20000 calls per week causing significant burden on not only the call-centers but also on the back-office who has to evaluate and manage the claims. In order to manage this increased burden, the insurance company has created an "event-based response system" [23], based on the severity of the storms. For each category of storm severity, there is a specific process. There are therefore variations in the process depending on if it is storm season and how severe the storm is (four categories). The variant to be executed is dependent on "*when*" (storm season or not) and also on "*when*" the storm is of what category, thus making it a time driven variation.

3 Validation

As a preliminary validation, our framework was applied on two collections of process models. Our first collection of process models is from a full-service (retail and commercial) bank, operating mainly in the Nordic markets. Our second collection is from a governmental agency providing an array of services related to land management (including maps and satellite images). By analyzing the processes related to the back-office processing of equities in the bank case, and by analyzing processes related to managing document processing in our second case, we seek to show that our framework can be applied to elicit variation points and variation drivers. Furthermore, we show that the variation drivers can be classified orthogonally. In other words, the research questions are: (1) can our framework be applied for identifying variation points and to elicit their drivers, and (2) can our framework be used to orthogonally classify the drivers at each variation point?

3.1 Background

Our first case is from a Nordic bank. The bank covers the entire spectrum of banking products such as retail banking, life insurance, and investment banking with more than 700 branches in northern Europe. This case covers the processes involved in equity trading services in one of its subsidiaries. The collection of processes covers the back-office operations of domestic and foreign equity trading. The collection of processes in the bank case consists of 8 top level processes that are considered to be variants (by our definition) of one another. Each of the 8 top level process models can be decomposed into sub processes, leading to a total of 30 process and sub process models. The collection of 8 top level process models is divided along domestic and foreign equities. In other words, domestic versus foreign equity trading is implicitly recognized as the main variation driver.

Our second collection of process models is from a governmental agency dealing with various issues related to land ownership and survey information. This case concerns management of documentation processing. There are 9 top level process models and additional 15 sub process models. In total, this collection is comprised of 24 process models. These process models cover the business processes of two geographical areas. The explicitly recognized variation driver is therefore market (two geographical areas). The differences in the business processes of these two geographical areas have been captured in the process models using annotations.

Our cases, together, consist of 17 top level process models and 37 sub process models making it a total of 54 process models.

3.2 Analysis of the Collection of Process Models Using Our Framework

In analyzing the data, the first step was to identify the variation points in order to identify all the variants in the consolidated processes. Using our definition of variation points, all branching points, were analyzed and designated either as a variation point or a decision point. This was achieved by identifying each variation point by assessing if the outgoing branches of that point belonged to different but similar outcomes. If the different paths stemming out from a candidate of variation point are considered to belong to the same variant, it was classified as a decision point.

Once a variation point had been recognized, we were able to identify the parameter that distinguishes between the variants at each variation point. Using the framework, we could assign each variation driver into our classification of drivers (i.e. operational, product, market, customer or time) by identifying which W-question best would correspond to the variation driver.

For illustration (Fig. 5), we consider a sub-process model for calculation of fees. We begin by identifying all XOR splits in the process model. We find the first one occurring just after the process called “Get Product Details”. As the outgoing branches can be considered to be variants (both leading to similar outcome but in a different way), we define it as a variation point. The variation driver is “Counter or Online Customer” and the variation options are identified as “Counter” and “Online”. That is, at this variation point, the next step of the process model is dependent the criterion of being counter or online customer. We find the W-question “who” to be the best match. Identifying the W-question “who” allows us to classify it as a customer variation. The second XOR split is defined as a decision point. At this point, the question of it being priority or not determines the next step in the process model. However, we see that both alternatives are within the same variant, as they lead to the same outcome. Therefore this point is classified as a decision point.

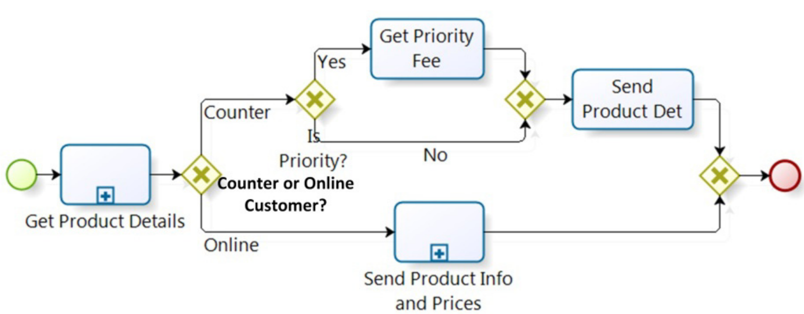


Fig. 5. Example of eliciting variation point and driver in a process model

3.3 Findings

The implicit variation driver in the collection of processes for the processing of equities was along the product, which was domestic versus foreign equity. We did not identify any additional variants from the collection of 8 top level process models. However, our analysis identified an additional 6 implicit variation drivers in the process models making it a total of 7 variation drivers. The additional implicit drivers identified can be labeled as Counterpart type and Execution type (Table 1).

Table 1. Analysis of variation drivers in the bank case

Product (what)	**Customer (who)**			**Operational (how)**		
Equity Type	**Counterpart Type**			**Execution Type**		
Domestic vs. Foreign	*Own vs. Custody*	*Own vs. No Custody*	*Custody vs. Without*	*Exchange vs. OTC*	*Exchange vs. Broker*	*OTC vs. Broker*
2	3	1	1	2	1	1
2	***4***			***4***		

By counterpart type is meant variation drivers determined by what kind of counterpart or customers the trades are being made with. The types identified are “Own” (when the bank is making a trade for itself), “Custody” (when the bank is making a trade on behalf of a client who has a custody service agreement) and “Without” (when the bank is making a trade for a client who does not have a custody service agreement). Execution type refers to how the trade is made. It could be “Exchange” (when the trade is made over the regulated domestic exchange stock market), “OTC” (when the trade is made as a bi-lateral agreement between two parties outside the exchange) or via a “Broker” (when an intermediary is used to make a trade). These could then be classified into three different classes of variation drivers.

It is noteworthy that our input was organized according to the variation driver that had the fewest occurrences (Domestic vs. Foreign). Counting, we found that equity type was responsible for 2 occurrences of variations, whereas counterpart and execution type caused 4 variation points. This indicates that our framework could be used for quantifying to what extent each variation driver is responsible for variants in a given collection of processes.

In our second case, we identified 8 additional variation points representing 5 variation drivers. Of the additional identified variation drivers, 3 are related to product and could be classified as product driven and two are related to customer and therefore can be classified as customer driven variations. Within product type, we found 3 distinct variation drivers. The first one concerned type of transaction (NASF vs. non-NASF), the second variation driver was related to number of transactions (single vs. multiple package), and the third referring to what kind of property deed is being processed. As to customer type, the first variation driver is related to how the customer has come in contact with the agency (via online vs. over the counter) and the second refers to if it’s an existing or new customer (new vs. existing).

Table 2. Analysis of variation drivers in the governmental agency case

Market (where)	**Product (what)**			**Customer (who)**	
Area	**Type of Product**			**Type of Customer**	
South vs. North	*NASF vs. non-NASF*	*Single vs. Multiple Package*	*Type of Deed*	*Type of Contact*	*Type of Customer*
9	2	1	1	2	1
9	***4***			***3***	

We also identified a candidate variation point related to managing refund of payments, but we chose not to define it as a variation point because it could be considered to be variants within the sub-process of payments and not of the overall process of management of documentation processing. However it could be defined as a variation point depending on the objective of the analyst and on what granularity level the analyst is working with, as we discussed in Section 2.1.

Our first research question was "could our framework be applied for identifying variation points and elicit their drivers?" Our analysis of two collections of process models consisting of a total of 54 process models indicates that our framework can be applied for identifying variation points and elicit their variation drivers. In our first case (the bank), we made explicit 6 variation drivers and in our second case (the governmental agency) we identified 5 variation drivers in the process models that was not known before our analysis.

Our second research question was "could our framework be used to classify the driver at each variation point orthogonally?" In our cases, we could classify all identified variation drivers orthogonally in operational, product, market or customer driven variations.

Our preliminary validation has limitations. Firstly, we have only validated our framework on two collections of process models covering 17 top-level process models. Hence, the conclusions are not generalizable. On the other hand, it should be underscored that the cases are taken from industrial practice. Secondly, there is a possible confirmatory bias in the study, as the collections of process models were analyzed by the authors of this paper. Finally, in one of the cases (the banking one) variants had already been implicitly recognized and captured as separate process models. Hence, this case did not lead to the identification of new variants, though it led to surfacing up implicit variation drivers.

4 Related Work

Ludwig *et al.* [13] applied the Work Practice Design (WPD) method to elicit variations with the end purpose of standardizing business processes. WPD as a method, much like similar approaches such as for example user-centered design, covers data collection such as interviews and observational studies that give the input for identifying and adjudicating variations in a process. The WPD in itself does not provide a systematic tool for identification of variations but rather will provide the analyst with the data necessary for variation identification.

Pascalau and Rath [15] introduced an ontology-based approach to manage variations in business process models by connecting the reason for which a variation exists to its variants. It is a method of managing variations that allows the annotation of business facts in the process models but it assumes that the business facts have been identified. Our framework is complementary as it serves the analyst to identify meaningful variations by analyzing a collection of process models using our framework.

La Rosa *et al.* [12] have introduced a questionnaire-based approach to be applied on reference models captured in c-EPC (Configurable Event-driven Process Chains). Analysts are given a set of questions that are linked to a consolidated process model representing all possible variants. By answering these questions, the method will extract the relevant variant from the consolidated process model and present it to the analyst. However, it is assumed that the questions and its corresponding "facts" are given and our framework is complementary as it assists the analyst in eliciting and categorizing the variation drivers from which such questions can be derived.

Identification of variations within the domain of feature diagrams have been studied fairly extensively and there seems to be an academic agreement that variability is more easily identified and managed using the concept of features within software product families [19]. However, feature diagrams take the viewpoint of the product and are primarily aimed at describing product variations as for example they occur in the context of software product lines. Our framework encompasses not only the product variations but also the market, customer, operational and time variations.

5 Conclusion

Managing variations in consolidated process models is a challenge that continues to be an open question in the academic community. Many approaches and methods [8, 11, 12, 15 and 17,] have been put forward to manage process variations. Our review of related work indicates that these methods and approaches are built on the assumption that the variations have already been successfully identified. Our framework complements this previous work by providing a systematic approach to identify and classify variations in a given process model or a collection of process models.

We applied our framework on two collections of process models. The first collection of process models had been arranged in clusters of two variants; one for domestic and the other for foreign equity and all the variants had implicitly been identified and modeled as separate process models. The second collection of process models had been modeled along geographical area but had not identified any other variation drivers in the process models. In the first case (the bank), we did not identify any additional variants that were not known before but our analysis identified additional 6 drivers of variations that were implicit in the collection of process models. These drivers could then be orthogonally classified as product, customer and operational driven variations. Our analysis also showed that variations along execution type (operational driver) and counterpart type (customer driven) were more common. In fact, the process models were arranged along the least occurring driver of variation. In our second case (governmental agency) we identified a total of 6 variation drivers. These drivers could be orthogonally classified as product, customer and market. Our analysis concludes that our framework can be applied for eliciting variation drivers and that the drivers can be orthogonally classified as operational, product, customer, market or time driven variations.

Naturally and as previously acknowledged, these cases should be treated as a preliminary validation only. A systematic analysis of other collections of process models by independent teams of analysts would be needed in order to conclusively assert the applicability of the framework.

Currently, the proposed framework allows for eliciting variation points and drivers in a given collection of process models. Once this elicitation completed, a possible extension is the analysis of the overall impact of drivers in the process. Some drivers located at the beginning of a process may have higher impact than others located within a specific region of the process or towards the end of a process. This gives rise to opportunities of assessing the impact of a driver with respect to a particular performance measure in a process. Providing manual or semi-automated methods to support such assessments is a possible direction for future work.

Acknowledgement. This research was supported by the European Social Fund via the Doctoral Studies and Internationalization Programme – DoRa.

References

1. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards Cross-Organizational Process Mining in Collections of Process Models and Their Executions. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBIP, vol. 100, pp. 2–13. Springer, Heidelberg (2012)
2. Dibb, S., Stern, P., Wensley, R.: Marketing knowledge and the value of segmentation. Marketing Intelligence & Planning 20(2), 113–119 (2002)
3. Dalek, C.K., Carlsson, B.: Manufacturing In Decline? A Matter of Definition. Economics of Innovation and New Technology 8(3) (1999)
4. Dumas, M.: Consolidated Management of Business Proess Variants, Keynote speech at PMC 2011, Workshop Proceedings (2011)
5. Foedermayr, E.K., Diamantopoulos, A.: Market Segmentation in Practice: Review of Empirical Studies, Methodological Assessment, and Agenda for Future Research. Journal of Strategic Marketing 16(3), 223–265 (2008)
6. Gottschalk, F., Wagemakers, T.A.C., Jansen-Vullers, M.H., Van der Aalst, W.M.P., La Rosa, M.: Configurable Process Models: Experiences from a Municipality Case Study. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 486–500. Springer, Heidelberg (2009)
7. Hallerbach, A., Bauer, T., Reichert, M.: Managing Process Variants in the Process Life Cycle. ICEIS (3-2), 154–161 (2008)
8. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and Management of Process Variants. In: Intl. Handbook on Business Process Management I, pp. 237–255. Springer
9. Kleijn, M.J., Dekker, R.: An overview of inventory systems with several demand classes. In: Speranza, M.G., Stahly, P. (eds.) New Trends in Distribution Logistics, pp. 253–265. Springer, Berlin (1999)
10. Klingemann, J.: Controlled Flexibility in Workflow Management. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 126–141. Springer, Heidelberg (2000)

11. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Merging Business Process Models. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010, Part I. LNCS, vol. 6426, pp. 96–113. Springer, Heidelberg (2010)
12. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.H.M.: Questionnaire-driven Configuration of Reference Process Models. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 424–438. Springer, Heidelberg (2007)
13. Ludwig, H., Rankin, Y., Enyedi, R., Anderson, L.C.: Process Variation Analysis Using Empirical Methods: A Case Study. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 62–65. Springer, Heidelberg (2011)
14. Manez, J.A., Waterson, M.: Multiproduct Firms and Product Differentiation: a Survey. The Warwick Economics Research Paper Series (TWERPS) 594, University of Warwick, Department of Economics (2001)
15. Pascalau, E., Rath, C.: Managing business process variants at eBay. In: Mendling, J., Weidlich, M., Weske, M. (eds.) BPMN 2010. LNBIP, vol. 67, pp. 91–105. Springer, Heidelberg (2010)
16. Reinhartz-Berger, I., Soffer, P., Sturm, A.: Organisational reference models: supporting an adequate design of local business processes. International Journal of Business Process Integration and Management 4(2), 134 (2009)
17. Rosemann, M., Van der Aalst, W.M.P.: A configurable reference modeling language. Inf. Syst (IS) 32(1), 1–23, 12–14 (2007)
18. Rummler, G.A., Ramais, A.J.: A Framework for Defining and Designing the Structure of Work. In: Handbook of Business Process Management, vol. 1. Springer (2010)
19. Svahnberg, M., van Gurp, J., Bosch, J.: A taxonomy of variability realization techniques. Softw., Pract. Exper. 35(8), 705–754 (2005)
20. Teichert, T., Shehu, E., Von Wartburg, I.: Customer segmentation revisited: The case of the airline industry. Transportation Research Part A: Policy and Practice 42, 227–242 (2008)
21. Tsiptsis, K., Chorianopoulos, A.: Data Mining Techniques in CRM: Inside Customer Segmentation (2009)
22. van der Aalst, W.M.P., Dumas, M., Gottschalk, F., ter Hofstede, A.H.M., La Rosa, M., Mendling, J.: Correctness-Preserving Configuration of Business Process Models. In: Fiadeiro, J.L., Inverardi, P. (eds.) FASE 2008. LNCS, vol. 4961, pp. 46–61. Springer, Heidelberg (2008)
23. Van der Aalst, W.M.P., Rosemann, M., Dumas, M.: Deadline-based escalation in process-aware information systems. Decision Support Systems 43(2), 492–511 (2007)
24. Wedel, M., Wagner, K.: Market Segmentation: Conceptual and Methodological Foundations. Kluwer, Dordrecht (2000)
25. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Heidelberg (2007)

Modeling Styles in Business Process Modeling

Jakob Pinggera[1], Pnina Soffer[2], Stefan Zugal[1], Barbara Weber[1], Matthias Weidlich[3], Dirk Fahland[3], Hajo A. Reijers[4], and Jan Mendling[5]

[1] University of Innsbruck, Austria
{jakob.pinggera,stefan.zugal,barbara.weber}@uibk.ac.at
[2] University of Haifa, Israel
spnina@is.haifa.ac.il
[3] Eindhoven University of Technology, The Netherlands
{d.fahland,h.a.reijers}@tue.nl
[4] Technion - Israel Institute of Technology
weidlich@tx.technion.ac.il
[5] Vienna University of Economics and Business
jan.mendling@wu.ac.at

Abstract. Research on quality issues of business process models has recently begun to explore the process of creating process models. As a consequence, the question arises whether different ways of creating process models exist. In this vein, we observed 115 students engaged in the act of modeling, recording all their interactions with the modeling environment using a specialized tool. The recordings of process modeling were subsequently clustered. Results presented in this paper suggest the existence of three distinct modeling styles, exhibiting significantly different characteristics. We believe that this finding constitutes another building block toward a more comprehensive understanding of the process of process modeling that will ultimately enable us to support modelers in creating better business process models.

Keywords: business process modeling, process of process modeling, modeling styles, cluster analysis.

1 Introduction

Considering the heavy usage of business process modeling in all types of business contexts, it is important to acknowledge both the relevance of process models and their associated quality issues. However, actual process models display a wide range of problems [1]. Following the SEQUAL framework [2], quality dimensions of models include syntactic, semantic, and pragmatic quality. Syntactic and semantic quality relate to model construction, and address the correct use of the modeling language and the extent to which the model truthfully represents the real world behavior it should depict, respectively. Pragmatic quality addresses the extent to which a model supports its usage for purposes such as understanding behavior or developing process aware systems. Considering process models whose purpose is to develop an understanding of real world behavior, pragmatic

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 151–166, 2012.

quality is typically related to the understandability of the model [3]. Clearly, an in-depth understanding of the factors influencing the various quality dimensions of process models is in demand.

Most research in this area puts a strong emphasis on the product or outcome of the process modeling act (e.g., [4,5]). For this category of research, the resulting model is the object of analysis. Many other works—instead of dealing with the quality of individual models—focus on the characteristics of modeling languages (e.g., [6,7]). Recently, research has begun to explore another dimension presumably affecting the quality of business process models by incorporating the process of creating a process model into their investigations (e.g., [8,9]). In particular, the focus has been put on the formalization phase in which a process modeler is facing the challenge of constructing a syntactically correct model reflecting a given domain description (cf. [10]). Our research can be attributed to the latter stream of research.

This paper contributes to our understanding of the process of process modeling (PPM) by investigating whether different ways of process modeling can be identified, i.e., can we observe different modeling styles when modelers create process models? Knowledge about different modeling styles will support the creation of customized process modeling environments, supporting modelers in creating high quality models. Similarly, a more comprehensive understanding of the PPM can be exploited for teaching students in how to create process models of high quality. We conducted a modeling session with 115 students, recording all their interactions with the modeling environment using a specialized tool. To identify different modeling styles the collected PPM instances were automatically clustered suggesting the existence of three different modeling styles. The modeling styles were subsequently analyzed using a series of measures for quantifying the PPM to validate differences between the three groups.

The paper is structured as follows. Section 2 presents backgrounds on the PPM and introduces measures for quantifying this process. Section 3 describes data collection and cluster analysis. Section 4 presents the results, followed by their discussion in Section 5. The paper is concluded with a discussion of related work in Section 6 and a brief summary in Section 7.

2 Backgrounds

This section provides background information on the PPM and explains how this process can be captured and quantified using a series of measures.

2.1 The Process of Process Modeling

During the formalization phase process modelers are working on creating a syntactically correct process model reflecting a given domain description by interacting with the process modeling tool [10]. This modeling process can be described as an iterative and highly flexible process [11,12], dependent on the individual modeler and the modeling task at hand [13]. At an operational level, the modeler's interactions with the tool would typically consist of a cycle of the three successive phases of (1) comprehension (i.e., the modeler forms a mental model of domain behavior), (2) modeling (i.e., the modeler maps the mental model to modeling constructs), and (3) reconciliation (i.e., the modeler reorganizes the process model) [9,8].

Comprehension. Research on human cognition and problem solving has shed light on comprehension. According to [14], when facing a task, the problem solver first formulates a mental representation of the problem, and then uses it for reasoning about the solution and which methods to apply for solving the problem. In process modeling, the task is to create a model which represents the behavior of a domain. The process of forming mental models and applying methods for achieving the task is not done in one step applied to the entire problem. Rather, due to the limited capacity of working memory, the problem is broken down to pieces that are addressed sequentially, chunk by chunk [8,9].

Modeling. The modeler uses the problem and solution developed in working memory during the previous comprehension phase to materialize the solution in a process model (by creating or changing it) [8,9]. The modeler's utilization of working memory influences the number of modeling steps executed during the modeling phase before forcing the modeler to revisit the problem for acquiring more information [9].

Reconciliation. After modeling, modelers typically reorganize the process model (e.g., renaming of activities) and utilize the process model's *secondary notation* (e.g., notation of layout, typographic cues) to enhance the process model's understandability [15,16]. However, the number of reconciliation phases in the PPM is influenced by a modeler's ability of placing elements correctly when creating them, alleviating the need for additional layouting [9].

2.2 Capturing Events of the Process of Process Modeling

To investigate the PPM, actions taken during modeling have to be recorded and mapped to the phases described above. When modeling in a process modeling environment, process modeling consists of adding nodes and edges to the process model, naming or renaming activities, and adding conditions to edges. In addition to these interactions a modeler can influence the process model's secondary notation, e.g., by laying out the process model using move operations for nodes or by utilizing bendpoints to influence the routing of edges, see [9] for details.

To capture modeling activities, and for obtaining a closer look on how process models are created in a systematic manner, we instrumented a basic process modeling editor to record each user's interactions together with the corresponding time stamp in an event log, describing the creation of the process model step by step. Editor and event recording are available within Cheetah Experimental Platform (CEP) [17].

2.3 Quantifying the Process of Process Modeling

A log of modeling events allows quantitative analysis of a PPM. Based on the conceptual background, comprehension (C), modeling (M), and reconciliation (R) phases can be identified by grouping events into respective phases (see [9] for details). Then, a PPM can be divided into *modeling iterations* [9]. One iteration is assumed to comprise a comprehension (C), modeling (M), and reconciliation (R) phase in this respective order. The iterations of a modeling process are identified by aligning its phases to the CMR-pattern. If a certain phase of this pattern is not present in the modeling process, the respective phase is skipped for the observed iteration and the process is considered to continue with the next phase of the pattern. In the following we present five measures quantifying the process of process modeling.

Number of Iterations. This measure counts the modeling iterations per PPM reflecting how often a modeler had to interrupt modeling for comprehension or reconciliation.

Share of Comprehension. When comprehending, a mental model of the problem and a corresponding solution is developed which is then formalized in modeling phases. Differences in the amount of time spent on comprehension can be expected to characterize modeling styles and to impact on the modeling result. We quantify this aspect as the ratio of the average length of a comprehension phase in a process to the average length of an iteration. The initial comprehension phase is neglected as it is typically subject to various influences unrelated to problem solving (e.g., the modeler did not start immediately).

Iteration Chunk Size. Modelers can be assumed to conduct modeling in chunks of different sizes. We quantified *chunk size* as the average number of create and delete operations executed in one iteration. This measure reflects the ability to model large parts of a model without the need to comprehend or reconcile.

Reconciliation Breaks. A steady process of modeling is assumed to be a sequence of iterations following the CMR-pattern. Reconciliation can sometimes be skipped if the modeler can place all model elements directly at the right spot clearly alleviating the need for reconciliation. However, some processes may even show iterations of CR-patterns, i.e., an iteration without a modeling phase, where a modeler interrupts the common flow of modeling for additional reconciliation. We quantified this aspect by the relative share of iterations that comprise unexpected reconciliation (without modeling) out of all iterations.

Delete Iterations. From time to time, modelers are required to remove content from the process model. This might happen when modelers identify errors in the model, which are subsequently resolved by removing some of the modeling constructs and implementing the desired functionality. This measure describes the number of iterations of the PPM containing delete operations relative to the total number of iterations of the PPM.

3 Clustering

To be able to make generalizations, we have used cluster analysis to a set of PPM instances. Cluster analysis allows us to identify groups of modelers exhibiting similar modeling styles. This section describes the modeling session, data preprocessing and cluster analysis.

3.1 Data Collection

The modeling session was designed to collect PPM instances of students creating a formal process model in BPMN from an informal description. The object that was to be modeled is a process describing the activities a pilot has to execute prior to taking off an aircraft[1].

To mitigate the risk that the PPM instances were impacted by complicated tools or notations [11], we decided to use a subset of BPMN for our experiment. In this way, modelers were confronted with a minimal number of distractions, but the essence of how process models are created could still be captured. A pre-test was conducted at the University of Innsbruck to ensure the usability of the tool and the understandability of the task description. This led to further improvements of CEP and minor updates to the task description.

The modeling sessions were conducted in November 2010 with students of a graduate course on Business Process Management at Eindhoven University of Technology and in January 2011 with students from Humboldt-Universität zu Berlin following a similar course. The modeling session at each university started with a demographic survey, followed by a modeling tool tutorial explaining the basic features of CEP. After that, the actual modeling task was presented in which the students had to model the above described "Pre-Flight" process. This was done by 102 students in Eindhoven and 13 students in Berlin. By conducting the experiment during class and closely monitoring the students, we mitigated the risk of falsely identifying comprehension phases due to external distractions. No time restrictions were imposed on the students.

3.2 PPM Profile for Clustering

When trying to identify different types of PPM instances using clustering, the question arises how to represent such a process to make clustering possible. Based on our previous experience with the PPM we decided to focus on four aspects. The adding of content, the removal of content, reconciliation of the model and comprehension time, i.e., the time when the modeler does not work on the process model. To also reflect that modeling is a time-dependent process, we do not just look at the total amount of modeling actions and comprehension, but on their *distribution* over time as follows. We sampled every process into segments of 10 seconds length. For each segment, we compute its *profile* (a, d, r, c), i.e., the numbers a, d, and r of add, delete, and reconciliation events,

[1] Material download: http://pinggera.info/experiment/ModelingStyles

Table 1. Classification of CEP's User Interactions

Interaction	Classification	Interaction	Classification
CREATE NODE	Adding	RENAME ACTIVITY	Reconciliation
DELETE NODE	Deleting	UPDATE CONDITION	Reconciliation
CREATE EDGE	Adding	MOVE NODE	Reconciliation
DELETE EDGE	Deleting	MOVE EDGE LABEL	Reconciliation
RECONNECT EDGE	Adding/Deleting	MODIFY EDGE BENDPOINT	Reconciliation

and the time c spent on comprehension. The *profile* of one PPM is then sequence $(a_1, d_1, r_1, c_1)(a_2, d_2, r_2, c_2)\ldots$ of its segments' profiles. The a, d, and r are obtained per segment by classifying each event according to Table 1. Adding a condition to an edge was considered being part of creating an edge. Comprehension time c was computed as follows. Group events to intervals: an interval is a sequence of events where two consecutive events are ≤ 1 second apart, its duration is the time difference between its first and its last event (intervals of 1 activity got a duration of 1 second). Then c is the length of the segment (10 secs) minus the duration of all intervals in the segment. For example, if the modeler moved activity A after 3 secs, activity B after 3.5 secs and activity C after 4.2 secs the comprehension time in this segment would be $10 - 1.2 = 8.8$ seconds. To give all PPM profiles equal length, shorter profiles were extended with segments of no interaction to reach the length of the longest PPM (required for clustering).

3.3 Clustering

The PPM profiles were exported from CEP [17] and subsequently clustered using Weka[2]. The KMeans algorithm, first proposed in [18], utilizing an euclidean distance measure was chosen for clustering as it constitutes a well known and easy to use means for cluster analysis. As KMeans might converge in a local minimum [19], the obtained clustering has to be validated. If the identified clusters exhibit significant differences with regard to the measures described in Section 2, we conclude that different modeling styles were identified. KMeans requires the number of cluster to be known a priori. As this was not the case we gradually increased the number of clusters starting from 2, resulting in only one major cluster. Setting the number of expected clusters to 3 revealed two major clusters and one cluster of 2 PPM instances. Most promising results were achieved by setting the number of clusters to be generated to 4 and starting with a seed of 10, returning 3 major clusters and one small cluster of 2 PPM instances. We considered these 3 major clusters for further analysis; increasing the number of expected clusters only generated additional small clusters.

4 Results

In this section we present results of the cluster analysis and validate the difference among the clusters using the measures described in Section 2.

[2] http://www.cs.waikato.ac.nz/ml/weka

Table 2. Statistics per cluster

Measure	C1	C2	C3
Number of instances	42	22	49
Avg. no. of adding operations	61.36	52.91	52.57
Avg. no. of deleting operations	10.81	3.91	4.55
Avg. no. of reconciliation operations	76.26	42.00	39.27
Avg. no. operations	148.43	98.82	96.39

4.1 Three Clusters

We identified three major clusters of 42, 22 and 49 instances, called C1, C2, and C3 in the sequel. In order to visualize the obtained clusters we calculated the average number of adding, deleting and reconciliation operations per segment for each cluster. Additionally, we calculated the moving average of six segments, i.e., one minute, providing us with a smoother representation of the modeling processes presented in Figures 1, 2, and 3 for C1, C2, and C3 respectively. The horizontal axis denotes the segments into which the PPM instances were sampled. The vertical axis indicates the average number of operations that were performed in this segment. For example, a value of 0.8 for segment 9 (cf. Fig. 2) indicates that all modelers in this cluster averaged 0.8 adding operations within this 10 second segment.

C1 (cf. Fig. 1) is characterized by long PPM instances, as the first time the adding series reaches 0 is after about 205 segments. Additionally, the delete series indicates more delete operations compared to the other clusters. Several fairly large spikes of reconciliation activity can be observed, the most prominent one after about 117 segments.

C2, as illustrated in Fig. 2, is characterized by a fast start as a peak in adding activity is reached after 13 segments. In general, the adding series is most of the time between 0.5 and 0.9 operations, higher compared to the other two clusters. The fast modeling behavior results in short PPM instances as the adding series is 0 for the first timer after about 110 segments.

On first sight, C3 (cf. Fig. 3) seems to be between C1 and C2. The adding curve is mostly situated between 0.4 and 0.7, a littler lower than for C2, but still higher compared to C1. Similar values can be observed for the reconciliation curve. The deleting curve remains below 0.1. The duration of the PPM instances is also between the duration of C1 and C2 as the adding series is 0 for the first time after about 137 segments.

Table 2 presents general statistics on the number of adding operations, the number of deleting operations, the number of reconciliation operations and the total number of operations for each cluster. Interestingly, modelers in C1 had more adding operations, more deleting operations and, probably most notable, almost twice as many reconciliation operations compared to C2 and C3. At a first glance, the numbers for C2 and C3 appear to be very similar.

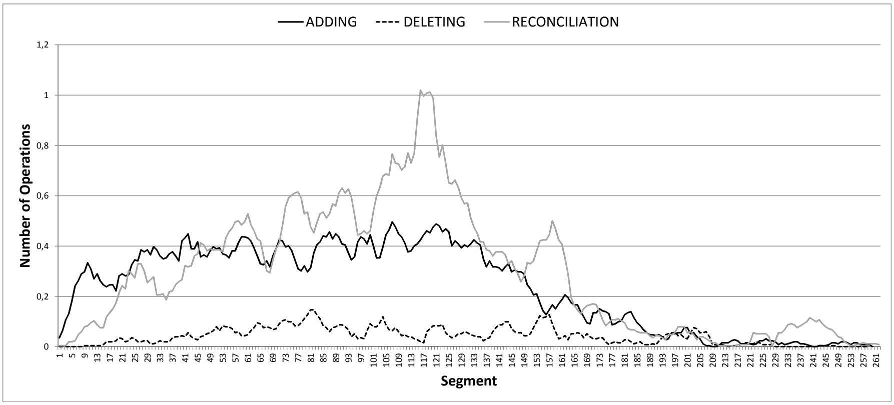


Fig. 1. Cluster C1

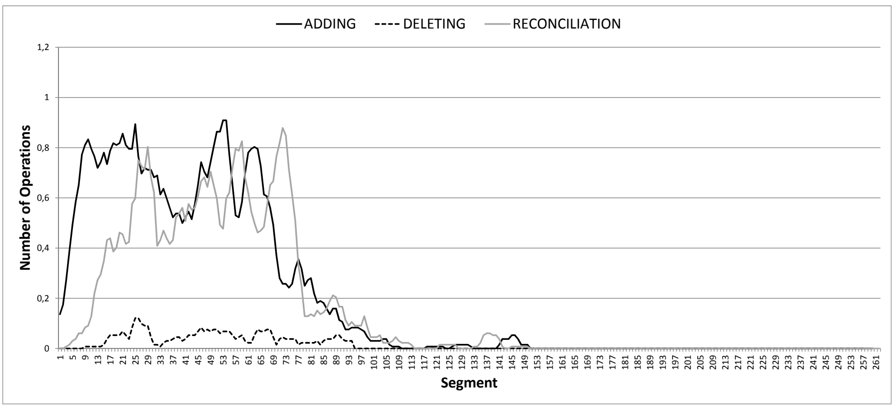


Fig. 2. Cluster C2

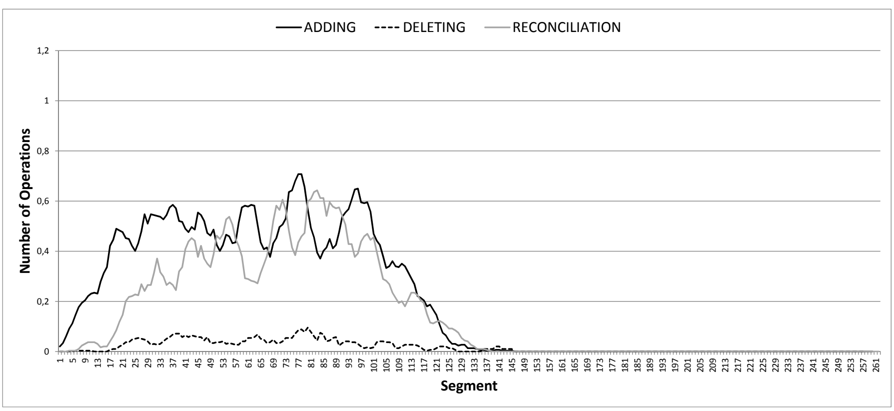


Fig. 3. Cluster C3

Table 3. Significant differences for statistics

Statistic		**All groups**	**Pairwise comparison**		
			1-2	1-3	2-3
Number of Adding Operations	Sig.	0.000[a]	0.003[a]	0.000[a]	
	test	Oneway ANOVA	Bonferroni post-hoc test		
Number of Deleting Operations	Sig.	0.000[a]	0.000[b]	0.000[b]	
	test	Kruskall-wallis	Mann-whitney test		
Number of Reconciliation Operations	Sig.	0.000[a]	0.000[b]	0.000[b]	
	test	Kruskall-wallis	t-test for unequal variances		
Number of Total Operations	Sig.	0.000[a]	0.000[b]	0.000[b]	
	test	Kruskall-wallis	t-test for unequal variances		

[a] $p < 0.05$ [b] $p < 0.05/3$

The following procedure for conducting the statistical analysis was used. If the data was normally distributed and homogenity of variances was given we used Oneway ANOVA to test for differences among the groups. Pairwise comparisons were done using the bonferroni post-hoc test. Note that the bonferroni post-hoc test uses an adapted significance level. Therefore, p-values less than 0.05 are considered to be significant, i.e., there is no need to divide the significance level by the number of groups, i.e., clusters. In case normal distribution or homogenity of variance was not given a non-parametric alternative to ANOVA, i.e., kruskall-wallis, was utilized to test for differences among the groups. Pairwise comparisons were done using the t-test for (un)equal variances (depending on the data) if normal distribution was given. If no normal distribution could be identified the mann-whitney test was utilized. In either case, i.e., t-test or mann-whitney test, the bonferroni correction was applied, i.e., the significance level was divided by the number of clusters.

The results are summarized in Table 3, indicating significant differences between C1 and C2 and C1 and C3, but not between C2 and C3. Only significant differences are stated.

4.2 Applying Measures

In order to further distill the properties of the three clusters, we calculated the measures described in Section 2.3 for each PPM. Table 4 provides an overview presenting the average values for each measure in each cluster. As indicated in Fig. 1, C1 constitutes the highest number of PPM iterations. Tightly connected to this observation is the average iteration chunk size. Modelers in C2 added by far the most content per iteration to the process model. Also the number of iterations containing delete iterations is higher for C1 than for the other clusters, which is consistent with the higher number of delete operations (cf. Table 2). The amount of time spent on comprehending the task description and developing the plan on how to incorporate them into the process model seems to be far larger for C1 compared to C2, which has the lowest share of comprehension, but also larger compared to C3. When considering reconciliation breaks C3 sets itself

Table 4. Measures per cluster

Measure	C1	C2	C3
Avg. no. of PPM iterations	21.50	12.32	14.69
Avg. iteration Chunk Size	3.66	5.28	4.24
Avg. share of comprehension	49.88	39.28	45.02
Avg. reconciliation breaks	21.37	18.14	13.85
Avg. delete iterations	17.06	10.07	10.83

Table 5. Significant differences for measures

Measure		All groups	Pairwise comparison		
			1-2	1-3	2-3
Iteration Chunk Size	Sig.	0.000[a]	0.000[b]	0.000[b]	0.007[b]
	test	Kruskall-wallis	t-test for unequal variances		
Number of Iterations	Sig.	0.000[a]	0.000[b]	0.000[b]	0.004[b]
	test	Kruskall-wallis	t-test for unequal variances		
Share of Comprehension	Sig.	0.000[a]	0.000[a]	0.036[a]	0.045[a]
	test	Oneway ANOVA	Bonferroni post-hoc test		
Delete Iterations	Sig.	0.005[a]	0.026[a]	0.011[a]	
	test	Oneway ANOVA	Bonferroni post-hoc test		
Reconciliation Breaks	Sig.	0.005[a]		0.004[a]	
	test	Oneway ANOVA	Bonferroni post-hoc test		

[a] $p < 0.05$ [b] $p < 0.05/3$

apart posting the lowest number of reconciliation breaks. C2 is somewhere in between and C1 has the highest number of reconciliation breaks.

Statistical analysis of the differences between the groups was performed following the procedure described in the previous section. An overview of the results is presented in Table 5. Only significant differences are stated. In constrast to the statistics presented in Table 3, we were able to identify significant differences between C2 and C3.

5 Discussion

In this section we present our insights when comparing the identified clusters and we discuss the lessons learned in this work and how they influence our future work. Additionally, limitations of this work are described.

5.1 Cluster C1

C1 can be clearly distinguished from C2 and C3. This becomes evident on visual inspection of Fig. 1, but also when considering the number of adding operations, the number of deleting operations, the number of reconciliation operations and the total number of operations. We identified statistically significant differences between C1 and C2 and between C1 and C3 for all statistics (cf. Table 3).

In general, modelers in C1 had rather long PPM instances, i.e., the number of PPM iterations was significantly higher compared to C2 and C3. In addition, modelers in C1 spent more time on comprehension compared to C2. Modelers started rather slowly, not eclipsing 0.5 adding operations. The slow modeling speed is underlined by the significantly lower chunk size compared to C2 and C3. During the whole process, adding operations are accompanied by a relatively high amount of delete operations. This is underlined by the significant differences in the number of delete iterations between C1 and C2 and C1 and C3 (cf. Table 5). Also, we observed a fairly large amount of reconciliation operations, culminating in a massive peak after about half of the PPM instances.

The results suggest that modelers in C1 were not as goal oriented as their colleagues in other clusters, since they spent a great amount of time on comprehension, added more modeling elements which were subsequently removed and put significantly more effort into improving the visual appearance of the process model. There might be multiple reasons for this behavior. On the one hand, it could point toward modelers having trouble executing the modeling task and therefore needed more reconciliation to facilitate their understanding of the process model at hand. On the other hand, their focus on layouting might have acted as a distraction from the modeling task, resulting in the higher number of adding operations and deleting operations. Still, other techniques will be required for further investigating this claim, e.g., think aloud protocols (cf. [20]).

5.2 Cluster C2

When inspecting Fig. 2 the very steep start of the adding curve strikes the eye, indicating that modelers started creating the process model right away. When focusing on reconciliation operations, several spikes in the layouting curve can be identified, notably one last spike right after the number of adding operations decreases. As already mentioned above, C2 is statistically significant different compared to C1 for all statistics presented in Table 2. No differences can be identified between C2 and C3.

Considering the measures described in Section 2.3, C2 has a significantly higher chunk size compared to C1 and C3. Similarly, we observed the lowest number of PPM iterations. This means that modelers add a lot more content per PPM iteration. In addition, modelers in C2 did not spend as much time on comprehension compared to modelers in C1 and C3.

In a nutshell, modelers of C2 are very focused and goal oriented following a straight path when creating the process model. They are quick in making decisions about how to proceed and only slow down their modeling endeavor from time to time for some reconciliation, resulting in short PPM instances.

5.3 Cluster C3

Fig. 3 shows the PPM instances for C3. The processes are shorter compared to C1 and longer compared to C2. It is lacking the fast start of the adding curve we identified for C2. The reconciliation curve is more or less following the adding curve. Notably, this is the only curve without a reconciliation spike once the number of adding operations decreases.

The calculated measures indicate clear differences to C1 when it comes to chunk size, number of iterations, share of comprehension, but also number of delete operations and reconciliation breaks. C2 and C3 differ in chunk size, the number of iterations and the time spent on comprehension.

When comparing C2 and C3, the question arises whether modelers in C3 followed the same strategy as modelers in C2, just a little slower. We believe that, in contrary to C2, modelers of C3 followed a more systematic approach to process modeling. They continuously reconciled their process model, alleviating the need for dedicated reconciliation breaks. This is indicated by the lack of a reconciliation spike after the decrease of adding operations in Fig. 3. Additionally, reconciliation breaks points into this direction (18.16 for C2 vs. 13.85 for C3). Fig. 4 depicts the reconciliation breaks box plot, hinting at a difference in reconciliation breaks between C2 and C3. Still, the difference did not turn out to be statistically significant leaving us with some future work on investigating whether this claim actually holds.

Additionally, we believe that different reasons for reconciliation breaks exist. On the one hand, modelers are forced to stop their modeling endeavour and layout the process model when they are overwhelmed by the complexity at hand. On the other hand, some modelers might stop modeling at strategic points to reconcile the process model in order to avoid situations like the one mentioned above before they even arise. Even though this explanation would fit the boxplot depicted in Fig. 4, further investigations are in demand to fully understand the reconciliation behavior of modelers.

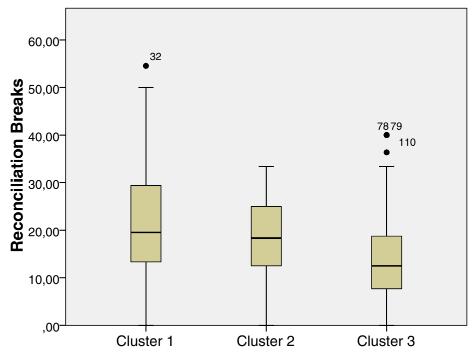


Fig. 4. Reconciliation Breaks

5.4 Lessons Learned

We were able to identify three different modeling styles using cluster analysis. Differences among the clusters were subsequently validated using a series of measures quantifying the PPM. Note that these measures were defined prior to performing the cluster analysis. The measures are based on the detected iterations of the PPM, approaching the PPM from a different angle. Therefore they enable us to validate the differences among the three clusters.

The detected modeling styles contribute to our understanding of the process of process modeling, as, to our knowledge, this is the first systematic attempt to establish a categorization of PPM instances in the domain of business process modeling. We believe that further refinements of the categorization will emerge, ultimately enabling us to create personalized modeling environments based on their observed modeling behavior. In addition, these findings can be exploited for teaching purposes. For example, teachers might be able to identify students facing difficulties during a modeling assignment based on their modeling behavior and provide them with additional support. Still, some research questions emerging from these findings have to be addressed first. The most pressing might be whether a modeler's personal style persists over several different modeling tasks or if the modeling style is determined by the modeling task at hand. To answer this question further empirical investigations are in demand. Based on some preliminary observations we would assume that the influence of the modeling task cannot be neglected. Even though a modeler might like to create a process model in a straight forward, goal oriented way, the complexity of the modeling task might force her to reduce the modeling speed and switch to a more conservative modeling style.

On a long-term basis questions on how to exploit this knowledge to improve the quality of the resulting process model become evident. Unfortunately, the naive assumption that one modeling style is superior to the others could not be confirmed. All clusters contained excellent process models and process models of low quality. This is not surprising though. Even modelers in C1 who face difficulties, exhibiting long PPM instances, can still come up with good process models if they succeed in overcoming the adversity they are facing.

5.5 Limitations

The interpretation of our findings is presented with the explicit acknowledgement of a number of limitations to our study. First of all, our respondents represented a rather homogeneous and inexperienced group. Although relative differences in experience were notable, the group is not representative for the modeling community at large. At this stage, in particular, the question can be raised whether experienced modelers also exhibit the same style elements as skillful yet inexperienced modelers. In other words, will experienced modelers display similar characteristics of style or can other styles be observed within their approaches? Note that we are mildly optimistic about the usefulness of the presented insights on the basis of modeling behavior of graduate students, since we have established in previous work that such subjects perform equally well in process modeling tasks as some professional modelers [21].

Secondly, the influence of the modeling task—more precisely, the modeling task's complexity (cf. [22])—on the PPM is not fully understood. All students in our modeling session were working on the same modeling assignment. Hence, the observed clusters might be specific to modeling tasks of this complexity level. Further investigations will be necessary to let sunlight fall on the influence of the modeling task, which might result in the emergence of additional clusters. Preliminary results of a different modeling task suggest the existence of modeling styles comparable to the results presented in this paper.

Thirdly, we can not rule out that KMeans identified a local minimum, resulting in a suboptimal clustering. To counter this threat we validated the clustering using a series of measures quantifying the PPM and identified significant differences among the three groups.

6 Related Work

Our work is essentially related to model quality frameworks and research on the process of modeling.

There are different frameworks and guidelines available that define quality for process models. Among others, the SEQUAL framework uses semiotic theory for identifying various aspects of process model quality [3], the Guidelines of Process Modeling describe quality considerations for process models [23], and the Seven Process Modeling Guidelines define desirable characteristics of a process model [24]. While each of these frameworks has been validated empirically, they rather take a static view by focusing on the resulting process model, but not on the act of modeling itself. Our research takes another approach by investigating the process followed to create the process model.

Research on the process of modeling typically focuses on the interaction between different parties. In a classical setting, a system analyst directs a domain expert through a structured discussion subdivided into the stages elicitation, modeling, verification, and validation [10,25]. The procedure of developing process models in a team is analyzed in [26] and characterized as a negotiation process. Interpretation tasks and classification tasks are identified on the semantic level of modeling. Participative modeling is discussed in [27]. These works build on the observation of modeling practice and distill normative procedures for steering the process of modeling towards a good completion. Our work, in turn, focuses on the formalization of the process model, i.e., the modeler's interactions with the modeling environment when creating the formal business process model.

7 Summary

This paper contributes to our understanding of the PPM as it constitutes the first systematic attempt to identify different modeling styles in the domain of business process modeling. We conducted a modeling session with 115 students of courses on business process management, collecting their PPM instances. We were able to identify three different modeling styles using cluster analysis and validated the retrieved clusters using a series of measures for quantifying the PPM. We believe that a better understanding regarding the PPM will be beneficial for future process modeling environments and will support teachers in mentoring their students on their way to professional process modelers.

Acknowledgements. This research was funded by the Austrian Science Fund (FWF): P23699-N23. The research stay of Dr. Pnina Soffer in Innsbruck, Austria was funded by BIT School.

References

1. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Springer (2008)
2. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Softw. 11, 42–49 (1994)
3. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. EJIS 15, 91–102 (2006)
4. Van der Aalst, W., Ter Hofstede, A.: Verification of workflow task structures: A petri-net-baset approach. IS 25, 43–69 (2000)
5. Gruhn, V., Laue, R.: Complexity metrics for business process models. In: Proc. ICBIS 2010, pp. 1–12 (2006)
6. Siau, K., Rossi, M.: Evaluation techniques for systems analysis and design modelling methods-a review and comparative analysis. ISJ 21, 249–268 (2011)
7. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Eng. 35, 756–779 (2009)
8. Soffer, P., Kaner, M., Wand, Y.: Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In: Proc. ER-BPM 2011, pp. 357–369 (2011)
9. Pinggera, J., Zugal, S., Weidlich, M., Fahland, D., Weber, B., Mendling, J., Reijers, H.A.: Tracing the process of process modeling with modeling phase diagrams. In: Proc. ER-BPM 2011, pp. 370–382 (2012)
10. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: A Fundamental View on the Process of Conceptual Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 128–143. Springer, Heidelberg (2005)
11. Crapo, A.W., Waisel, L.B., Wallace, W.A., Willemain, T.R.: Visualization and the process of modeling: a cognitive-theoretic view. In: Proc. KDD 2000, pp. 218–226 (2000)
12. Morris, W.T.: On the Art of Modeling. Management Science 13, B–707–B–717 (1967)
13. Willemain, T.R.: Model Formulation: What Experts Think about and When. Operations Research 43, 916–932 (1995)
14. Newell, A., Simon, H.: Human problem Solving. Prentice Hall (1972)
15. Petre, M.: Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. Commun. ACM, 33–44 (1995)
16. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007)
17. 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)
18. MacQueen, J.: Some methods of classification and analysis of multivariate observations. In: Proc. Berkeley Symposium on Math., Stat., and Prob., pp. 281–297 (1967)
19. Hamerly, G., Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. In: Proc. CIKM 2002, pp. 600–607 (2002)
20. Ericsson, K.A., Simon, H.A.: Protocol analysis: Verbal reports as data. MIT Press (1993)

21. Reijers, H., Mendling, J.: A study into the factors that influence the understandability of business process models. IEEE Transactions on Systems Man and Cybernetics, Part A 41, 449–462 (2011)
22. Cardoso, J.: Business process control-flow complexity: Metric, evaluation, and validation. JWSR 5, 49–76 (2008)
23. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–49. Springer, Heidelberg (2000)
24. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information & Software Technology 52, 127–136 (2010)
25. Frederiks, P., Weide, T.: Information modeling: The process and the required competencies of its participants. DKE 58, 4–20 (2006)
26. Rittgen, P.: Negotiating Models. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 561–573. Springer, Heidelberg (2007)
27. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)

Expressiveness and Understandability Considerations of Hierarchy in Declarative Business Process Models*

Stefan Zugal[1], Pnina Soffer[2], Jakob Pinggera[1], and Barbara Weber[1]

[1] University of Innsbruck, Austria
{stefan.zugal,jakob.pinggera,barbara.weber}@uibk.ac.at
[2] University of Haifa, Israel
spnina@is.haifa.ac.il

Abstract. Hierarchy has widely been recognized as a viable approach to deal with the complexity of conceptual models. For instance, in declarative business process models, hierarchy is realized by sub-processes. While technical implementations of declarative sub-processes exist, their application, semantics, and the resulting impact on understandability are less understood yet—this research gap is addressed in this work. In particular, we discuss the semantics and the application of hierarchy and show how sub-processes enhance the expressiveness of declarative modeling languages. Then, we turn to the impact on the understandability of hierarchy on a declarative process model. To systematically assess this impact, we present a cognitive-psychology based framework that allows to assess the possible impact of hierarchy on the understandability of the process model.

Keywords: Declarative Business Process Models, Hierarchy, Understandability, Cognitive Psychology.

1 Introduction

Using modularization to hierarchically structure information has for decades been identified as a viable approach to deal with complexity [1]. Not surprisingly, business process modeling languages provide support for hierarchical structures, e.g., sub-processes in BPMN and YAWL. However, in general, *"the world does not represent itself to us neatly divided into systems, subsystems... these divisions which we make ourselves"* [2]. In this sense, a viable discussion about the proper use of modularization for the analysis and design of information systems as well as its impact on understandability is still going on. In business process management, sub-processes have been recognized as an important factor influencing model understandability [3], however, there are no definitive guidelines on their use yet. For instance, recommendations regarding the size of a sub-process in an *imperative process model* range from 5–7 model elements [4] over

* This research is supported by Austrian Science Fund (FWF): P23699-N23 and the BIT fellowship program.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 167–181, 2012.

5–15 model elements [5] to up to 50 model elements [6]. For declarative process models, which have recently gained attention due to their flexibility [7], the proper usage of modularization is even less clear. While work has been done with respect to the technical support of declarative sub-processes, it remains unclear *whether and when* hierarchy has an influence on the *understandability* of the model. In general, empirical research into the understandability of conceptual models (e.g., ER diagrams or UML statecharts) has shown that hierarchy can have a positive influence [8], negative influence [9] or no influence at all [10]. For declarative process models, the situation is less clear as no empirical studies have been conducted so far. However, as declarative process models appear to be especially challenging to understand [11], it seems particularly important to improve their understandability. In the following, we will shed light on the question which influence on understandability can be expected for hierarchy in declarative process models.

The contribution of this work is twofold. First, the semantics of hierarchy in declarative process models is elaborated on. In particular, we will show that hierarchy is not just a question of structure, but also enhances expressiveness and has implications on the restructuring of a model. Second, the impact of hierarchy on the understandability of the model will be investigated systematically. We will present a cognitive-psychology based framework that explains general effects of hierarchy, but also takes into account peculiarities of declarative process models. The framework allows to assess the possible impact of hierarchy, i.e., whether a certain modularization of a declarative process model has a positive influence, negative influence or no influence at all. This, in turn, allows for a systematic empirical investigation in future work.[1]

The remainder of this paper is structured as follows. Section 2 introduces declarative process models. Then, Section 3 discusses the semantics of hierarchy in declarative process models. Subsequently, Section 4 deals with the application of hierarchy in declarative process models, whereas Section 5 investigates the impact on understandability. Finally, related work is presented in Section 6 and the paper is concluded with a summary and an outlook in Section 7.

2 Background: Declarative Processes

There has been a long tradition of modeling business processes in an imperative way. Process modeling languages supporting this paradigm, like BPMN, EPC and UML Activity Diagrams, are widely used. Recently, *declarative approaches* have received increasing interest and suggest a fundamentally different way of describing business processes [12]. While imperative models specify exactly *how* things have to be done, declarative approaches only focus on the logic that governs the interplay of actions in the process by describing the *activities* that can be performed, as well as *constraints* prohibiting undesired behavior. An example of a constraint in an aviation process would be that crew duty times cannot exceed

[1] Please note that even though we take into account declarative models in general, we will make use of the declarative language *ConDec* [12] for the discussion.

a predefined threshold. Constraints described in literature can be classified as execution and termination constraints. *Execution* constraints, on the one hand, restrict the execution of activities, e.g., an activity can be executed at most once. *Termination* constraints, on the other hand, affect the *proper* termination[2] of process instances and specify when process termination is possible. For instance, an activity must be executed at least once before the process can be terminated. Most constraints focus either on execution *or* termination semantics, however, some constraints also combine execution and termination semantics (e.g., the succession constraint [12]).

To illustrate the concept of declarative processes, a model specified in ConDec [12] is shown in Fig. 1 a). It contains activities *A* to *F* as well as constraints *C1* and *C2*. *C1* prescribes that A must be executed at least once (i.e., *C1* restricts the termination of process instances). *C2* specifies that *E* can only be executed if *C* has been executed at some point in time before (i.e., *C2* imposes restrictions on the execution of activity E). In Fig. 1 b) an example of a process instance illustrates the semantics of the described constraints. After process instantiation, *A*, *B*, *C*, *D* and *F* can be executed. *E*, however, cannot be executed as *C2* specifies that *C* must have been executed before (cf. grey bar in Fig. 1 b) below "E"). Furthermore, the process instance cannot be terminated as *C1* is not satisfied, i.e., *A* has not been executed at least once (cf. grey area in Fig. 1 b) below "Termination"). The subsequent execution of B does not cause any changes as it is not involved in any constraint. However, after *A* is executed, *C1* is satisfied, i.e., *A* has been executed at least once and thus the process instance can be terminated (cf. Fig. 1 b)—after *e4* the box below "Termination" is white). Then, *C* is executed, satisfying *C2* and consequently allowing *E* to be executed (the box below "E" is white after *e6* occurred). Finally, the execution of *E* does not affect any constraint, thus no changes with respect to constraint satisfaction can be observed. As all termination constraints are still satisfied, the process instance can still be terminated. Please note that declarative process instances have to be terminated explicitly, i.e., the end user must decide when to complete the process instance. Termination constraints thereby specify when termination is allowed, i.e., the process instance I could have been terminated at any point in time after *e4*.

As illustrated in Fig. 1, a process instance can be specified through a list of *events* that describe changes in the life-cycle of *activity instances*, e.g., *"e1: B started"*. In the following, we will denote this list as *execution trace*, e.g., for process instance I: <*e1*, *e2*, *e3*, *e4*, *e5*, *e6*, *e7*, *e8*>. If activities are non-overlapping, we merge subsequent start events and completion events, e.g., <*B started*, *B completed*, *A started*, *A completed*> is abbreviated by <*B*, *A*>.

3 Discussion of Semantics

This section aims at establishing an understanding of the semantics of subprocesses in a declarative model. Based on this, the next section discusses their

[2] In the following we will use *termination* as synonym for *proper termination*.

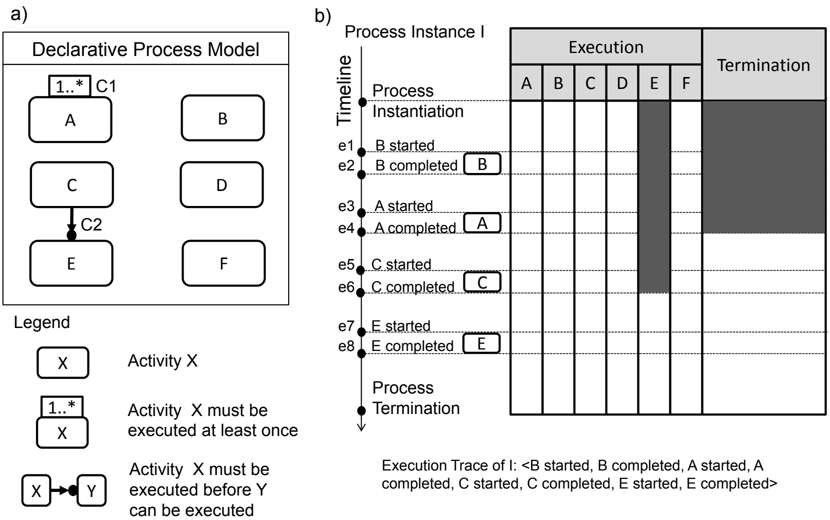


Fig. 1. Executing a declarative process

possible use, and then we turn to discuss their possible effect on model understanding. To our knowledge, the semantics of declarative sub-processes have not been discussed explicitly yet, but their use has been suggested in the context of imperative-declarative model combinations [13]. In general, a sub-process is introduced in a process model via a *complex activity*, which refers to a process model. When the complex activity is executed, the referred process model, i.e., the sub-process, is instantiated. Thereby, sub-processes are viewed as *individual* process instances, i.e., when a complex activity is started, a new instance of the sub-process the complex activity is referring to is created (cf. [13]). The parent process, however, has no information about the internals of the sub-process, i.e., the sub-process is executed in isolation. Communication with the parent process is done only via the sub-process' life-cycle[3]. Thereby, the life-cycle state of the complex activity reflects the state of the sub-process [13], e.g., when the complex activity is in state *completed*, also the sub-process must be in state *completed*.

Considering this, it is essential that sub-processes are executed in isolation, as isolation forbids that constraints can be specified between activities included in different sub-processes. In other words, in a hierarchical declarative process model with several layers of hierarchy, the constraints of a process model *can neither directly* influence the control flow of any parent process, *nor directly influence* the control flow of any sub-process on a layer below. Please note that control flow may still be *indirectly influenced* by restricting the execution of a sub-process, thereby restricting the execution of the activities contained therein.

[3] We do not take into account communication via input- and output data here, as we focus on control flow behavior only.

To illustrate these concepts, consider the process model in Fig. 2 a). It consists of activity A and complex activity B, which in turn contains activities C and D. C and D are connected by a precedence constraint, i.e., D can only be executed if C was executed before. Fig. 2 b) shows an example of a process instance that is executed on this process model. On the left a timeline lists all events that occur during the process execution, e.g., starting or completing an activity. To the right, the enablement of the activities is illustrated. Whenever the area below an activity is colored white, it indicates that this activity is currently enabled. The timeline is to be interpreted the following way: By instantiating the process, activities A and B become enabled, as no constraints restrict their execution. C and D cannot be executed, as they are confined in complex activity B and no instance of B is running yet. Also the subsequent execution of A (*e1*, *e2*) does not change activity enablement. However, with the start of B (*c3*), C becomes enabled, as it can be executed within the new instance of B. Still, D is not enabled yet as the precedence constraint is not satisfied. After C is executed (*e4*, *e5*), the precedence constraint is satisfied, therefore also D becomes enabled. After the execution of D (*e6*, *e7*), the user decides to complete sub-process B (*e8*). Hence, C and D cannot be executed anymore. Still, A and B are enabled as they can be executed directly within process instance I. Finally, after the process instance is completed by the end user through explicit termination, no activity is enabled anymore.

4 Using Hierarchy in Declarative Models

Regardless of the modeling language, hierarchy allows to structure models and to hide modeling elements in sub-models. In this section, the use of hierarchy, given the semantics of Section 3, is discussed.

4.1 Running Example

To illustrate and discuss the implications of hierarchy on declarative process models, we make use of a running example. We chose the business process of writing a scientific paper and created two business process *models* describing the process. In Fig. 3 the process is modeled without hierarchy, whereas in Fig. 4 hierarchical structures are used. Due to space restrictions, the examples are not described in detail, but will be used in the following for illustration purposes.

4.2 Preconditions for Using Sub-Processes

While for imperative models any Single-Entry-Single-Exit fragment can be extracted to a sub-process [14], [15], in declarative models the structure is not informative enough. Rather, two main conditions should hold for the introduction of sub-processes. First, the activities in a sub-process should relate to a certain intention [16] to be fulfilled.[4] For instance, in Fig. 4, *Read reviews for*

[4] Formalizing intentions in declarative process models is out of scope for this work and will be left for future research.

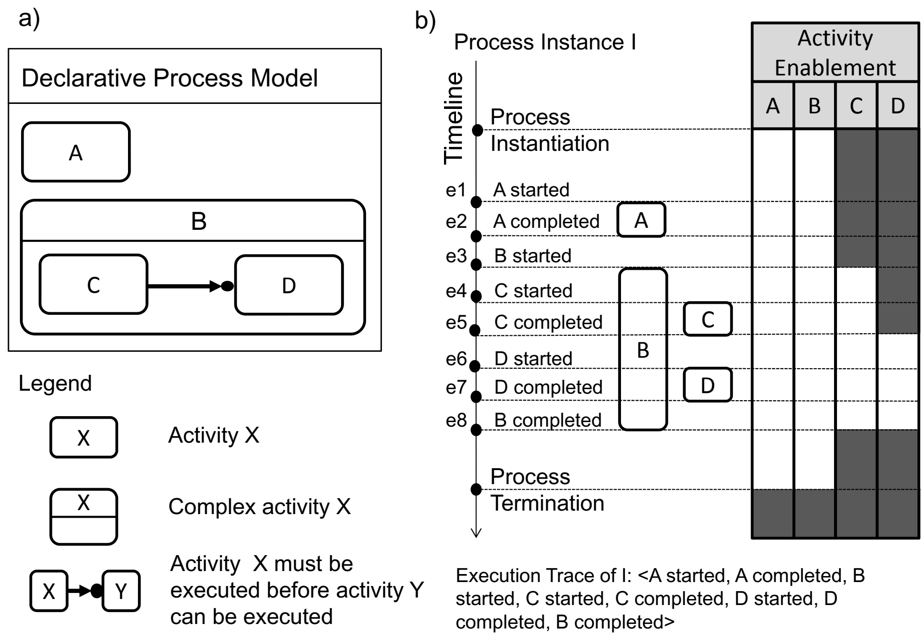


Fig. 2. Execution of a sub-process

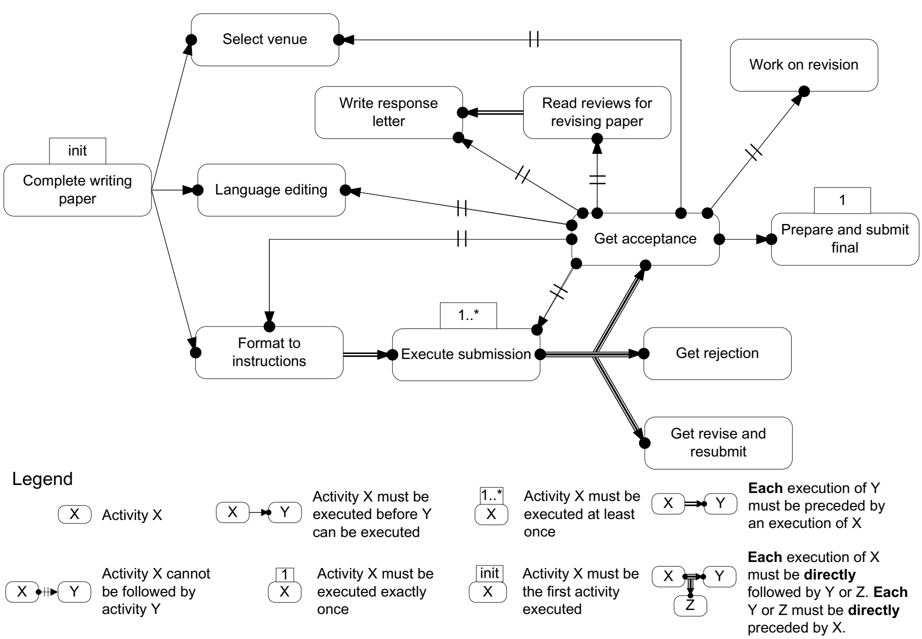


Fig. 3. Example of a flat model

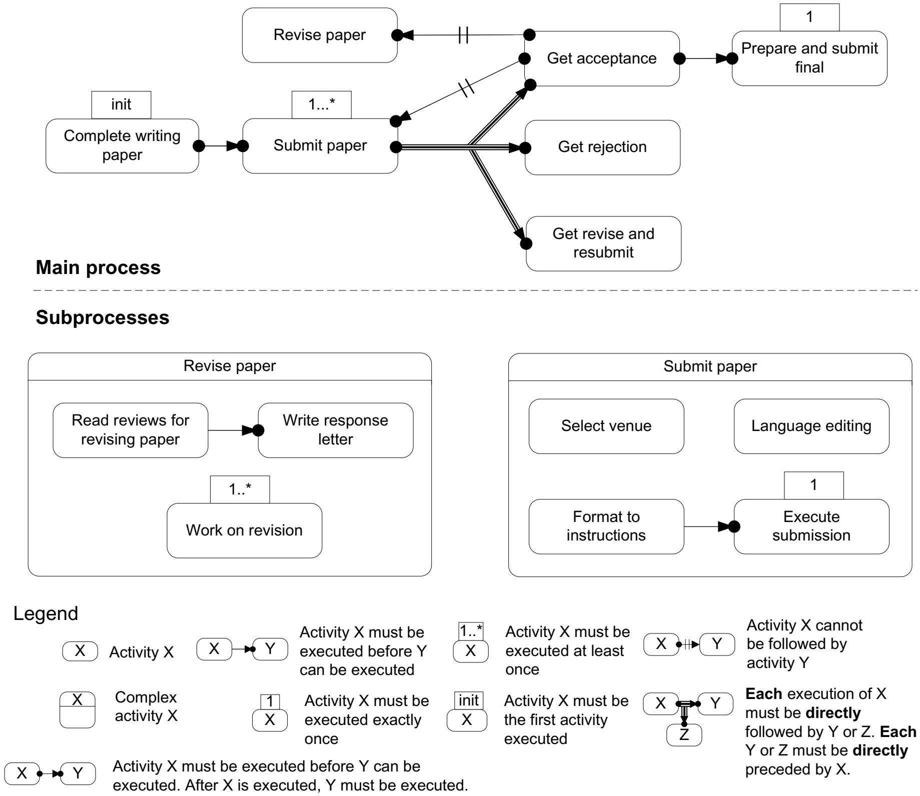


Fig. 4. Example of a hierarchical model

revising paper, *Write response letter* and *Work on revision* all serve the purpose of revising a paper. Once the sub-process of *Revise paper* is completed, it is clear that the paper has been revised. On a higher abstraction level it may not make a difference, e.g., how many times *Work on revision* has been executed or whether the reviews have been read. But knowing the paper has been revised is substantial for the continuation of the process. This information is not available in the flat model (and it only exists in the mind of the human who executes the process). Second, the activities included in a sub-process should be such that they can be executed in isolation from the top-level process. This is due to the local nature of the constraints within the sub-process, and the lack of communication with the parent process, as discussed in Section 3. In other words, a sub-process cannot include any activity that has constraints specifically relating that activity to activities outside the sub-process. Still, if all the activities considered for inclusion in a sub-process share a common constraint with some other activity, then this constraint holds for the entire sub-process. In the flat model (cf. Fig. 3), activities *Read reviews for revising paper*, *Write response letter* and *Work on revision* all have a constraint restricting them from following

Get acceptance. In the hierarchical model (cf. Fig. 4), these constraints are *aggregated* to one constraint related to the top-level complex activity of *Revise paper*. As the constraints are aggregated so a single constraint, we refer this to as *aggregation of constraints*.

4.3 Enhanced Expressiveness

For *imperative* process models, hierarchical decomposition is viewed as a structural measure that may impact model understandability [17], but does not influence semantics. In declarative process models, however, hierarchy also has implications on semantics. More precisely, hierarchy enhances the expressiveness of a declarative modeling language. The key observation is that by specifying constraints that refer to complex activities it is possible to restrict the life-cycle of a sub-process. A constraint that refers to a complex activity thereby not only influences the complex activity, but also all activities contained therein.

This, in turn leads to two effects. First, constraints can be specified that apply for a set of activities (cf. aggregation of constraints in Section 4.2). Second, the specification of constraints, that apply in a certain context only, is supported. Consider for instance *Work on revision* and *Revise paper* in Fig. 4. *Work on revision* is mandatory within the context of *Revise paper*. Hence, *Work on revision* must be executed at least once whenever *Revise paper* is executed, but it might not be executed at all (if *Revise paper* is not executed).

To illustrate how these two effects enhance expressiveness, consider model M in Fig. 5, which solely uses constraints defined in [12]. The chained precedence constraint between C and D specifies that for each execution of D, sub-process C has to be executed directly before. When executing sub-process C, in turn, A has to be executed exactly once and B has to be executed exactly twice (in any order). Hence, the constraint between C and D actually refers to a *set of activities*. For each execution of D, A has to be executed exactly once and B has to be executed exactly twice. In other words, constraints on A and B are only valid in the *context of* C. Such behavior cannot be modeled without hierarchy, using the same set of constraints.

4.4 Impact on Adaptation

Constructing hierarchical models supports top-down analysis, i.e., creating the top-level model first and further refining complex activities thereafter. While this seems like a natural way of dealing with complexity, in some cases, it is desirable to transform a flat model to a hierarchical one. In the following we will argue why refactoring [14], i.e., changing hierarchical structures in a control-flow preserving way, is only possible under certain conditions for declarative process models. Refactoring requires that *any* hierarchical model can be translated into a model without hierarchy, but the same control-flow behavior (and vice versa). As discussed, expressiveness is enhanced by hierarchy. In other words, there exists control flow behavior that can be expressed in an hierarchical model, but

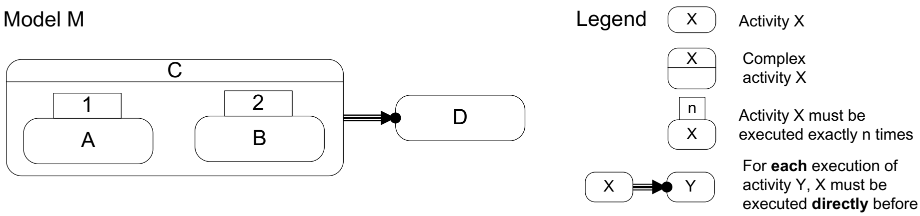


Fig. 5. Enhanced expressiveness

not in a model without hierarchy—cf. Fig. 5 for an example. Hence, only those hierarchical models that do not make use of the enhanced expressiveness can be refactored.

5 Model Understandability

So far we discussed that hierarchy in declarative process models is not just a question of structure, but also affects semantics. In the following, we will describe how these effects impact the understandability of a declarative process model.

5.1 Framework for Assessing Understandability

The influence of hierarchy on model understandability has been investigated in a number of different modeling languages, such as ER-Models [18], imperative business process models [8] and UML Statecharts [10] (for an overview see [17]). While reported results do not entirely clarify when and how understandability is affected, a trade-off between (sub)model size and degree of hierarchy can be observed. For instance, in small models hierarchy may have no [10] or even a negative impact [9], while for large models a positive influence could be observed [8].

In [17], we introduced a cognitive-psychology-based theory describing when and why hierarchy has an impact on understandability (for a introduction to cognitive psychology in business process modeling we refer to [19]). In this work we present an enhanced version that is still generic but also takes into account the idiosyncrasies of hierarchy in declarative process models. The central concept of the framework is *mental effort* [20], i.e., the mental resources required to solve a problem. In the context of this work, solving a problem refers to understanding the semantics of a declarative process model, i.e., answering questions about a model. According to the framework, hierarchy is the source of two opposing forces influencing this problem solving process. Positively, *abstraction* decreases mental effort by *hiding information* and supporting the *recognition of patterns*. Negatively, *fragmentation* increases mental effort by forcing the reader to *switch attention between fragments* and *integrating* information from fragments.

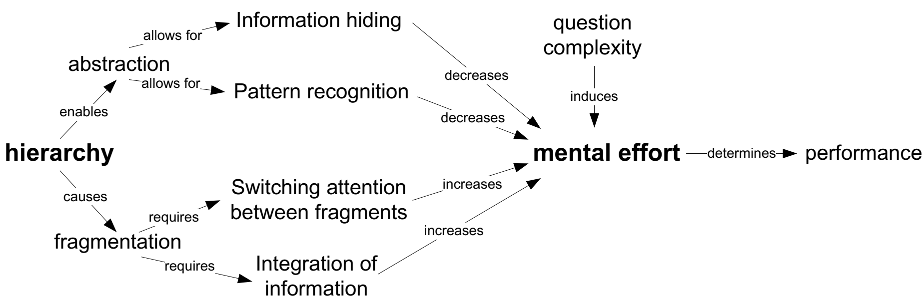


Fig. 6. Framework for assessing hierarchy, adapted from [17]

Abstraction. Hierarchy allows to aggregate model information by hiding the internals of a sub-process using a complex activity. Thereby, irrelevant information can be hidden from the reader, leading to decreased mental effort, as argued in [18]. From the perspective of cognitive psychology, this phenomenon can be explained by the concept of *attention management* [21]. During the problem solving process, i.e., answering a question about a model, attention needs to be guided to certain parts of a model. For instance, when checking whether a certain execution trace is supported by a process model, activities that are not contained in the trace are irrelevant for answering the question. Here, abstraction allows removing this irrelevant information, in turn supporting the attention management system and thus reducing mental effort. To illustrate this effect for declarative process models, consider the process model shown in Fig. 4. For answering the question, whether *Get acceptance* can be executed after *Complete writing paper* it is sufficient to look at activities *Complete writing paper*, *Submit paper* and *Get acceptance*. In other words, hierarchy helps to abstract from all activities contained in *Submit paper*, making the question easier to answer.

Besides reducing mental effort by improving attention management, abstraction presumably supports the identification of higher level patterns. It is known that the human's perceptual system requires little mental effort for recognizing certain patterns [21], [22], e.g., recognizing a well-known person does not require thinking, rather this information can be directly *perceived*. Similarly, in process models, by abstracting and thereby aggregating information, presumably information can be easier perceived. Consider for example the process models depicted in Fig. 3 and Fig. 4. The models are (almost) information equivalent, still we argue that for the model with sub-processes the overall structure and intention of the process is easier to grasp. By introducing complex activities, it is easier to see that the process is about iteratively reworking a paper until it gets accepted. For the sibling-model in Fig. 3, however, the reader first has to mentally group together activities before the overall intention of the process becomes clear.

Fragmentation. Empirical evidence shows that the influence of hierarchy can range from positive over neutral to negative (cf. [8], [9], [10], [18]). To explain the negative influence, we refer to the *fragmentation* of the model. When extracting a sub-process, modeling elements are removed from the parent model and placed within the sub-process. When answering a question that also refers to the content of a sub-process, the reader has to *switch attention* between the parent model and the sub-process. In addition, the reader has to *mentally integrate* the sub-process into the parent model, i.e., interpreting constraints in the context of the parent process. From the perspective of cognitive psychology, these phenomena are known to increase mental effort and referred to as *split-attention effect* [23]. To exemplify this effect, consider the process model in Fig. 4. To determine how often activity *Execute submission* must be executed, it is required to look at activity *Submit paper* too, as *Execute submission* is contained therein. In other words, the reader has to split attention between these two activities. In addition, the reader has to integrate the execution semantics of *Submit paper* with the execution semantics of *Execute submission*. Both activities are mandatory, i.e., must be executed at least once, hence for any execution of the overall process, *Execute submission* must be executed at least once. In other words, it is necessary to mentally integrate the constraints restricting the execution of *Submit paper* as well as constraints restricting the execution of *Execute submission*.

Interplay of Abstraction and Fragmentation. According to the model illustrated in Fig. 6, a question's complexity induces a certain mental effort, e.g., locating an activity is easier than validating an execution trace. In addition, mental effort may be decreased by information hiding and pattern recognition, or increased by the need to switch between sub-processes and integrate information. Thereby, abstraction as well as fragmentation occur at the same time. A model without sub-processes apparently cannot benefit from abstraction, neither is it impacted by fragmentation. By introducing hierarchy, i.e., creating sub-processes, *both* abstraction and fragmentation are stimulated. Whether the introduction of a new sub-process influences understandability positively or negatively then depends on whether the influence of abstraction or fragmentation predominates. For instance, when introducing hierarchy in a small process model, not too much influence of abstraction can be expected, as the model is small anyway. However, fragmentation will appear, regardless of model size. In other words, hierarchy will most likely show a negative influence or at best no influence for small models (cf. [9], [24]).

5.2 Impact of Idiosyncrasies on Understandability

In Section 4, we have shown that hierarchy enhances expressiveness and allows to aggregate constraints. In the following, we will discuss the impact of these two phenomena on understandability.

Enhanced Expressiveness and Complex Mental Integration. As argued, hierarchy provides enhanced expressiveness. However, this also comes at a price, as the constraint that is referring to a sub-process has to be integrated with the semantics of the constraints *within* the sub-process. To illustrate such integrations, consider the process model in Fig. 4. Activity *Work on revision* has to be executed at least once, i.e., is mandatory. However, this activity is contained in complex activity *Revise paper*, which is optional. In other words, *Work on revision* is mandatory for *Revise paper*, which is optional for the main process. Consequently, also *Work on revision* is optional for the main process.

Such mental integrations can be found in any hierarchical conceptual model. For instance, in an imperative process model, mental integration refers to transferring the token from the parent process to the start event of the sub-process. As argued, however, integrations are particularly complex in declarative process models. Hence, it can be expected that a strong influence on the understandability can be observed.

Aggregation of Constraints. As discussed in Section 4.3, hierarchy allows to aggregate and thus reduce the number of constraints. In the context of the proposed framework, we can identify three forces. Positively, aggregation reduces the number of constraints, hence hiding information. In addition, a reduced number of constraints fosters the the layout of the process model. This, in turn, supports the recognition of patterns, i.e., making the model easier to understand. Negatively, complex mental integration operations, as discussed before, may diminish the described gains. Whether positive or negative influences predominate will have to be investigated empirically, as discussed in the following.

5.3 Discussion

So far we argued that hierarchy in declarative process models can be attributed to increases as well as decreases in understandability. In the following, we will discuss the impact of the identified influences. Positively, we see a big potential for hierarchy in declarative process models. In an imperative process model, control flow is modeled explicitly. Hence, process models are usually structured according to their control flow. Such a strategy is in general not possible for a declarative process model, as constraints do not necessarily prescribe sequential information. Sub-processes, however, allow to group activities and thereby to introduce structure to the model. Sub-processes, however, allow to group activities by a mutual intention they serve and thereby to introduce structure to the model and add higher-level information. As argued in our framework, this allows *recognizing patterns* and makes it easier to grasp the intention of a business process (cf. Fig. 3 and Fig. 4). Also the ability of sub-processes to *hide information*, i.e., activities and constraints, can be expected to contribute to the understandability of models. It is assumed that several interconnected constraints quickly become challenging for the human mind [11], [12], [25]. Hence, hiding information can be expected to be especially beneficial in declarative process models.

On the other hand, as argued in Section 5.2, we assume that the integration of constraints poses a significant challenge for the reader. In particular, it is not clear yet whether an average process modeler is able to efficiently perform such mental integrations. This is, however, necessary for the meaningful application of enhanced expressiveness by hierarchy. If efficient mental integration was not possible, enhanced expressiveness would be rendered useless as resulting models would be hardly understandable.

The presented framework can be seen as a first step towards a systematic assessment of the impact of hierarchy on understandability in declarative process models. Even though it is based upon well-established concepts from cognitive psychology, the claims still have to be empirically challenged. In particular, we postulated that the integration of constraints poses a significant, but manageable challenge for the reader. Similarly, we assume that *large* declarative process models tend to be too complex for humans to deal with (cf. [12]). To corroborate the postulated claims, we are currently planning a thorough empirical investigation, cf. Section 7. Therein, we plan to assess understandability in two ways. First, we assume that the easier the understanding of a model is, the less mental effort is required. Hence, we will use Likert-scales to assess mental effort (for the reliability of such measures see [26]). Second, and similar to [8], [9], [24], we will measure the ratio of correctly answered questions as well as the required duration. Apparently, hierarchical models will not be allowed to use constructs that go beyond the expressiveness for non-hierarchical models. Otherwise, information equivalency is not given anymore, imparing the experimental setup. Due to this reason, we plan to conduct experiments for such models first and hope to be able to extrapolate findings to expressiveness-enhanced models.

6 Related Work

In this work we discussed characteristics of hierarchy in declarative process models and the impact on understandability. The impact of hierarchy on understandability has been studied in various conceptual modeling languages, such as imperative business process models [8], ER diagrams [18], [27] and UML statechart diagrams [9], [24], [28] (an overview is presented in [17]). Still, none of these works deals with the impact of hierarchy on understandability in declarative process models. The understandability of declarative process models in general has been investigated in the work of Zugal et al. [11], [29], [30], however, in contrast to this work, hierarchy is not discussed. With respect to understandability of process models in general, work dealing with the understandability of imperative business process models is related. In [6] modeling guidelines are presented that target to improve the understandability of imperative process models. The understandability of imperative process models is investigated empirically in [31]. Finally, in [12] the technical aspects of declarative business process models, such as the definition of modeling languages or verification of models is investigated. In contrast to this work, understandability aspects are neglected and the unique semantics and expressiveness enabled by sub-processes is not elaborated.

7 Summary and Outlook

In this work we examined hierarchy in declarative business process models. After elaborating on the semantics, we discussed the usage and peculiarities of hierarchy. In particular, we showed that hierarchy enhances expressiveness, but cannot be used arbitrarily to any model fragment. Subsequently, we discussed implications on the understandability of declarative process models. Thereby, we built upon previous work and proposed a cognitive-theory based framework to systematically assess the impact of hierarchy on understandability in declarative process models. In general it can be said that hierarchy should be handled with care. On the one hand, information hiding and increased pattern recognition promise gains in terms of understandability. On the other hand, the integration of constraints presumably poses a significant challenge for the reader. In addition, switching between sub-processes may compromise the understandability of respective models. We acknowledge that, even though the framework is based on well-established concepts from cognitive psychology, an empirical validation still has to be conducted.

In this sense, our next steps clearly focus on empirical validation. In particular, two main research directions are envisioned. First, we will investigate whether information hiding lowers cognitive load and hence improves understandability. Second, the integration of constraints and the required mental effort will be scrutinized.

References

1. Parnas, D.L.: On the Criteria to be Used in Decomposing Systems into Modules. Communications of the ACM 15, 1053–1058 (1972)
2. Goguen, J.A., Varela, F.J.: Systems and Distinctions; Duality and Complementarity. Int. J. Gen. Syst. 5, 31–43 (1979)
3. Damij, N.: Business process modelling using diagrammatic and tabular techniques. Business Process Management Journal 13, 70–90 (2007)
4. Sharp, A., McDermott, P.: Workow Modeling: Tools for Process Improvement and Application Development. Artech House (2011)
5. Kock, N.F.: Product flow, breadth and complexity of business processes: An empirical study of 15 business processes in three organizations. Business Process Reengineering & Management Journal 2, 8–22 (1996)
6. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information & Software Technology 52, 127–136 (2010)
7. Pesic, M., Schonenberg, H., Sidorova, N., van der Aalst, W.: Constraint-Based Workflow Models: Change Made Easy. In: Proc. CoopIS 2007, pp. 77–94 (2007)
8. Reijers, H., Mendling, J., Dijkman, R.: Human and automatic modularizations of process models to enhance their comprehension. Inf. Systems 36, 881–897 (2011)
9. Cruz-Lemus, J.A., Genero, M., Piattini, M.: Using Controlled Experiments for Validating UML Statechart Diagrams Measures. In: Cuadrado-Gallego, J.J., Braungarten, R., Dumke, R.R., Abran, A. (eds.) IWSM-Mensura 2007. LNCS, vol. 4895, pp. 129–138. Springer, Heidelberg (2008)
10. Cruz-Lemus, J., Genero, M., Piattini, M., Toval, A.: Investigating the nesting level of composite states in uml statechart diagrams. In: Proc. QAOOSE 2005, pp. 97–108 (2005)
11. Zugal, S., Pinggera, J., Weber, B.: Toward Enhanced Life-Cycle Support for Declarative Processes. Journal of Software: Evolution and Process 24, 285–302 (2012)
12. Pesic, M.: Constraint-Based Workflow Management Systems: Shifting Control to Users. PhD thesis, TU Eindhoven (2008)
13. Pesic, M., Schonenberg, H., van der Aalst, W.: DECLARE: Full Support for Loosely-Structured Processes. In: Proc. EDOC 2007, pp. 287–298 (2007)

14. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry 62, 467–486 (2011)
15. Weber, B., Reichert, M., Rinderle, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. DKE 66, 438–466 (2008)
16. Soffer, P., Rolland, C.: Combining Intention-Oriented and State-Based Process Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 47–62. Springer, Heidelberg (2005)
17. Zugal, S., Pinggera, J., Weber, B., Mendling, J., Reijers, H.A.: Assessing the Impact of Hierarchy on Model Understandability—A Cognitive Perspective. In: Proc. EESSMod 2011, pp. 18–27 (2011)
18. Moody, D.L.: Cognitive Load Effects on End User Understanding of Conceptual Models: An Experimental Analysis. In: Benczúr, A.A., Demetrovics, J., Gottlob, G. (eds.) ADBIS 2004. LNCS, vol. 3255, pp. 129–143. Springer, Heidelberg (2004)
19. Zugal, S., Pinggera, J., Weber, B.: Assessing process models with cognitive psychology. In: Proc. EMISA 2011, pp. 177–182 (2011)
20. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12, 257–285 (1988)
21. Larkin, J.H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11, 65–100 (1987)
22. Scaife, M., Rogers, Y.: External cognition: how do graphical representations work? Int.J. Human-Computer Studies 45, 185–213 (1996)
23. Sweller, J., Chandler, P.: Why Some Material Is Difficult to Learn. Cognition and Instruction 12, 185–233 (1994)
24. Cruz-Lemus, J.A., Genero, M., Manso, M.E., Morasca, S., Piattini, M.: Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies. Empir. Software Eng. 25, 685–719 (2009)
25. Weber, B., Reijers, H.A., Zugal, S., Wild, W.: The Declarative Approach to Business Process Execution: An Empirical Test. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 470–485. Springer, Heidelberg (2009)
26. Paas, F., Tuovinen, J.E., Tabbers, H., Gerven, P.W.M.V.: Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. Educational Psychologist 38, 63–71 (2003)
27. Shoval, P., Danoch, R., Balabam, M.: Hierarchical entity-relationship diagrams: the model, method of creation and experimental evaluation. Requirements Engineering 9, 217–228 (2004)
28. Cruz-Lemus, J.A., Genero, M., Morasca, S., Piattini, M.: Using Practitioners for Assessing the Understandability of UML Statechart Diagrams with Composite States. 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. 213–222. Springer, Heidelberg (2007)
29. Zugal, S., Pinggera, J., Weber, B.: The Impact of Testcases on the Maintainability of Declarative Process Models. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 163–177. Springer, Heidelberg (2011)
30. Zugal, S., Pinggera, J., Weber, B.: Creating Declarative Process Models Using Test Driven Modeling Suite. In: Nurcan, S. (ed.) CAiSE Forum 2011. LNBIP, vol. 107, pp. 16–32. Springer, Heidelberg (2012)
31. Reijers, H.A., Mendling, J.: A Study into the Factors that Influence the Understandability of Business Process Models. SMCA 41, 449–462 (2011)

Integrating Ordinary Users into Process Management: Towards Implementing Bottom-Up, People-Centric BPM

Michael Prilla and Alexander Nolte

Department Information and Technology Management, Institute for Applied Work Science, University of Bochum, Germany
{michael.prilla,alexander.nolte}@iaw.rub.de

Abstract. Despite its increasing success in organizations, traditional BPM embodies a top-down approach performed by a small group of experts, limiting process stakeholders to part-time information providers, hindering proactive contributions. In this paper, we argue that BPM can benefit from being complemented with a bottom-up and people-centric strategy, allowing for interventions by process stakeholders. However, this cannot be realized by turning ordinary users into BPM or modeling experts. Instead, there is a need to find appropriate means to engage these people into BPM, process development and modeling. In this paper, we present two explorative empirical studies exploring such means. As a result of analyzing these studies, we present five proposals towards the implementation of stakeholder involvement. Our work does not want to replace existing BPM procedures, but to complement them. Thus, it is a starting point for further research and as an opportunity to join forces with other researchers pursuing similar goals.

1 Introduction: BPM and Stakeholder Integration

Business Process Management (BPM) is a field of interest both in practice and science. It aims at identifying, capturing and formalizing business processes in organizations and supports the execution of them, including monitoring and control. Its popularity is mirrored by a vivid research community as well as by its dissemination in practice.

Notwithstanding the success and value of BPM, there are voices claiming that currently BPM relies too much on experts, making it an activity performed by a few while affecting many actors in organizations. As a result of that, process specifications often differ from real processes as perceived by process stakeholders, who are **non-experts** in BPM, and thus do not represent existing work practice (e.g. [1]). As this paper argues, this problem can be tackled by a better **integration of process stakeholders** into process management, enabling them to proactively[1] contribute to process development. Examples of such contributions are information on process exceptions, workarounds in daily practice or changes to improve a process.

[1] The term 'proactive', refers to tasks performed by people themselves on their own accord – as opposed to tasks where people only provide information.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 182–194, 2012.

This paper describes explorative empirical[2] work on bottom-up, people-centric BPM and diminishing the gap between BPM experts and process stakeholders. This, borrowing a phrase from web 2.0, transforms stakeholders from process consumers into **"process prosumers"** using processes and being involved in their production.

This papers contribution is threefold. First, it provides insights into the interaction of users who are no modeling experts with process models and thus transcends existing knowledge on this issue. Second, our insights challenge existing assumptions and propositions concerning stakeholder integration into BPM. Third, the paper proposes steps towards the implementation of stakeholder integration and bottom-up BPM, following the goal to **enhance quality of and commitment to process models** (and thus, to represented processes) as a result of stakeholder integration. Summing up, the exploratory work described in this paper provides a starting point for research and development concerning stakeholder integration and bottom-up strategies into BPM without providing proven results.

In the remainder of this paper, we describe our idea of bottom-up, people-centric BPM (section 2). In section 3, we present two empirical studies conducted to explore the interaction of non-expert modelers with process models. In section 4, we present five proposals for the implementation of stakeholder involvement into BPM. Section 5 compares our approach with existing work. The paper concludes with a summary and an outlook on further work.

2 Bottom-Up, People-Centric Processes in BPM: A Comparison

Currently there is a discussion in the BPM community of how to integrate stakeholders properly (e.g. [1, 2]). However, this discussion is still in an early state, as there is surprisingly little research on issues such as process model usage by users who are no modeling experts (cf. [3]). Therefore, many questions remain: What do processes mean to people and how can they deal with process models? To which extent can non-expert modelers actively contribute to BPM? What are the barriers causing the lack of integration of these people into BPM?

Existing approaches of stakeholder integration try to answer these questions by using existing knowledge for process development [4], getting process feedback from users [5], making process descriptions more understandable [6] or helping people to comment on processes [1]. In this paper, we present an exploratory empirical approach towards **bottom-up, people-centric BPM**.

Currently BPM uses a **top-down** approach employing experts to identify and specify high-quality processes for execution (cf. [7]). This makes process management and execution dependent on a small group of analysts, consultants and managers as the skills to adequately specify and run processes are too big a burden for ordinary users. Therefore, stakeholders have to rely on experts to ask them for contributions and integrate these contributions into process documentation. This might be a sufficient approach when models are fed back to stakeholders in the form of e.g. software prototypes, which can still be adapted, but if they remain in the organization, describing

[2] By "empirical", we refer to an exploratory approach using interviews and experiments.

how procedures are conducted, active contribution is needed to ensure quality and commitment. Using a **bottom-up** approach enables such active contribution (see **Fig. 1** for a comparison between top-down and bottom-up strategies in BPM).

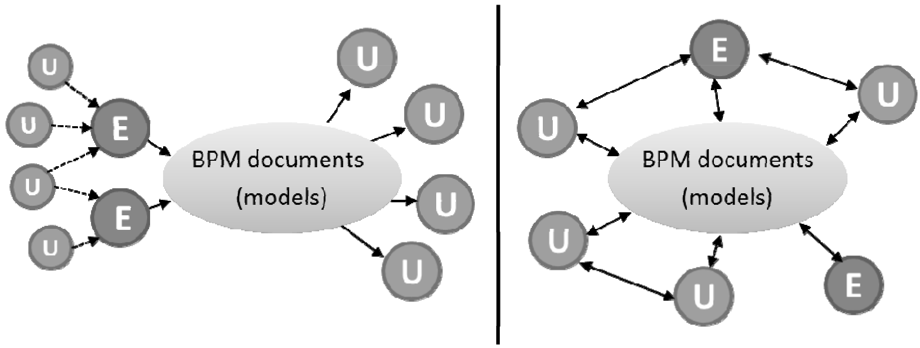


Fig. 1. Top-down BPM with users (U) informing experts (E), who specify processes and roll them out to users (AS-IS situation, top). In people-centric, bottom-up BPM both users and experts can develop and consume processes (TO-BE situation, bottom).

BPM is mainly **expert-driven**. Stakeholder involvement is limited to phases of analysis and (maybe) design, there is no opportunity for stakeholders to intervene or control [8]. In other words, process stakeholders have to wait until experts think it is their turn to speak up. This decouples them from process development and thus slows down change processes. In contrast, a **people-centric** approach would actively include stakeholders and their contributions, diminishing the gap between stakeholders and BPM experts, thus speeding up changes in processes and improving the quality of documentation. Of course, this needs transparency of processes and their development.

Models are central artifacts for documentation and analysis in BPM. In other words, contributing to process models means contributing to BPM. Models, however, are usually artifacts of BPM experts and are not available for users. Therefore, only little is known about the interaction of ordinary users with models in practice.

Additionally, adequate tool support is required as BPM systems and modeling tools are typically built to suit the needs of BPM experts. Consequently, they are hard to use for most process stakeholders and sometimes not even accessible to them. Therefore, the implementation of bottom-up and people-centric BPM needs a holistic approach, employing both technical and organizational changes.

3 A Walk on the Wild (User) Side: Two Empirical Studies for the Exploration of Stakeholder Involvement into BPM

In order to explore means and potentials of stakeholder integration into BPM, we conducted two empirical studies. In contrast to other approaches (see section 5), our work explicitly addresses other actors than BPM experts. We used two important prerequisites in these studies. First, we did not require non-expert modelers to learn a process modeling language, as in practice most ordinary users will not do that because of the complexity of modeling languages [9]. Our experiences from modeling workshops, in which participants were asked to contribute verbally to models, show that users gain a sufficient understanding of a modeling language after a short period of time (e.g. [10, 11]). Second, we wanted to enable people to contribute to process development with as little guidance as possible, as guidance provides a bottleneck to stakeholder integration. Based on these prerequisites, the first step to understand the interaction of non-modelers with process models is an investigation of this interaction. This needs an explorative research approach, implying that our work can only provide proposals, but no final, proven solutions.

We are aware of the fact that our ambition sounds paradox: How can we explore an interaction that does rarely take place yet? Obviously, asking non-expert modelers about their usage of models is not sufficient, as they currently do not use them in practice yet. We conducted interviews with process experts, asking them to describe the usage of processes and process models by non-expert users (see section 3.1). From these, we gained insights into the current practice of non-expert model interaction. Second, we conducted an experiment in which users specified process elements with a text-based interface and interacted with resulting models (see section 3.2). This enabled us to observe users' behavior toward process models in a controlled setting.

3.1 Study 1: Non-expert Usage of Process Models in Practice

In a first study to explore non-expert interaction with process models and process management, we conducted a series of interviews with BPM practitioners working closely with process stakeholders. We chose these people to provide information on the target user group and their behavior as only experts can provide an overview of non-experts' interaction with models in practice. An analysis of these insights allowed us to understand the usage of models by non-expert modelers and derive measures for stakeholder integration.

Table 1. Participants in study 1, including affiliation and role in the company, main purpose of model usage (information given by the participants) and notations used

Partner	Company	Role	Purpose of Modeling	Notations
I1	Technical Supp.	Project manager	Project planning	Flowcharts
I2	Energy supply	Analyst	Process improvem.	BPMN/UML
I3	IT-Consulting	Project manager	Project planning	UML
I4	IT-Consulting	Req. Analyst	Requirements	BPMN
I5	Software Dev.	Coordinator	Development	UML
I6	Software Dev.	Req. Analyst	Requirements	UML

Course of the Study. The study was conducted by a series of six interviews with BPM experts from five different companies (see Table 1 for the variation in domains). The interviewees had more than ten years of experience in process management and modeling. In their daily work, they fulfilled different roles and serve as facilitators of modeling workshops (especially the project managers), process or requirements analysts and project managers. All of them work closely with stakeholders on models. As can be seen from Table 1, the participants cover a wide set of modeling purposes and notations used. Each interview lasted about 60 to 90 minutes and covered the whole BPM lifecycle, including questions about the usage of process models by non-expert modelers, the availability of models for them, model exchange between users, information provision during process documentation as well as barriers and success factors of non-expert usage and existing technical or organizational support.

For later analysis, each interview was audio-recorded and transcribed. After that, based on grounded theory [12] to avoid a bias in analysis, we developed a coding scheme out of the transcript material by coding the transcripts in three consecutive loops. In total, we derived a catalogue of 102 (sub-) categories from this, from which we were able to derive 14 use cases and more than 30 requirements for the integration of stakeholders into BPM (cf. [8]). The results of this study as described below include insights providing new opportunities for BPM and show a perspective on the integration of stakeholders, which is usually missing in BPM research.

Stakeholders Are Cut from the Development Cycle of Processes after Their Information on the Process Has Been Captured Once. All interviewees of the study reported that usually users are asked for information, models are created and – if at all – users are asked for feedback on resulting models. This only involves stakeholders in special phases of process development and cuts them from actively contributing any other time. This is also implemented in IT support: Models, as the interviewees stated, reside in special BPM systems or file shares only accessible to BPM experts. There is no access for stakeholders: *"(…) we don't have a document management system for models to search or browse from time to time"*[3] (I3). As a consequence, there is only little proactive feedback from process stakeholders.

People Want to Use Process Models for Communication, but Cannot Get Hold of Them. Process management experts use models as artifacts of knowledge exchange: *"We – my colleagues and me – place them besides each other"* (I2). This also applies in the cooperation among stakeholders: *"(…) by these models, they get a common basis, which they can communicate about."* (I4). This shows the potential of model usage by non-expert modelers, but our interviewees also told us that usage is hindered by lacking awareness of existing models: *"I often think that the diagrams are not enough within reach. They kind of disappear in the depths of IT"* (I2). This shows that availability and awareness are important prerequisites for stakeholder integration. It also shows that current BPM solutions do not make models sufficiently available for stakeholders. Once might argue that if stakeholder ask for models, experts will usually provide them, but this is no sufficient solution: In contrast, it makes users dependent on model provision by experts expert only applies to models users know about – using such solutions, users will never use models they were not aware of.

People Would Participate in Process Development, but Cannot Express Themselves Properly. Our interviewees told us that they usually show process models to stakeholders to assure their quality: "*(…) I showed the model to this person again and*

[3] The interviews were conducted in German and the quotes have been translated by the authors. To guarantee the anonymity of interviewees, we refer to them as "I" with a running number.

explained to him my understanding. And then he either corrected me or approved what I said" (I3). Additionally, they ask stakeholders when they want to change models: *"you ask the person: something is missing here, this area has been left out. And you do that cooperatively"* (I5). This shows that stakeholders depend on experts' triggers for contributions. It also shows that stakeholders are capable of reading processes. However, the interviewees stated that most modeling is done by BPM experts. Non-experts, as they explained, feel uncomfortable to model. This indicates an important difference between **reading models** and **active modeling**: Users can read models, but cannot use modeling language.

People Would Contribute to Process Development, but Expert Guidance and Control Discourage Them from Doing So. Our interviewees told us that *"[models] are usually regarded as my artifact"* (I2). While this is interesting on its own, we found another interesting fact: All interviewees reporting this distance also stated that experts control model development and change: *"(…) we are always present when models are changed. Thus, we can co-decide what is to be included in the model and what isn't"* (I2). This indicates that expert guidance not only drives process development, but also discourages users from active participation. Thus, the influence of BPM experts is decisive and changing it can lead to changing the involvement of stakeholders in BPM. This is not to blame or to get rid of experts but to state that their current role hinders stakeholder integration.

Discussion: Contribution to Existing Research. As mentioned above, there is only little work on different BPM actors and their interaction with process models. Our study stresses the assumption that stakeholders are not well integrated into BPM. This underpins the need for stakeholder integration and its potential to improve process management. According to our observations:

- Users can read and use models without being taught to be modelers or being strictly guided by BPM experts.
- Model usage is present among users in organizations, but contrary to current assumptions BPM does not make them available.
- The role of BPM experts hinders the integration of stakeholders.

We also found that despite their ability to read models, non-expert modelers feel uncomfortable to express themselves in modeling language. Therefore, our work sheds light on a common misunderstanding: The fact that stakeholders do not model actively does not mean that they are not capable of contributing to process management. This raises the question to what extend stakeholders people are able to contribute to models without control of an expert. We investigated this question more thoroughly in a second study.

3.2 Study 2: Non-expert Modelers Specification of Processes

In study 1 we identified the problem that people find it difficult to express themselves in a modeling language. In order to explore this, we created an experimental setting in which we reduced the need to use a modeling language. We provided participants with an easy to use interface originally developed for brainstorming [13]. It allows users to enter contributions into a text input box and send it to a model (see Fig. 2). After it is sent, the contribution becomes a model element labeled according to text input. The idea of this experiment was to explore to what extend users who are no experts in modeling are capable of documenting processes on their own.

Course of the Study. We used two different scenarios representing real-world processes, which participants work in every day and thus should know by heart (cf. Table 2). Each process includes two different roles, being taken by one participant each. We deliberately chose processes that are simple enough to be used in a short experiment like this, but complex enough to have impact in practice, as they specify everyday work. We conducted five sessions with two participants each, lasting for 30-45 minutes. Table 2 gives an overview of the setting.

Besides differences in roles taken and scenarios, the participants can be differentiated by their affinity to modeling. Three of them can be considered modeling experts, two of them use modeling tools occasionally and five participants were new to modeling. In addition, the users of the software in scenario (1) were not trained in formal bug reporting or specification procedures and the users in scenario (2) were ordinary users and had no further insights into processes of a library.

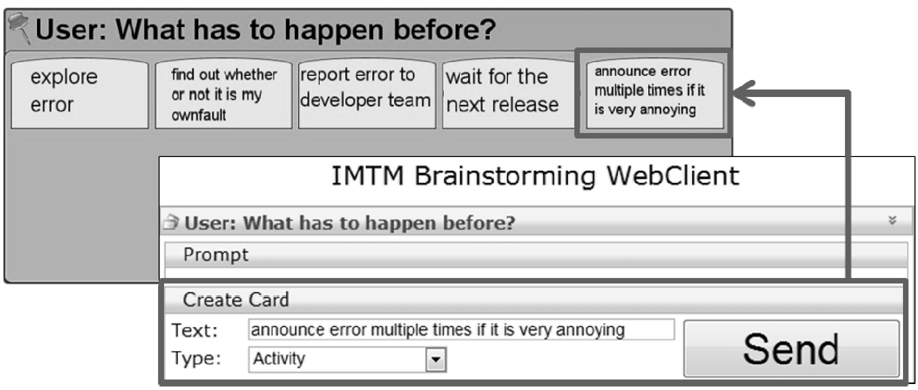


Fig. 2. Contributions via a single text input box in a web interface (bottom) result in an element with the respective label in a modeling tool (top)

In each workshop, participants started contributing tasks to a predefined model template (see Fig. 2) representing their view of the process. After that, they sorted these tasks according to their sequence in the process. Then, they were told to collaboratively identify differences in content (different steps) and sequence (same step, different sequence) of their models. Finally, they were asked to articulate the most significant difference between them. After the workshop, we asked participants to reflect their results. For analysis, we videotaped all activities and an observer made notes. During the experiment, a facilitator was present to explain tasks to the participants, but only served as a guide and did not intervene during the experiment.

Table 2. Scenarios used in the experiment, including roles and participants

Scenario	Roles included	Participants
(1) Bug reporting and solving for software development	User (of the software), Developer	3 users of a tool, 3 developers
(2) Book ordering in a library	User (of the library), Librarian	2 library users (researchers), 2 librarians

Insights: Translation, Adequate Interaction and Modeling in a Sandbox as Success Factors. Our study indicates that given adequate means of interaction, non-expert users are able to contribute to process documentation. In what follows, we will present some insights derived from the study. We are however aware that – due to the exploratory nature of our study – our analysis only provides observations and first insights into the domain and they are not generally applicable.

People Can Contribute to Processes If They Do Not Have to Express Themselves in Modeling Language. In all experiments, modeling experts and non-expert users roughly contributed an equal number of elements to the processes. There were also no big differences in the value of the contributions, as in the discussions, almost every activity was considered useful. In the concluding interviews, most participants reported a number of activities she hadn't thought of but which were present in the perspective of the respective other participant and that they had gained deeper insights into the respective other perspective by discussing the processes. This indicates that without the need to use a modeling language, non-expert modelers can catch up to experts in contributing to process documentation.

People Can Think in Processes. The participants identified only two differences in process sequences during the discussion. This indicates that they were all capable of thinking in correct sequences and thus, that there is also no difference between modeling experts and non-expert users in this ability. In addition, both modeling experts and non-expert users had to apply little changes to their own respective sequences. This leads to the assumption that non-expert modelers as well as modeling experts are capable to think in sequences.

People Are Able to Use Process Models to Analyze Their Work Processes and Reflect on Them. The discussion of differences between the respective sequences helped all participants to reflect on their respective activities in the displayed process. Also, they were able to gain insight into the respective other perspective and to create a shared understanding of the process. Furthermore gathering the process elements helped them to meet at eye level, as one participant told us: *"… being able to create a compressed visualization of the own view made the following discussion much easier…"*. This indicates that people can make sense of process models and use them for discussion and reflection without being guided or facilitated.

Discussion: Contribution to Existing Research. Releasing non-expert modelers from the necessity of translating their thoughts into a modeling language affects their contribution to modeling. Input means such as text input to models diminish the language problem and enable non-experts' contributions to models. Contrary to current beliefs in research our study shows that non-expert modelers are able to become active BPM contributors rather than just information providers. They are able to think in sequences and to reflect on processes they created.

Additionally, given the right means for contribution, non-expert users require only little guidance to create useful process descriptions. When complexity rises we expect BPM experts to still be required, but they can be relieved from the burden to translate every contribution by non-BPM-experts into a modeling language.

4 Five Proposals for the Implementation of Stakeholder Involvement into BPM

Our work indicates that ordinary users (non-experts in modeling) can actively contribute to BPM. Here, we present five proposals showing directions for the implementation of the resulting potentials for BPM.

Make Models Available: Ordinary users – not BPM experts – don't use BPM tools. Simply providing access to these tools will not improve the usage of models, as our studies show: Models would still be perceived as artifacts of process experts and it would remain hard to find process models as they reside in expert-oriented BPM systems.

To make models tools of everyday work, they should be available from existing tools used by non-expert modelers such as corporate knowledge management systems (cf. [8]). This increases the chances for non-experts to find models and perceive them as valuable artifacts.

Redefine Roles in BPM: Besides users becoming active BPM contributors, the role of BPM experts has to change as well (cf. 3.1). They have to become coordinators of process specification rather than controllers, guiding users and compiling their contributions. Given that users can model parts of processes on their own, experts should become model managers rather than translators for every contribution, which not only enable but also encourage stakeholders to become active. However, BPM experts still need to support users in complex situations as well as to ensure model correctness and quality. Taken together, this idea of guided active participation speeds up process development, motivates users' contributions and implements guidance throughout the modeling activity for the production of useful visualizations.

To make these changes happen, corresponding role models and processes have to be developed, implemented and introduced. This can be aided by recently popular tools supporting cooperative modeling, which enable users to actively contribute to models (e.g. [5, 14]).

Provide Suitable Interactions for Non-modelers: Active user participation in modeling requires tools that enable users to contribute without knowing a modeling language. In our work, we found that simple text input interfaces, sequence manipulation by dragging elements and commenting are suitable means for this purpose. To our knowledge, there is no work on similar functions available. Thus, our list needs to be evaluated and extended by further work.

For the implementation of this claim, we suggest building prototypes of modeling tools with suitable interaction means like the one presented in this paper and evaluate their usage. Our list of functionalities as well as work of others (e.g. [1]) provides a good starting point for that.

Make Models Tools of Everyday Use: If process models remain expert artifacts, feedback by process stakeholders will remain low. Therefore, process stakeholders should be encouraged actively to use process models as means for communication with experts and among each other. This depends on the implementation of the first three proposals. In addition, actors such as facilitators and managers should encourage users to use models in daily work and the availability of models should be communicated to users during the whole BPM cycle (see Fig. 3).

Intertwine Top-Down and Bottom-Up Strategies: We suggest complementing the existing BPM cycle [7] with means of user-driven BPM. This means that information proactively provided by users might cause the cycle to pause and prompt experts to go a step back that otherwise would have to wait until the cycle is completed, saving time and effort. This can be achieved by a mixture of facilitated workshops, pro-active user contributions and user involvement between workshops by annotations of process parts. However, this needs users to recognize the benefits models have for them. This, in turn, can only be done by practically making users work with models, applying self-directed strategies of model usage [11]. They need to recognize that their own problems can be solved or diminished by active participation in modeling processes and that the overall performance of their department or organization increases.

5 Related Work

A lot of work deals with topics familiar to ours and served as a basis for our research. However, there is only little work investigating the focal point of this integration: How to involve stakeholders in BPM?

Research on the **participatory design of processes**, puts forward the use of process models as artifacts of communication (e.g. [13]) in facilitated workshops with structured discussions about processes ([10]). This however does not solve the problem of proactive stakeholder integration as these workshops still need to be conducted by experts [15]. In research on **collaborative modeling** the documentation of processes is left to the actors themselves (e.g. [5]).This approach however is targeted towards modeling experts or requires the presence of a mediator helping non-BPM-experts to model, thus being so solution for stakeholder integration. **Model wikis** (e.g. [16]) make participation in BPM easier, but are not well connected to BPM tools and thus will not be used by experts.

The **awareness and availability of models** marks another topic of related work. Early contributions propose model catalogues (e.g. [17]), later ones favor model repositories (e.g. [18]). Recent trends can be found in platforms enabling the contextualization of models [19], the semantic retrieval of models (e.g. [20]) and process publishing via intranet or internet platforms (e.g. [21]). While these approaches work well for BPM actors, for non-BPM-experts they are yet another tool they have difficulties in using. As a consequence, recent contributions favor lightweight solutions such as social tagging (e.g. [2, 8]).

Recently, there have been contributions focusing on **process model usability and understanding** (e.g. [3]). However, these approaches mainly take into account the question of understanding models, whereas we focus on identifying means to contribute to them.

At the edges of BPM, research is concerned with combining knowledge management and BPM (e.g. [4, 22]) or managing knowledge about processes (e.g. [1]). These approaches show the value of stakeholder knowledge for BPM, but are targeted towards stakeholders as passive information providers (see above for disadvantages of this) and none of them shows opportunities for direct user integration into BPM.

The variety of existing work shows that our research is shaped by many different approaches and fits well into it. However, it also shows that active contribution of stakeholders into BPM has not been covered to a sufficient extent yet, leaving a gap that this paper contributes to.

6 Conclusion and Further Work

In this paper we described the benefits of a bottom-up and people-centric approach in BPM. We shed light onto current barriers in BPM and non-expert user interaction with process models and developed an approach of stakeholder integration into BPM.

Our findings stem from two empirical studies. The first one indicates a need for a more user-centric approach in BPM as process models play no significant role in the work practice of most process stakeholders. The second study indicates the ability of users to actively document processes in models given the right means of contribution through suitable tools and methods. To our knowledge, with these studies, we are walking on unknown territory. Due to the exploratory nature of both our studies we are able to make proposals for the interaction of process stakeholders with process models, not to derive general principles. These proposals lead to five proposals to support the implementation of a people-centric approach in BPM.

Our further work will be focused on the exploration of non-expert interaction with process models. This will include the question whether or in which situations a facilitator is needed for process specification and adaptation. Furthermore, we will create corresponding prototypes and evaluate them on a broader basis, including different interaction modes. In the future, we hope to find collaborators on this issue and form a new branch of BPM research.

References

1. Witschel, H.F., Hu, B., Riss, U.V., Thönssen, B., Brun, R., Martin, A., Hinkelmann, K.: A Collaborative Approach to Maturing Process-Related Knowledge. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 343–358. Springer, Heidelberg (2010)
2. Bruno, G., Dengler, F., Jennings, B., Khalaf, R., Nurcan, S., Prilla, M., Sarini, M., Schmidt, R., Silva, R.: Key challenges for enabling agile BPM with social software. Journal of Software Maintenance and Evolution: Research and Practice 23, 297–326 (2011)

3. Mendling, J., Reijers, H., Recker, J.: Activity labeling in process modeling: Empirical insights and recommendations. Information Systems (2010)
4. Jung, J., Choi, I., Song, M.: An integration architecture for knowledge management systems and business process management systems. Computers in Industry 58, 21–34 (2007)
5. Rittgen, P.: Collaborative Modeling - A Design Science Approach. In: Proceedings of the 42nd Hawaii International Conference on System Sciences (2009)
6. Mendling, J., Reijers, H., van der Aalst, W.: Seven process modeling guidelines. Information and Software Technology (2009)
7. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
8. Prilla, M.: Models, Social Tagging and Knowledge Management – A fruitful Combination for Process Improvement. In: Proceedings of 2nd Workshop on Business Process Management and Social Software in Conjunction with the Business Process Management Conference (2009)
9. 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)
10. Herrmann, T.: Systems Design with the Socio-Technical Walkthrough. In: Whitworth, B., de Moor, A. (eds.) Handbook of Research on Socio-Technical Design and Social Networking Systems. Information Science Reference (2009)
11. Prilla, M., Nolte, A.: Fostering self-direction in participatory process design. In: Proceedings of the Participatory Design Conference, PDC 2010 (2010)
12. Strauss, A.L., Corbin, J.M.: Basics of qualitative research: Techniques and procedures for developing grounded theory. Sage Publications (1998)
13. Herrmann, T., Nolte, A., Prilla, M.: The Integration of Awareness, Creativity Support and Collaborative Process Modeling in Colocated Meetings. International Journal on Computer-Supported Cooperative Work (IJCSCW). Special Issue on Awareness (2012)
14. Kolfschoten, G.L., Frantzeskaki, N., de Haan, A., Duivenvoorde, G.P.J., Verbraeck, A.: Collaborative Modeling Lab to Increase Learning Engagement; Comparing Manual modeling with Interactive Whiteboards. In: 42nd Hawaii International Conference on System Sciences, HICSS 2009, pp. 1–8 (2009)
15. Richardson, G.P., Andersen, D.F.: Teamwork in group model building. System Dynamics Review 11, 113–137 (1995)
16. Dengler, F., Lamparter, S., Hefke, M., Abecker, A.: Collaborative Process Development using Semantic MediaWiki. In: Hinkelmann, K., Wache, H. (eds.) Proceedings of the 5th Conference of Professional Knowledge Management, pp. 97–107. Gesellschaft für Informatik, Solothurn (2009)
17. Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C.S., Bernstein, A., Herman, G., Klein, M., O'Donnell, E.: Tools for inventing organizations: toward a handbook of organizational processes. Management Science 45, 425–443 (1999)
18. Markovic, I.: Advanced Querying and Reasoning on Business Process Models. In: Abramowicz, W., Fensel, D. (eds.) Proceedings of the 11th International Conference on Business Information Systems (BIS 2008), pp. 189–200. Springer, Innsbruck (2008)
19. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. International Journal of Business Process Integration and Management 3, 47–60 (2008)

20. Thomas, O., Fellmann, M.: Semantic EPC: Enhancing Process Modeling Using Ontology Languages. In: Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N. (eds.) SBPM. CEUR-WS.org. (2007)
21. De Roure, D., Goble, C., Bhagat, J., Cruickshank, D., Goderis, A., Michaelides, D., Newman, D.: myExperiment: Defining the Social Virtual Research Environment. In: Proceedings of the 2008 Fourth IEEE International Conference on eScience, Indianapolis, Indiana, USA, pp. 182–189. IEEE Computer Society, Washington, DC (2008)
22. Prilla, M., Herrmann, T.: Semantically Integrating Heterogeneous Content: Applying Social Tagging as a Knowledge Management Tool for Process Model Development and Usage. In: Proceedings of IKNOW 2007, pp. 16–24 (2007)

Optimized Time Management for Declarative Workflows

Irene Barba[1], Andreas Lanz[2], Barbara Weber[3], Manfred Reichert[2], and Carmelo Del Valle[1]

[1] Departamento de Lenguajes y Sistemas Informáticos, University of Seville, Spain
{irenebr,carmelo}@us.es

[2] Institute of Databases and Information Systems, Ulm University, Germany
{Andreas.Lanz,Manfred.Reichert}@uni-ulm.de

[3] Department of Computer Science, University of Innsbruck, Austria
Barbara.Weber@uibk.ac.at

Abstract. Declarative process models are increasingly used since they fit better with the nature of flexible process-aware information systems and the requirements of the stakeholders involved. When managing business processes, in addition, support for representing time and reasoning about it becomes crucial. Given a declarative process model, users may choose among different ways to execute it, i.e., there exist numerous possible enactment plans, each one presenting specific values for the given objective functions (e.g., overall completion time). This paper suggests a method for generating optimized enactment plans (e.g., plans minimizing overall completion time) from declarative process models with explicit temporal constraints. The latter covers a number of well-known workflow time patterns. The generated plans can be used for different purposes like providing personal schedules to users, facilitating early detection of critical situations, or predicting execution times for process activities. The proposed approach is applied to a range of test models of varying complexity. Although the optimization of process execution is a highly constrained problem, results indicate that our approach produces a satisfactory number of suitable solutions, i.e., solutions optimal in many cases.

Keywords: declarative models, temporal constraints, constraint programming, planning, scheduling, clinical guidelines.

1 Introduction

Nowadays, there exists a growing interest in aligning information systems (IS) in a process-oriented way and in managing the supported processes effectively. Typically, processes are specified in an imperative way. However, declarative process models have been increasingly used allowing their users to specify *what* has to be done instead of *how* [24]. Given a declarative process model, users may choose among numerous ways to execute this model, i.e., there exist many different enactment plans for a given declarative model, each one presenting specific values for relevant objective functions (e.g., overall completion time or costs).

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 195–210, 2012.

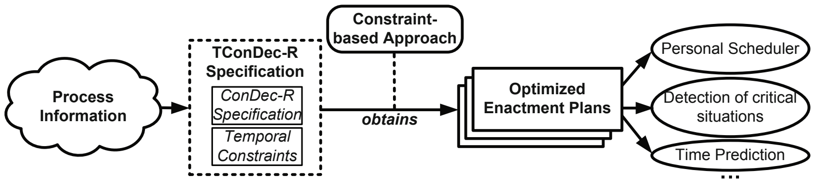


Fig. 1. Overview of our approach

Moreover, formal specification and operational support of temporal constraints constitute fundamental challenges for any process-aware information system. In [16], we presented a set of workflow time patterns for the systematic evaluation and comparison of workflow metamodels and tools supporting temporal aspects. These time patterns are based on empirical evidence we gained from several case studies.

For supporting users working on declarative workflows with explicit temporal constraints, this paper suggests a method for generating optimized enactment plans. That is, generating plans fixing the start and end times of the activities and resources used from declarative models, while considering resources and temporal constraints. In particular, generated plans aim at optimizing given objectives (e.g., minimizing overall completion time). We built upon the work presented in [3] where we proposed an extension of the declarative language ConDec [24], named ConDec-R. This includes capabilities for reasoning about resources and parallel execution of non-preemptive activities with known duration. Moreover, we proposed an approach for generating optimized enactment plans based on ConDec-R specifications. This paper significantly extends this work by additionally supporting selected time patterns [16], i.e., temporal ConDec-R (TCondec-R) specifications are considered. Hence, higher expressiveness can be achieved and more realistic problems managed.

Figure 1 provides an overview of our approach. Taking process information as a starting point, the TConDec-R specification is defined. From this specification, optimized enactment plans can be automatically generated. For this, activities to be executed are selected and ordered (*planning problem* [14]), considering the control-flow as well as the temporal constraints imposed by the constraint-based specification. Furthermore, as stated, the generation of enactment plans related to a declarative model requires that both the temporal and the resource perspectives are considered (*scheduling problem* [7]). For planning and scheduling (P&S) the activities in a way optimizing the objective function, a constraint-based approach is used.

The generated plans can improve process support and be used for different purposes: (i) providing users with a personal schedule, allowing them to improve their performance regarding activity executions [11], (ii) facilitating early detection of critical situations through early notifications and escalations, and (iii) predicting execution times for future activities, which allows users to make informed decisions [31]. In summary, the main contributions of this paper are: (1) an extension of the approach presented in [3] (i.e., generating optimized enactment plans from ConDec-R specifications) by providing improved expressiveness through complex temporal constraints [16], and (2) the application of the proposed approach to a range of test models of varying complexity.

Section 2 introduces an application example that emphasizes the need for our approach. Section 3 gives backgrounds on related research areas. Section 4 details the TConDec-R language and Section 5 shows how optimized plans can be generated. Section 6 deals with the evaluation, while Section 7 presents a critical discussion. Section 8 summarizes related work and Section 9 concludes the paper.

2 Application Example

To motivate the need for our approach we consider computer support for clinical guidelines. Clinical processes require the cooperation of different organizational units and medical disciplines [18]. In this context, clinical guidelines have been suggested for different medical disciplines to assist physicians in deciding about appropriate medical treatment for their patients under specific clinical circumstances [12]. Overall goal is to improve the quality of patient care and to reduce costs. Capturing respective clinical knowledge and incorporating it in clinical guidelines can potentially increase the effectiveness of patient treatment processes [18,22]. In such an environment optimal process support becomes crucial. Traditional languages for modelling clinical computer-interpretable guidelines (CIGs) are of imperative nature [23,30], which usually results in complex process models for which all possible treatment scenarios need to be pre-specified. Moreover, imperative languages usually present limited capabilities to provide flexibility for modelling and executing clinical guidelines [22,21,26]. This constitutes a barrier for applying process management to healthcare since the state of patients usually cannot be predicted, and hence the exactly required treatment (or sequence of diagnostic and therapeutic procedures) is not known a-priori. To increase flexibility and to reduce complexity of clinical process models, declarative CIG models [22,21] have been increasingly used to better fit with the nature of process-aware clinical IS and the requirements of the involved stakeholders [20].

In addition, temporal constraints play a fundamental role in the context of clinical guidelines [29,9,2,8]. For example, for most therapeutic procedures, the execution of related activities has to obey temporal constraints concerning activity orders, activity durations, and the temporal time lags between activities. In turn, in other scenarios (e.g., drug administration), activities have to be repeated periodically. Moreover, there are implicit temporal constraints that can be derived from the control-flow of a process model (e.g., synchronization), or from the scheduling constraints of a CIG.

CIGs are usually modelled by hypothesizing their application in an environment providing all required resources; guidelines are developed at an abstract level without focusing on a specific execution context [18,20]. This way, executing a CIG model requires that temporal constraints and the resource perspective are considered, i.e., reasoning about resource needs and availability is required. Moreover, given a declarative CIG model, clinical staff may choose among numerous ways to execute such model. The selection of an appropriate enactment plan, however, can be quite challenging since performance goals of the process should be considered and resource capacities be taken into account.

As stated, the proposed approach considers declarative models with explicit temporal constraints and resource reasoning, and hence, it is suitable for managing CIGs. However, our approach is not restricted to clinical environments, but can also be applied to other domains where processes are rather flexible and where temporal constraints play an important role (e.g., automotive industry and flight planning [16]).

3 Background

To automatically generate optimized enactment plans from constraint-based specifications (cf. Section 3.1), the areas of constraint programming, planning, and scheduling (cf. Section 3.2) are combined in this work.

3.1 Constraint-Based Process Models

In our proposal we use the declarative language ConDec [25,24] as basis for the control-flow specification. We consider ConDec to be a suitable language, since it allows specifying process activities together with the constraints to be satisfied for correct process enactment and for achieving the specified goal. Moreover, ConDec allows specifying a wide set of process models in a simple and flexible way. ConDec-R extends ConDec with estimates and resources [3].

Definition 1. *A* ***constraint-based process model*** $S = (Acts, C_{BP}, R)$ *consists of a set of activities* $Acts$*, a set of constraints* C_{BP}*, and a set* R *of available resources. For each activity* $a \in Acts$*, resource constraints can be specified by associating the role of the required resource with that activity.*

The activities of a constraint-based process model can be executed arbitrarily often if not restricted by any constraint. ConDec templates [24] constitute parameterized graphical representations of high-level constraints between activities which can be divided into the following categories:

1. **Existence Constraints:** unary relationships concerning the number of times an activity is executed. As example, *Exactly(N,A)* specifies that *A* must be executed exactly *N* times.
2. **Relation Constraints:** positive binary relationships used to establish what should be executed. As example, *Precedence(A,B)* specifies that *B* may only be executed if *A* is executed beforehand.
3. **Negation Constraints:** negative binary relationships used to forbid the execution of activities in specific situations. As example, *NotCoexistence(A,B)* specifies that if *B* is executed *A* cannot be executed, and vice versa.

Usually, several ways to execute constraint-based process models exist, i.e., there are different ways to execute a constraint-based process model while fulfilling all constraints. The different valid execution alternatives, however, can vary greatly in respect to their quality, i.e., in how well different performance objectives can be achieved. Thus, we propose to automatically generate optimized execution plans for a constraint-based model. We accomplish this by applying constraint programming for P&S the process activities (cf. Section 5).

3.2 Scheduling, Planning and Constraint Programming

The area of scheduling [7] includes problems for which it becomes necessary to determine an enactment plan for a set of activities related by temporal constraints. Moreover, the execution of activities requires resources, hence these activities may compete for limited resources. In general, the goal in scheduling is to find a feasible plan satisfying both temporal and resource constraints. Usually, several objective functions are considered for optimization, e.g., minimization of completion time. In a wider perspective, in AI planning [14], the activities to be executed are not established a priori, hence it becomes necessary to select them from a set of alternatives and to establish an ordering.

Constraint programming (CP) [27] has been successfully used for P&S purpose [28]. To solve a problem through CP, it needs to be modelled as a *constraint satisfaction problem* (CSP).

Definition 2. *A* ***CSP*** $P = (V, D, C_{CSP})$ *is composed out of a set of variables* V*, a set of domains of values* D *for all variables, and a set of constraints* C_{CSP} *between variables, such that each constraint represents a relation between a subset of variables and specifies the allowed combinations of values for these variables.*

A solution to a CSP consists of assigning values to CSP variables, such that the assignments satisfy all the constraints. Further, in CP, global constraints, i.e., constraints capturing a relation between a non-fixed number of variables, can be defined to improve the modelling of the problems.

Similar to CSPs, constraint optimization problems (COPs, cf. Def. 3) require solutions that optimize certain objective functions.

Definition 3. *A* ***COP*** $P_o = (V, D, C_{CSP}, o)$ *is a CSP including an objective function* o *to be optimized.*

Several mechanisms are available for solving CSPs and COPs, e.g., complete search algorithms, i.e., performing a complete exploration of a search space which is based on all possible combinations of assignments of values to the CSP variables. Regardless of the used search method, the global constraints can be implemented through filtering rules (i.e., rules responsible for removing values which do not belong to any solution) to efficiently handle the constraints in the search for solutions.

4 TConDec-R: Temporal Constraint-Based Process Language

To schedule process activities when generating optimized enactment plans, ConDec-R is used (cf. Section 3.1). As motivated, we extend ConDec-R to TConDec-R (cf. Def. 4) by including templates related to selected time patterns[16]:[1] pattern *TP1* (*Time Lags between Two Activities*) enables the definition of different kinds of time lags between two activities; pattern *TP2* (*Durations*) allows specifying the duration of process elements; pattern *TP4* (*Fixed Date Element*) provides support for specifying a deadline;

[1] Since events are not specified in the considered constraint-based language, in this approach, unlike in [16], only time patterns over activities are considered.

pattern *TP5* (*Schedule Restricted Element*) allows restricting the execution of a particular element by a schedule; pattern *TP6* (*Time Based Restrictions*) allows restricting the number of times a particular process element can be executed within a predefined time frame; pattern *TP7* (*Validity Period*) allows restricting the lifetime of a process element to a given validity period; pattern *TP8* (*Time Dependent Variability*) allows varying control-flow depending on the execution time or time lags between activities/events; pattern *TP9* (*Cyclic Elements*) allows specifying cyclic elements which are performed iteratively considering time lags between cycles; and pattern *TP10* (*Periodicity*) allows specifying periodically recurring process elements according to an explicit periodicity rule (for a description of the complete set of time patterns, see [16]). Moreover, for every TConDec-R temporal template all the relations which are stated in Allen's interval algebra [1] (i.e., start-start, start-end, end-start, and end-end) can be specified.

Definition 4. *A* ***TConDec-R process model*** $TCR = (Acts, C_T, R)$ *is a constraint-based process model* $S = (Acts, C_{BP}, R)$*,* $C_{BP} \subseteq C_T$*, in which* C_T *includes temporal constraints.*

As example, Fig. 2(a) shows a simple TConDec-R model representing the therapy of a patient: (1) $Acts$ is composed out of two activities: A, which has an estimated duration of 2h and requires a resource with role $R0$, and B, which has an estimated duration of 4h and requires a resource with role $R1$; (2) C_T is composed out of the following constraints: a) $Exactly(3, A)$, meaning that A must be executed exactly three times, b) $Exactly(2, B)$, expressing that B must be executed exactly twice, c) $DailyScheduleStart(A, [8am, 10am])$, meaning that each execution of A must be started between 8 am and 10 am (specific case for TP5), d) $CyclicStart-Start(B, [12h, 48h])$, meaning that between the start of two executions of B there must be at least 12h and at most 48h (specific case for TP9), and e) $PrecedenceEnd-Start(A, B, [2h, 4h])$, meaning that there must be a time lag of at least 2h and at most 4h between the end of any execution of A and the start time of the first execution of B (specific case for TP1); and (3) R is composed out of $\{[R0, 1], [R1, 1]\}$, which means that there is 1 resource with role $R0$, and 1 resource with role $R1$. In this example, all activities may be only executed between 8am and 4pm (specific case for TP5).

5 From TConDec-R to Optimized Enactment Plans

Activities and constraints are specified in a TConDec-R model. Thereby, several ways to execute this model might exist. Each of these execution alternatives leads to specific values of the objective function, i.e., the overall completion time, to be optimized. To generate optimized execution plans for a specific TConDec-R model, a constraint-based approach for P&S the process activities is proposed. This constraint-based approach includes the modelling of the declarative workflow as COP (cf. Def. 3, Section 5.1), the use of global constraints implemented through filtering rules (cf. Section 5.2), and search algorithms for solving the COP (cf. Section 5.3).

5.1 COP Model for TConDec-R Specifications

As first step, the TCondec-R model needs to be represented as CSP. Regarding the CSP model, recurring process activities (repeated activities, cf. Def. 5), which may be executed arbitrarily often if not restricted by any constraint, are modelled as sequence of

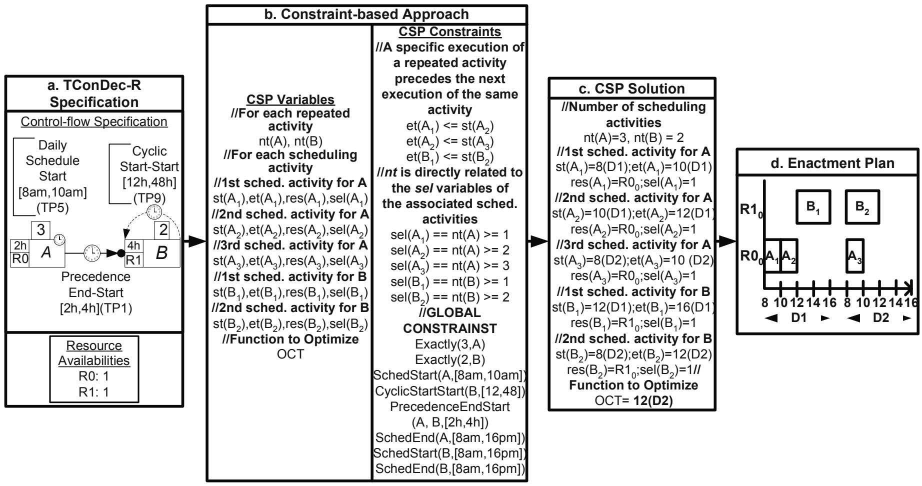


Fig. 2. From TConDec-R specification to process enactment plan

optional scheduling activities (cf. Def. 6). This is required since each execution of a process activity is considered as a single activity to be allocated to a specific resource and be temporarily placed in the enactment plan, i.e., stating values for its start and end times.

Definition 5. *A* ***repeated activity*** $ra = (dur, role, nt)$ *is a process activity which may be executed several times, i.e., several instances of the same activity may exist in the context of a particular process instance. A repeated activity is described by the estimated duration of the process activity (i.e., dur), the role of the required resource for activity execution (i.e., role), and a CSP variable specifying the number of times the process activity is executed (i.e., nt).*

For each repeated activity, *nt* scheduling activities exist, which are added to the CSP problem specification, apart from including a variable *nt*.

Definition 6. *A* ***scheduling activity*** $a_i = (st, et, res, sel)$ *represents the i-th execution of a repeated activity a, i.e., a specific process activity instance, where st and et are CSP variables indicating the start/end times of activity execution (each execution of a process activity needs to be temporarily placed in the enactment plan), res is a CSP variable representing the resource used for execution, and sel is a CSP variable indicating whether the activity is selected for execution.*

Moreover, an additional CSP variable representing the overall completion time (OCT), is included in the CSP model, extending the CSP to a COP (cf. Def. 7).

Definition 7. *A* ***COP-TConDec-R problem*** *related to a TConDec-R process model* $TCR = (Acts, C_T, R)$ *(cf. Def. 4) is a COP* $P_o = (V, D, C_{CSP}, o)$ *(cf. Def. 3) where:*

– *The set of variables V is composed out of all CSP variables included in the CSP model plus the CSP variable related to overall completion time (OCT), i.e.,* $V = \{nt(a), a \in Acts\} \cup \{st(a_i), et(a_i), res(a_i), sel(a_i), i \in [1..nt(a)], a \in Acts\} \cup OCT$.

```
CyclicStartStart(a,[li,ls]) is added OR bounds of st(ai) for any i changed ->
  for (int i = 1; i < UB(nt(a)); i++){
    SchedulingActivity a1 = ai;
    SchedulingActivity a2 = ai+1;
    if (LB(st(a1))+(li)>LB(st(a2))) {LB(st(a2))<-LB(st(a1))+(li)} //LB(st(ai+1))>=LB(st(ai))+li
    if (UB(st(a1))>UB(st(a2))-(li)) {UB(st(a1))<-UB(st(a2))-(li)} //UB(st(ai))<=UB(st(ai+1))-li
    if (LB(st(a2))-(ls)>LB(st(a1))) {LB(st(a1))<-LB(st(a2))-(ls)} //LB(st(ai))>=LB(st(ai+1))-ls
    if (UB(st(a2))>UB(st(a1))+(ls)) {UB(st(a2))<-UB(st(a1))+(ls)} //UB(ai+1)<=UB(st(ai+1))+ls
  }
```

Fig. 3. Filtering Rule for the CyclicStartStart Template

- *The set of constraints C_{CSP} is composed out of the global constraints (implemented by the filtering rules) related to the TConDec-R constraints included in C_T together with the constraints from the proposed CSP model*[2]*, i.e.:*
  - *A specific execution of a repeated activity precedes the next execution of the same activity, i.e., $\forall i: 1 \leq i < nt(a): et(a_i) \leq st(a_{i+1})$ for each repeated activity $a \in Acts$.*
  - *The nt variable is directly related to the sel variables of the associated scheduling activities, i.e., $\forall i: 1 \leq i \leq nt(a): sel(a_i) = 1 \wedge \forall i > nt(a): sel(a_i) = 0$ for each repeated activity $a \in Acts$.*
  - $OCT = max_{a \in Acts}(et(a_{nt(a)}))$.
- *The set of domains D is composed out of the domains for each variable from V.*
- *The objective function to be optimized is overall completion time, i.e., $o = OCT$.*

In this way, the COP model which was proposed for ConDec-R specifications [3] has been extended by including: (1) a new global constraint for each of Allen's interval algebra relation of each specific case of every supported temporal constraint, i.e., time patterns TP2, TP4, TP5, TP6, TP7, TP8, TP9, and TP10, and (2) a new global constraint for each of Allen's interval algebra relation of every relation and negation ConDec constraint for allowing the specification of time lags (i.e., time pattern TP1). Moreover, when all process activities may be executed in a specific time frame $[li, ls]$, the constraints $DailyScheduleStart(a, [li, ls])$ and $DailyScheduleEnd(a, [li, ls])$ are included for every activity $a \in Acts$ which is not involved in any other schedule constraint (cf. Fig. 2(b)). This is needed since the *st* and *et* variables can take any value, e.g., a value corresponding to 4 am, if not restricted by any constraint.

Figure 2 also shows the translation from a TConDec-R specification into a CSP so that the CSP variables and constraints are stated as explained in Def. 7 (cf. Fig. 2(b)).

5.2 Filtering Rules

For each TConDec-R template our constraint-based proposal includes a related global constraint implemented through a filtering rule. Since we extend ConDec-R[3] [3] by time patterns, new filtering rules related to these time patterns have been developed, i.e., one

[2] Resources are implicitly constrained since the solver which is used provides a high-level constraint modelling specific to scheduling which includes the management of shared resources.

[3] A detailed description of the ConDec-R filtering rules can be found at http://regula.lsi.us.es/MOPlanner/FilteringRules.pdf

```
DailyScheduleEnd(ai,[li,ls]) is added OR bounds of et(ai) changed ->
  // a)
  if((LB(et(ai))%(24*60)) < (li)){
    int day = LB(et(ai))/24*60;
    int newValue = day*(24*60) + li;
    LB(et(ai)) <- newValue;
  }
  // b)
  if((LB(et(ai))%(24*60)) > (ls)){
    int day = LB(et(ai))/24*60;
    int newValue = (day+1)*(24*60) + li;
    LB(et(ai)) <- newValue;
  }
```

```
  // c)
  if((UB(et(ai))%(24*60)) > (ls)){
    int day = UB(et(ai))/24*60;
    int newValue = day*(24*60) + ls;
    UB(et(ai)) <- newValue;
  }
  // d)
  if((UB(et(ai))%(24*60)) < (li)){
    int day = UB(et(ai))/24*60;
    int newValue = (day-1)*(24*60) + ls;
    UB(et(ai)) <- newValue;
  }
```

Fig. 4. Filtering Rule for the DailyScheduleEnd Template

filtering rule for each new global constraint (cf. Section 5.1). As examples, Fig. 3 and 4 show the filtering rules related to the $CyclicStartStart(a,[li,ls])$ and $DailySchedule-End(a,[li,ls])$[4] global constraints, where $UB(var)$ and $LB(var)$ represent the upper and lower bounds of the domain of var, respectively. Most of the newly developed filtering rules present a propagation reasoning similar to the one included in the ConDec-R filtering rules, i.e., they basically differ in the consideration of the time lags (see Fig. 3 for an example). However, for the filtering rules related to the schedule templates, it becomes necessary to reason about the day in which the upper and lower bounds of the start and/or end time variables are placed. Specifically, for the filtering rule of Fig. 4, for every activity execution a_i the next reasoning is carried out:[5] a) if the lower bound of $et(a_i)$ corresponds to a time of a day d which is lower than the time li, then that lower bound is updated to the time li of the day d; b) if the lower bound of $et(a_i)$ corresponds to a time of a day d which is greater than the time ls, then that lower bound is updated to the time li of the day after d; c) if the upper bound of $et(a_i)$ corresponds to a time of a day d which is greater than the time ls, then that upper bound is updated to the time ls of the day d; and d) if the upper bound of $et(a_i)$ corresponds to a time of a day d which is lower than the time li, then that upper bound is updated to the time ls of the day before d.

In this way, the constraints stated in the TConDec-R specification (cf. Def. 4) can be easily included in the CSP model through the related global constraints. Moreover, the related filtering rules increase the efficiency in the search for solutions, since during the search process these filtering rules remove inconsistent values from the domains of the variables. In the CSP model, initial estimates are made for upper and lower bounds of variable domains, and these values are refined during the search process.

5.3 Search Algorithms

Once the problem is modelled, several constraint-based mechanisms can be used to obtain the solutions of the COP (cf. Def. 3), i.e., optimized enactment plans (cf. Def. 8).

[4] Note that since the $DailyScheduleEnd(a,[li,ls])$ constraint individually affects each activity execution, the filtering mechanism for every scheduling activity is carried out in a separated way to increase the efficiency. In this way, the $DailyScheduleEnd(a,[li,ls])$ constraint is implemented through the set $\{DailyScheduleEnd(a_i,[li,ls]), i \in [1..nt(a)]\}$ of filtering rules.

[5] To deal with different time granularities, all the temporal specifications of the TConDec-R model are automatically converted to minutes when generating the CSP.

For the empirical evaluation of this paper, we use the heuristic complete search method *setTimes* [17] since it has demonstrated its ability to obtain good solutions to complex scheduling problems.

Definition 8. *An* ***enactment plan*** *consists of: (1) the number of times each activity is executed, (2) the start and end times for each activity execution, and (3) the resource which is used for each activity execution.*

Figure 2(d) shows an enactment plan which represents the CSP solution of Fig. 2(c) related to the TConDec-R specification of Fig. 2(a).

Since the generation of optimal plans has NP-complexity [13], it is not possible to ensure the optimality of the generated plans for all cases. The developed constraint-based approach, however, allows solving the considered problems in an efficient way, reaching solutions which are optimal in many cases (cf. Section 6).

6 Empirical Evaluation

To evaluate the effectiveness of our approach, a controlled experiment has been conducted. Section 6.1 describes the design underlying the experiment, and Section 6.2 shows the experimental results and the data analysis.

6.1 Experimental Design

Purpose: The purpose of the empirical evaluation is to analyze the behavior of our proposal in the generation of optimal enactment plans from TConDec-R (i.e., temporal ConDec-R) specifications.

Objects: Considering the application scenario from Section 2, the empirical evaluation is based on the generic TConDec-R model (cf. Fig. 5), which represents a specific treatment to be applied to #P patients. This scenario has been selected, since it includes typical relations present in actual CIGs. It further contains a representative set of time patterns. In this context, we presume that all activities may be executed between 8am to 8pm.

The generic TConDec-R model of Fig. 5 is specified by replacing the variables $\{\#P, \#R0, \#R1, D_{a\in Acts}\}$ with specific values, being D_a the estimated duration for a. Regarding the number of patients values $\#P \in \{5, 10, 15\}$, and for the number of resources with roles $R0$ and $R1$ values $\{5, 10\}$ are considered. In addition, different games of durations for each process activity are assumed (G), since this aspect has great influence on the complexity of the search for optimums. Note that the considered problems are an extension of typical scheduling problems. 30 instances are randomly generated for each TConDec-R model by varying activity durations between 5 and 30 minutes.[6]

Independent Variables: For the empirical evaluation, (1) the number of patients (i.e., #P), (2) the number of available resources with role $R0$ or $R1$, respectively (i.e., #$R0$,

[6] The set of games which are used for the empirical evaluation are available at http://www.lsi.us.es/~quivir/irene/Games.rar

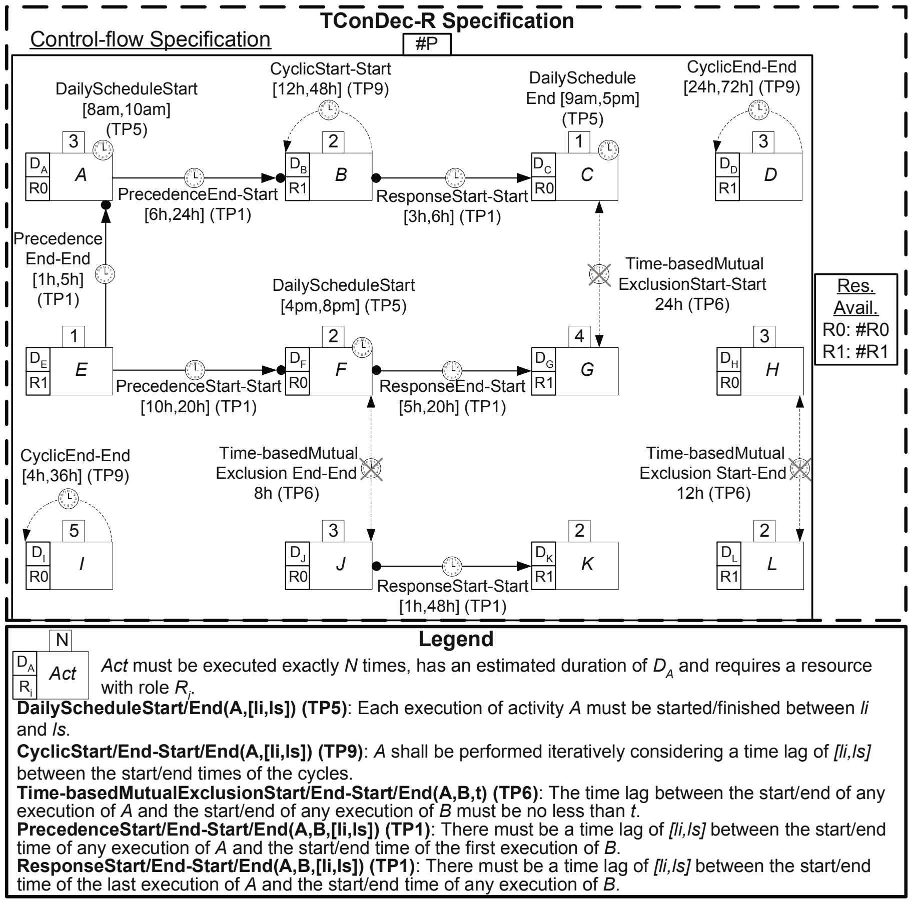


Fig. 5. A generic TConDec-R model

#$R1$), and (3) the game which establishes the activity durations (i.e., G) are taken as independent variables.

Response Variables: The suitability of our approach is tested regarding the following variables: (1) percentage of optimal solutions found (i.e., $\%Opt$)[7], (2) average time (in seconds) for getting optimal solutions, considering the cases in which the optimal solution is found (i.e., $T_{Opt}(s)$), and (3) average value of the objective function obtained (i.e, overall completion time $OCT(min)$).

Experimental Design: For the model depicted in Fig. 5, 360 instances (i.e., $3 \times 2 \times 2 \times 30$) are generated considering different values of $\#P$ (3 values), $\#R0$ (2 values), $\#R1$ (2 values), and G (30 problem instances). The response variables are then calculated by considering the average values for the 30 problem instances.

[7] The optimality of the solutions can be only ensured if the search algorithm stops before reaching the time limit. Otherwise the optimality of the reached solution is unknown.

Experimental Execution: The machine we use is an Intel Core2, 2.13 GHz, 1.97 GB memory, running on Windows XP. For the experiments, the complete search method *setTimes* [17] is run until a 5-minutes CPU time limit is reached. To implement the constraint-based problems (cf. Section 5), COMET [10] is used, which is able to generate high-quality solutions for highly constrained problems in an efficient way. This system provides a scheduling module offering high-level constraint modelling and search abstraction, both specific to scheduling.

6.2 Experimental Results and Data Analysis

For each problem (i.e., $\{\#P, \#R0, \#R1\}$) Table 1 shows: (1) the total number of scheduling activities (cf. Def. 6) to be planned and scheduled ($\#SchedAct$), and (2) the values of the response variables (i.e., $\%Opt$, T_{Opt}, and OCT) for the 30 problem instances randomly generated.[8] As expected, the percentage of optimal solutions found decreases and the average time for getting optimal solutions increases as the number of patients (and hence the number of scheduling activities) increases. Specifically, for 5 patients (155 scheduling activities) the optimum is found in almost all cases (the average value for $\%Opt$ is equal to 99.16%), for 10 patients (310 scheduling activities) the average value for $\%Opt$ is equal to 36.66%, and for 15 patients (465 scheduling activities) the average value for $\%Opt$ is equal to 6.66%. Moreover, in almost all cases, the value for $\%Opt$ increases and the value for $T_{Opt}(s)$ decreases as the number of available resources increases. As expected, the average value for OCT increases as the number of patients (and hence the number of scheduling activities) increases and the number of available resources decreases. Additionally, it can be seen that the number of available resources with role $R1$ seems to be more influential than the number of available resources with role $R0$ in all response variables.

In general, experimental results show that despite NP-complexity of the problems considered, the values for the percentage of optimal solutions found and for the average time for getting optimums are quite good for medium-sized problems (between 155 and 465 scheduling activities). Note that getting the optimum for scheduling problems of 155-465 activities can entail a great complexity. In fact, there are many scheduling benchmarks of smaller size for which their optimal values are not even known.

7 Discussion and Limitations

The current approach allows modelling processes in an easy way, since the considered declarative specifications are based on high-level constraints. Furthermore, time patterns can be easily specified since the proposed constraint-based language includes temporal constraints. This is a big advantage. Although temporal constraints play an important role in the context of long-running processes, time support is very limited in existing process management systems [16]. With our extension, an increased expressiveness to the specification language is provided (compared to [3]), and hence more realistic problems can be managed, e.g., CIG support in the clinical domain

[8] The set of optimized enactment plans which were generated during the empirical evaluation are available at `http://www.lsi.us.es/~quivir/irene/OptimizedEnactmentPlans.rar`

Table 1. Experimental results (5-minutes time limit)

$\#P$	$\#R0$	$\#R1$	$\#SchedAct$	$\%Opt$	$T_{Opt}(s)$	$OCT\,(min)$
5	5	5	155	96.66	0.21	3666
5	5	10	155	100	3.03	3618
5	10	5	155	100	0.94	3618
5	10	10	155	100	0.98	3618
10	5	5	310	3.33	0.86	4602.58
10	5	10	310	46.66	31.45	3833.66
10	10	5	310	10	0.83	4511.85
10	10	10	310	86.66	3.36	3715.33
15	5	5	465	0	-	6437.20
15	5	10	465	16.66	9.27	4590.43
15	10	5	465	3.33	1.45	6388.27
15	10	10	465	6.66	1.50	4317.93

(cf. Section 2). Moreover, one advantage of our proposal is that optimized enactment plans are generated by considering all process activities; hence, it allows for a global optimization of the objective functions. Finally, the automatic generation of optimized plans can deal with complex problems in a simple way, as demonstrated in Section 6. Hence, a wide study of several aspects can be carried out by simulation.

Nonetheless, the proposed approach also presents a few limitations. First, the analysts must deal with a new language for the constraint-based specification, thus a period of training is required to let them become familiar with TConDec-R specifications. Secondly, the optimized process models are generated by considering estimated values for the number of process instances, activity durations, and resource availability, and hence the current proposal is only appropriate for processes in which these values can be estimated. However, P&S techniques can be applied to replan the activities in the enactment phase by considering the actual values of the parameters, as stated in [4].

8 Related Work

This paper extends the approach presented in [3] by providing improved expressiveness through temporal constraints [16]. We are not aware of any other approach for generating enactment plans from declarative specifications, however, there exist some further proposals which could be extended in such direction [21,20]. Similar to our work, [21] presents a declarative language based on ConDec (i.e., CIGDec) for the modelling and enactment of CIGs. From CIGDec specifications an automaton representing all feasible traces can be generated. The overall completion time of all the traces could be calculated [31], and hence optimized enactment plans be generated. However, as a disadvantage of this approach, generating the automaton is NP complete, and, unlike the proposed approach, no heuristics is used. Additionally, CLIMB [20] could be used to generate quality traces from declarative specifications, and calculate its completion time. Then, the best traces could be selected. Unlike our approach, [20] neither considers optimality nor resource availability. Finally, the time patterns presented in [16] are not considered in [21,20].

Many constraint-based approaches for modelling and solving P&S problems have been proposed [27]. Moreover, several proposals exist for filtering algorithms related to specialized scheduling constraints [5]. Therefore, the considered problem could be managed by adapting existing constraint-based approaches. However, these problems include many non-typical scheduling constraints from ConDec, which entail complex reasoning about several combined innovative aspects, such as the alternating executions of repeated activities together with the varying number of times which these activities are executed. Therefore, we implemented our own specific filtering rules to increase the efficiency in the search for solutions.

Furthermore, constraint-based approaches for process design verification have been proposed in process-aware IS [19]. Unlike our approach, they do not consider the generation of optimized process enactment plans.

Related to the clinical domain, the CIG languages presented in [21,20,6,15] do not consider time patterns. However, there are approaches focussing on the treatment of temporal aspects in CIGs (e.g., [29,9,2,8]). Opposed to our work, the works presented in [29,9,2,8] do not consider optimality issues when managing temporal constraints.

9 Conclusions and Future Work

This paper presents a method for generating optimized enactment plans (e.g., minimizing overall completion time) from declarative temporal process models. The generated plans can be used for different purposes, e.g., providing users with a personal schedule, facilitating early detection of critical situations, or predicting execution times for process activities. The proposed approach is applied to a range of test models of varying complexity. Results indicate that, despite the NP-complexity of the considered problems, our approach produces solutions being optimal in many cases. As for future work, we will explore various constraint-based solving techniques and analyze their suitability for the generation of optimized enactment plans.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. In: Proc. Communications of the ACM, pp. 832–843 (1983)
2. Anselma, L., Terenziani, P., Montani, S., Bottrighi, A.: Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. Artificial Intelligence In Medicine 38, 171–195 (2006)
3. Barba, I., Del Valle, C.: A Constraint-based Approach for Planning and Scheduling Repeated Activities. In: Proc. COPLAS, pp. 55–62 (2011)
4. Barba, I., Weber, B., Del Valle, C.: Supporting the Optimized Execution of Business Processes through Recommendations. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 135–140. Springer, Heidelberg (2012)
5. Barták, R., Cepek, O.: Incremental propagation rules for a precedence graph with optional activities and time windows. Transactions of the Institute of Measurement and Control 32(1), 73–96 (2010)

6. Bottrighi, A., Chesani, F., Mello, P., Molino, G., Montali, M., Montani, S., Storari, S., Terenziani, P., Torchio, M.: A Hybrid Approach to Clinical Guideline and to Basic Medical Knowledge Conformance. In: Combi, C., Shahar, Y., Abu-Hanna, A. (eds.) AIME 2009. LNCS, vol. 5651, pp. 91–95. Springer, Heidelberg (2009)
7. Brucker, P., Knust, S.: Complex Scheduling (GOR-Publications). Springer (2006)
8. Combi, C., Gozzi, M., Juarez, J.M., Oliboni, B.: Conceptual Modeling of Temporal Clinical Workflows. In: Proc. TIME, pp. 70–81 (2007)
9. Duftschmid, G., Miksch, S., Gall, W.: Verification of temporal scheduling constraints in clinical practice guidelines. Artificial Intelligence In Medicine 25(2), 93–121 (2002)
10. Dynadec. Comet Downloads (2010), `http://dynadec.com/support/downloads/` (accessed October 3, 2011)
11. Eder, J., Pichler, H., Gruber, W., Ninaus, M.: Personal Schedules for Workflow Systems. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 216–231. Springer, Heidelberg (2003)
12. Field, M.J., Lohr, K.N.: Guidelines for clinical practice: from development to use. National Academy Press, Institute of Medicine, Washington, D.C (1992)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
14. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann, Amsterdam (2004)
15. Grando, M.A., van der Aalst, W.M.P., Mans, R.S.: Reusing a Declarative Specification to Check the Conformance of Different CIGs. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBIP, vol. 100, pp. 188–199. Springer, Heidelberg (2012)
16. Lanz, A., Weber, B., Reichert, M.: Workflow Time Patterns for Process-Aware Information Systems. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 94–107. Springer, Heidelberg (2010)
17. Le Pape, C., Couronne, P., Vergamini, D., Gosselin, V.: Time-versus-capacity compromises in project scheduling. In: Proc. PlanSIG, pp. 498–502 (1994)
18. Lenz, R., Reichert, M.: IT support for healthcare processes - premises, challenges, perspectives. Data & Knowledge Engineering 61(1), 39–58 (2007)
19. Lu, R., Sadiq, S., Governatori, G., Yang, X.: Defining Adaptation Constraints for Business Process Variants. In: Abramowicz, W. (ed.) BIS 2009. LNBIP, vol. 21, pp. 145–156. Springer, Heidelberg (2009)
20. Montali, M.: Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach. PhD thesis, Department of Electronics, Computer Science and Telecommunications Engineering. University of Bologna (2009)
21. Mulyar, N., Pesic, M., van der Aalst, W.M.P., Peleg, M.: Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 335–346. Springer, Heidelberg (2008)
22. Mulyar, N., van der Aalst, W.M.P., Peleg, M.: A pattern-based analysis of clinical computer-interpretable guideline modelling languages. Journal of the American Medical Informatics Association 14, 781–787 (2007)
23. Peleg, M., et al.: Comparing computer-interpretable guideline models: A case-study approach. Journal of the American Medical Informatics Association 10(1), 52–68 (2003)
24. Pesic, M.: Constraint-Based Workflow Management Systems: Shifting Control to Users. PhD thesis, Technische Universiteit Eindhoven, Eindhoven (2008)
25. Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P.: Constraint-Based Workflow Models: Change Made Easy. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 77–94. Springer, Heidelberg (2007)

26. Reichert, M.: What BPM Technology Can Do for Healthcare Process Support. In: Peleg, M., Lavrač, N., Combi, C. (eds.) AIME 2011. LNCS, vol. 6747, pp. 2–13. Springer, Heidelberg (2011)
27. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier (2006)
28. Salido, M.A.: Introduction to planning, scheduling and constraint satisfaction. Journal of Intelligent Manufacturing 21(1), 1–4 (2010)
29. Shahar, Y.: A framework for knowledge-based temporal abstraction. Artificial Intelligence 90(1/2), 79–133 (1997)
30. ten Teije, A., Miksch, S., Lucas, P.: Computer-based Medical Guidelines and Protocols: A Primer and Current Trends. IOS Press (2008)
31. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Information Systems 36(2), 450–475 (2011)

MobiQ: Mobile Based Processes for Efficient Customer Flow Management

Mersini Paschou, Evangelos Sakkopoulos, Efrosini Sourla, and Athanasios Tsakalidis

Computer Engineering and Informatics Department,
University of Patras, 26500, Patras, Greece
{paschou,sakkopul,sourla,tsak}@ceid.upatras.gr

Abstract. Queues are still a part of the everyday life, though Internet and mobile technologies are already available. Queues are formed whenever a bottleneck of customer demand for a service appears. There are workarounds for customer flow management or queue management in order to provide a level of freedom to the customers such as queue ticket numbers. However, there can be many more things that can be done to free additional time for the customer until his/her turn comes to be services. In this work, we propose an solution framework for servicing customer flow with *live* feedback. Our goal is to improve customers' experience and Quality of Services (QoS), by integrating mobile access with Short Message Service (SMS) to improve customer flow in popular crowded services, where long queues are created. A fully functioning real life prototype is presented, demonstrating the feasibility and effectiveness in terms of time and effort.

Keywords: mobile services, customer flow management, web services, m-tickets, m-government, mobile queue management.

1 Introduction

The era of globalization or digital world has created a new kind of information society and knowledge-based economy, which changed our human behavior. Information and Communication Technologies (ICT) are considered to be the powerful tool which constitutes an important factor for those changes. The term “e-Services” and respectively “m-Services” for mobile phones and PDAs, refers to the use of ICT to provide and improve everyday services, transactions and interactions with citizens, businesses, and other private or public bodies.

Electronic tickets (e-tickets) are an example of such a class of e-services. Generally speaking, e-tickets are the Internet counterpart of real-world tickets, thus they give evidence that their holder has paid or is entitled to some service [1]. Unfortunately, no matter how widespread the use of ICT is, there is still a great number of services that require physical presence of the customer and cannot be completed using only individual e-Services. Customer Flow Management systems for private and public

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 211–225, 2012.

organizations or enterprises fail at facing the problem of the resulting long queues and waiting times. Such cases include banks, pharmacies, post offices, hospitals, shopping malls, police stations, employment services and in a variety of contexts from travel, to entertainment or healthcare. [2]

In these cases, customers first enter the reception area and then get a ticket that puts them in a waiting queue until their time to be served comes. Finally, they are informed to enter the service area. These systems have a number of deficiencies which adversely affect users' experience and are mentioned below:

1. First of all, customers have to enter the reception area first, in order to get a ticket. No electronic tickets production system exists for customers to receive e-tickets before they arrive at the building, either while being at home or when being on the road, moving in the city and doing chores at different services.
2. Furthermore, customers have to wait in the facilities of the organization or enterprise until they are served, due to the fact that no system exists to inform them whenever they want about their refreshed waiting time. In other words, customers/citizens take a risk to lose their turn, if they leave the queue in order to perform some other task as they have no "live" feedback which would allow them to perform any sort of time management.

The use of mobile phones allows people to perform time management and face the two problems stated above. This works especially well when it comes to small groups of people coordination and agenda aligning. For example, friends call each other to setup a meeting while being mobile on the road and agree to update themselves in case of small schedule or place changes until they meet. The mobile phone then becomes a leading artifact for temporality in a flexible society. More and more, the mobile phone replaces the calendar and the watch as they incorporate and enhance them both [3]. Of course, when it comes to service organizations and its customers, then the "live" and constant update through phone calls is practically impossible as it requires much effort and becomes expensive and complicated (in terms of both money and time). Studying customer queues particularly includes the ways of delivering the waiting time by steps. In the proposed solution we re-engineered ways of delivering waiting time through Internet and GSM [4].

In this work, we propose m-Services which are based on mobile data access and Short Message Service (SMS) messages, to provide an integrated solution framework for servicing customer flow management with live feedback. A fully functioning real life prototype is also presented and discussed to show the feasibility and effectiveness of the software approach. The prototype is adapted to the business model of Banking sector [5], [6], however this has no impact on the generality of the architectural approach. In short, the m-Services allow e-tickets creation before customers reaching the servicing area using three different options:

(i) via mobile web access specially adapted to mobile phones,
(ii) using stand alone mobile applications downloaded at the mobile devices and
(iii) using SMS messages sent from a mobile phone.

Moreover, the m-Services include functionality to provide “live” feedback to the e-ticket holders about their up-to-date, updated waiting time at the queue through all the above access options (i-iii). Additionally, in order to present holistic software architecture the proposed framework also provides its services (iv) via desktop internet based access. To the best of authors' knowledge, this framework constitutes one of the first attempts in managing customer flow through mobile devices with all different possible means.

Waiting time is an important issue and has strong effect on customer satisfaction [7], [10]. It is argued that queues engender a loss of personal control and an overestimation of time spent in waiting as well as boredom and physical discomfort for many visitors. Delivering different queue forms or physical incorporation of queues into operational space and attention to physical comfort has been used to improve satisfaction [8]. Having customers additionally informed proves to be helpful [9]. The key concept of the proposed solution is to re-engineer the queue waiting process and introduce new means to improve user experience such as software based time announcements through the Internet and mobile networks. It is important to have in mind that the proposed solution adds and extends existing queue management systems that use paper tickets (numbers that show who is next). We show that using mobiles and the Internet there is an improved and alternative way to provide the queue management service.

Overall, the discussed prototype is adapted to the business model of Banking sector, yet causing no impact on the generality of the architecture. Briefly, mobile services allow e-tickets derivation for customers who are not present at the servicing area, providing an interim solution for those services that require physical presence. Moreover, mobile services provide "live" feedback to e-ticket holders, informing them of their updated waiting time at the queue. The evaluation results are encouraging and the prototype has got much attention by the Greek e-Gov community.

The rest of the paper is organized as follows: Section 2 discusses related work about flow management services and m-Services. Section 3 presents the key aspects of the mobile framework proposed. Section 4 discusses theoretical background behind customer flow management and queue management. Section 5 discusses the functionality of the real life prototype for the different available access options (mobile web, mobile applications, SMS and desktop web). Finally, section 6 concludes the paper and provides future ideas for further investigation.

2 Related Work

Organizations and enterprises provide, in many cases, a wide variety of e-Services which include internet services and services for mobile devices. These services involve business and financial transactions as well as information services. However, the former e-Services rarely don't include services for customer flow management using e-tickets, for customers who choose to be served locally in the shops, company or organization branches. Practically nonexistent are services that allow e-ticket holders to get updated about the current-refreshed waiting time at the queue, thus providing "live" feedback.

Proponents of m-Services argue it can help make public information and services available "anytime, anywhere". An example of such beneficial use of mobile technologies would be sending a mass alert to registered citizens via short message service, or SMS, in the event of an emergency. To quote m-Services theorist and proponent Ibrahim Kuchshu, “As e-business evolves towards m-business, e-Government seems to follow the trend with a few but significant mobile government (i.e. m-Services in general) applications” [11].

2.1 E-Tickets for Purchasing

In order to present the key differences between queue tickets for customer management and traditional e-tickets for product and services purchasing a short description is given below. E-tickets are the Internet counterpart of real-world tickets and give evidence that their holder has paid or is entitled to some service (e.g., entering a place of entertainment, upgrading software from the Internet).

Users can acquire e-tickets by purchasing them from a web server, or simply receiving them from a vendor, as part of a promotion, or from another user who previously acquired them. E-tickets can be stored in a desktop computer or in a personal digital assistant (PDA) for future use [12]. They are very popular nowadays in the airline industry. An e-ticket is a paperless electronic document used for ticketing passengers, particularly in the commercial airline industry. E-Tickets have been introduced in road, urban or rail public transport as well.

2.2 Customer Flow Management

Before proceeding further, we shortly present existing business ready solutions for customer flow management follows, in order to show the current state of the art business and software approaches. Below, we present two of the most popular customer flow management systems: Q-Matic and Nemo-Queue.

Q-Matic Customer Flow Management Systems has created a Customer Flow Management (CFM) system for private and public organizations and enterprises, which face the problem of big queues and waiting times [15]. It's all about managing the flow of customers and their experiences from their initial contact with the company, through to service delivery as well as seeking their feedback and views after they've received the service they need.

Nemo-Queue Customer Flow Management Systems is a pioneer in customer flow management and queuing systems, with experience from thousands of installed systems throughout the world. CFM systems can be used in several different areas where the customer flow management can be improved such as: banks, pharmacies, post offices, hospitals, shopping malls, police stations and employment services [16].

The problems that the aforementioned systems (and similar ones) can’t solve, have been described analytically in the previous section and refer to the fact that customers (a) have to enter the reception area to acquire a ticket and (b) have no “live” feedback to perform any sort of time management.

2.3 SMS Based Solution for Information Services

System Pandora [14], which uses SMSs to provide services related to education, is presented bellow as an example of novel SMS based m-Services. This system, which was developed in the Aegean University, provides a network place in which the user can create an account, so that he/she acquires access in the provided services. In the next level, the user can subscribe to some service and regularly receive informative SMSs, with the only pre-requisite being an e-mail address of the Aegean university.

Abilities of activation of an account via the system network place and renewal of the balance with the use of cards from well-known mobile telephony companies of Greece are offered. The balance renewal is accomplished with the dispatch of a suitable SMS and each renewal suffices for a relatively big number of uses of the provided services. Descriptions and details for the provided services exist to the panel of services allocated in the network place as well as in the examples that present their syntax. The system offers its users the flexibility to be informed and act via mobile phones both for issues of academic interest and for various other issues of interest.

2.4 Time Management Using Mobile Devices

An attempt to provide time management for service provision and organization directly by the customers is the work presented in [3]. A mobile application downloaded and installed locally at the mobile phone is utilized to facilitate service meetings and transportation timelines. Unfortunately at the time of development, mobile data access and throughput was not enough to support the idea of the authors efficiently.

In this work, all possible different mobile access possibilities are given in order to schedule and get updates about customer queues and flow management. Specially designed mobile web access, internet accessing mobile applications locally stored and SMS based communication are employed, in order to provide flexibility in the communication and support for a wide range of different mobile phones and carriers. Next, we present the key concepts of the proposed software approach.

3 Customer Flow Management Extending to Mobile Devices

To ease the readership and to provide more clear flow of concepts, we discuss ideas and techniques for the prototype's case applied to the banking sector. This does not narrow at all the application spectrum and generalization of the solution and it is meant to assist flow of ideas.

The proposed solution in short allows e-tickets creation, via internet applications and applications for mobile phones, before customers reach the servicing area. Moreover, it informs regularly or ad-hoc upon request the e-ticket holders about their updated waiting time. MobiQ approach provides users with a ticket-code that determines its holder's position in a waiting queue at a specific date/time. It works in fact as an electronic version of «paper tickets» citizens/ customers use in order to obtain their positions in waiting queues.

In the case of the paper ticket, the waiting time written on it is calculated only once, when customer gets the ticket, so he is informed of the time he will have to wait before served by a teller. This waiting time is indicative and often not accurate, so customers are "trapped" in the bank waiting for their turn (see Fig. 1(a)), "anxious" and possibly "afraid" to lose their turn in the meantime in case they leave the bank to perform other tasks (see Fig. 2(a)). On the contrary, the e-ticket holder can be informed of the renewed waiting time, at any time (see Fig. 1(b) and Fig. 2(b) – The

MobiQ solution). The MobiQ solution allows the customer the flexibility to allocate his time as desired, until his service time arrives, and to be informed, whenever he wants, for the new waiting time, which may have changed significantly due to the nonlinear service execution time.

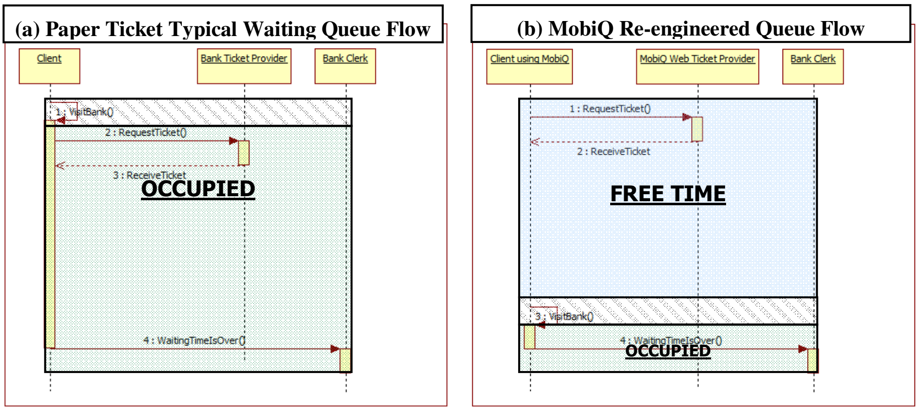


Fig. 1. (a) Paper Ticket Queue, (b) MobiQ Re-engineered Queue

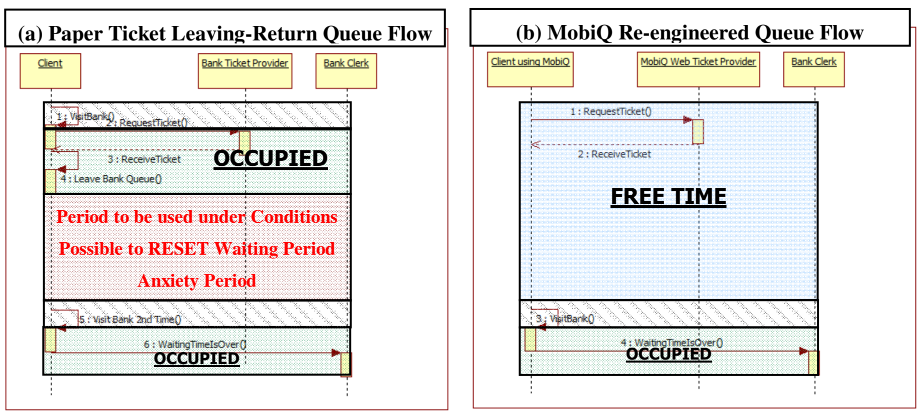


Fig. 2. (a) Paper Ticket Leaving-Returning 2nd Time to the Queue, (b) MobiQ Re-engineered Queue

The anonymous user has the option to request and receive an e-ticket in various ways. He may receive it either through a website, which is designed both for desktop PCs as well as for access through a mobile device. Additionally, a customer can be served through a portable handheld device (e.g. mobile phone/ PDA), running a local application supplied by the bank (i.e. service) or its mobile carrier. Moreover, the same services are provided through SMS technology. Likewise, he/she may request information for his/her ticket's waiting time, either through the website, with an SMS or through the mobile local application, at any time. When the customer uses a mobile device to obtain an e-ticket, some additional features are available, such as storing e-tickets in the local database and managing them (insertion of new tickets and ticket deletion). The SMSs are also available anytime as an evidence for the user.

The m-Services proposed have mechanisms for the daily calculation of the current average waiting and service time for each bank service, at regular intervals. The framework also includes two Web Services which provide a set of functions to the applications they exchange data with.

The Websites are connected to one of the web services, while the mobile local application and the SMS Server are connected to the second one. Both applications use these web services to receive the current waiting time for a bank service, request and receive e-tickets, but also to receive the renewed waiting time of an existing e-ticket. The web services in turn, communicate with the database in order to perform the functions requested. This service oriented approach allows us to initiate as many service instances as needed, in order to evenly distribute the workload to them.

Finally, the m-Services framework is accompanied by a simulation tool for testing purposes. This application simulates the service process of e-tickets stored for the current day, for every company/organization, bank in this case. This simulation allows the calculation of the final waiting and service time of each e-ticket, in order for the mechanisms mentioned above to operate properly. The communication between the application and the database is achieved through a web service.

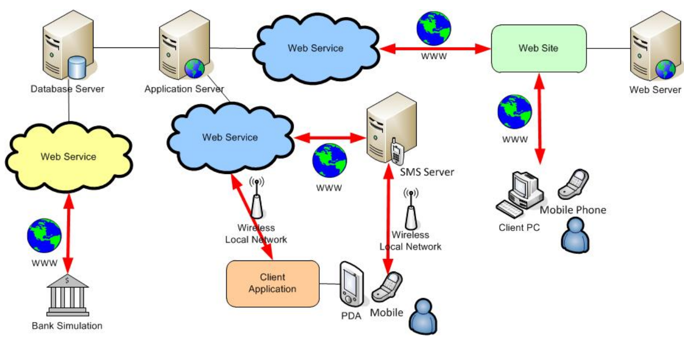


Fig. 3. The Architecture of the Integrated System: Components Interconnection

The general architecture of the proposed m-Services regarding its components interconnection is shown in Fig. 3. In Fig. 4 the general architecture of the integrated system with regard to the set of services it offers is presented.

4 Theoretical Background

This section presents the theoretical background used for the computation of the average waiting and service times for each Bank Service, as well as the renewed waiting time for an e-Ticket after its holder's request. Assume that there exists a bank service with L servers – desks. Assume that a new customer C arrives. The moment he requests a new e-Ticket, there are NW customers waiting and some customers are being served. We assume that until that moment, NS customers have already been served from the beginning of the process. Assume that the service time of the i-th customer is xi. Until that moment, the average service time for the current day is:

$$\overline{x} = \frac{1}{N_S}\sum_{i=1}^{N_S} x_i \tag{1}$$

where:

N_S is the number of customers who have been served until that moment

x_i is the service time of the i-th customer, $1 \leq i \leq Ns$

In order to compute the average service time, we must take into account the service times of the previous days. If we take into account the past five days, Eq. (1) is transformed into:

$$\overline{x} = \frac{1}{2}\left[\left(\frac{1}{N_S}\sum_{i=1}^{N_S} x_i\right) + \left(\frac{\overline{x_{n-1}}+\overline{x_{n-2}}+\overline{x_{n-3}}+\overline{x_{n-4}}+\overline{x_{n-5}}}{5}\right)\right] \tag{2}$$

where $\overline{x_j}$ is the average service time of the j-th previous day. In Eq. (2) we presume that the two addends are equivalent.

In order to calculate the waiting time for the customer C who wants to get an e-ticket, we calculate the time that will pass until all customers who are at the current moment in queue or in desks are served, taking into account the number of desks. The waiting time W_C for the new customer C is:

$$W_C = \frac{N_W \cdot \overline{x}}{L} + \overline{x} = \frac{\overline{x}(N_W + L)}{L} \tag{3}$$

where:

N_W is the number of waiting customers until the current moment

$\bar{x}$ is the average service time calculated by Eq. (2)

L is the number of desks

Eq. (3) represents the worst case where the entire service time for customers who are served the current moment is taken into account. However, if we take into account only the customers in the waiting queue, Eq. (3) becomes:

$$W_C = \frac{N_W \cdot \overline{x}}{L} \tag{4}$$

When the queue is too long in regard to the number of desks, that is $N_W >> L$, then Eq. (3) tends to become Eq. (4).

In case, customer C is already in a waiting queue and wants to be informed for his renewed waiting time, we can use Eq. (3) again, taking into account that term N_W refers to customers waiting in queue and precede customer C.

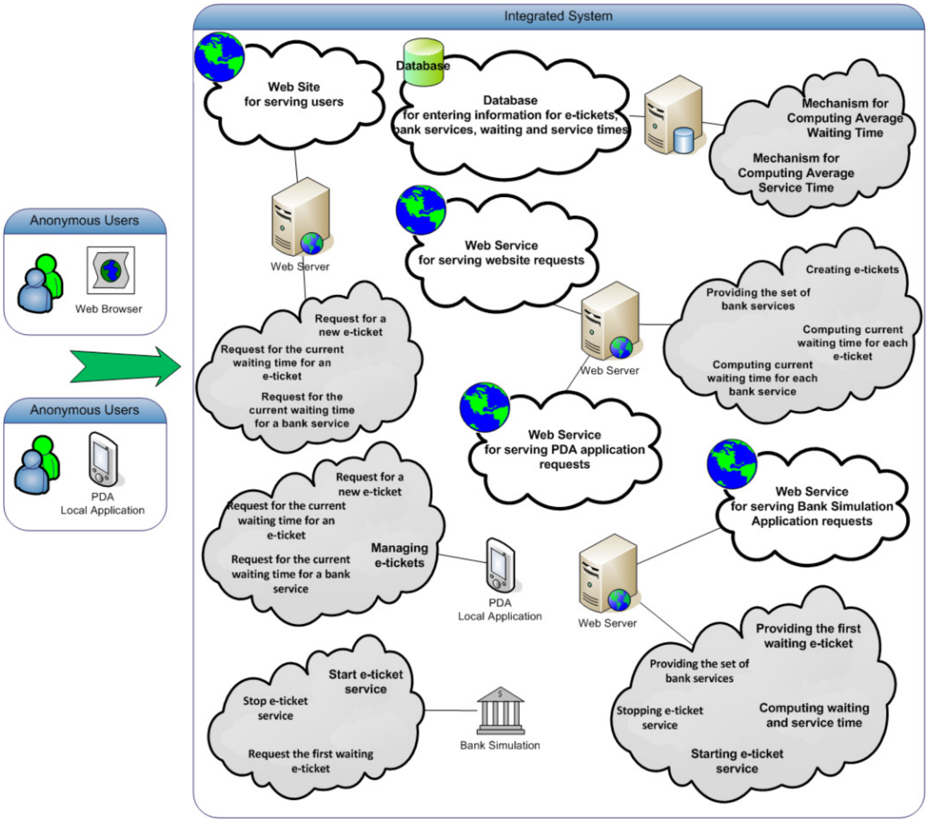


Fig. 4. The Architecture of the Integrated System: Set of Services

5 Functionality and Prototype

5.1 Web Services for Serving Requests from the Website and the Mobile Local Application

The mobile services include two types of XML Web Services which provide a set of functions to the applications with which they exchange data. The Websites are connected to the first web service, while the mobile Local Application and the SMS Server are connected to the second one. Both applications use these web services to receive the current waiting time for a bank service, request and receive e-tickets, as well as receive the renewed waiting time of an existing e-ticket. The two web services in turn, communicate with the database in order to perform the functions requested.

The services provided by the two Web Services connected to the Websites the SMS Server and the mobile Local Application, are:

- Providing the Set of Bank Services
- Creating e-Tickets
- Computing the Current Waiting Time for a Bank Service
- Computing the Current Waiting Time for an e-Ticket

5.2 E-Ticket Structure

The e-ticket code is the number that corresponds in an electronic ticket and is returned in the user who requested the electronic ticket. With this code the user can be informed for his new waiting time. The code is composed of 13 digits and has the following form: YYYY MM DD SS NNNN, where:

- YYYYMMDD is the date during which the e-ticket is valid. It is expressed in year (YYYY), month (MM), day (DD).
- SS is the Bank – Service the electronic ticket is requested for.
- NNNN is the number of the electronic ticket.

5.3 Websites

Website for Access through Mobile Devices. The m-Services include a Website, via which anonymous users can request e-Tickets and be informed at any given time for the renewed waiting time that corresponds to their ticket or the bank branch that interests them, using access to the Internet from their mobile device. Snapshots of the procedures are shown in Fig. 5 and Fig. 6.

One can observe that the web pages of the presented web site contain merely the information that is absolutely essential: there are only few and simple graphics and in general the interface is less impressive than the ones we are accustomed to explore from our personal computers. This way of designing is essential because the screen of many mobile devices (mostly old ones) have limited space, the keys are difficult to use in comparison to the mouse or a touch screen and finally the memory and the processing power of a mobile device is usually limited. The Website provides the following services for its users:

- Request for current Waiting Time for a Bank – Service
- Request for New Electronic Ticket
- Request for the current Waiting Time for an Electronic Ticket

Fig. 5. Captions on Nokia and Sony Ericsson mobiles

Fig. 6. Snapshots on simulator for the mobile Website

The mobile services has also a second Website through which anonymous users can request e-Tickets and be informed whenever they want for their renewed waiting time (Fig. 7). It is designed for desktop pc users.

The Websites are connected to the corresponding Web Service to receive the current waiting time for a specific Bank Service, to request and receive an e-Ticket, as well as to receive the renewed waiting time for a specific e-ticket.

5.4 Locally Stored Mobile Application

The Integrated System has a mobile Local Application through which users who have installed it in their mobile devices can request e-Tickets and be informed whenever they want about their renewed waiting time (Fig. 8). This application offers additional functionality, such as storing e-Tickets in the local database and managing them (new ticket insertion, ticket deletion).

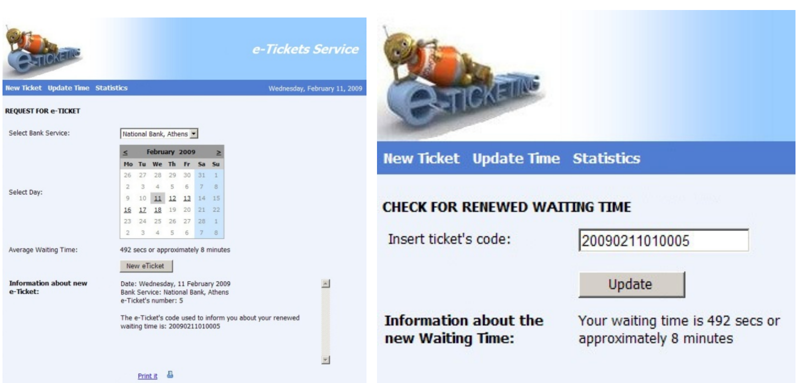


Fig. 7. Snapshots of the Website (a) right after the creation of the e-Ticket and (b) after user's request for the renewed waiting time

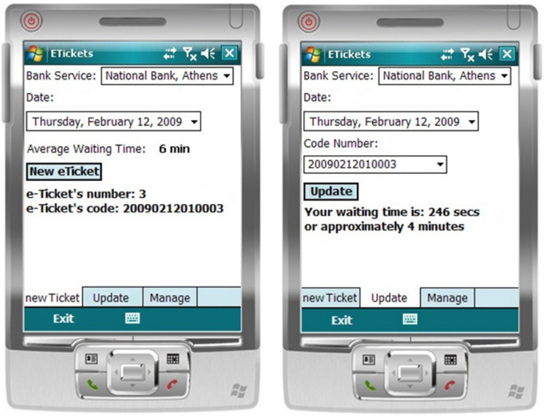


Fig. 8. Snapshots of the mobile Local Application (a) right after the creation of the e-Ticket and (b) after user's request for the renewed waiting time

The mobile Local Application is connected to the corresponding Web Service in order to receive the current waiting time for a specific Bank Service, to request and receive an e-Ticket, as well as to receive the renewed waiting time for a specific e-ticket. The services provided by the PDA Local Application are:

- Request for receiving the Current Waiting Time of a Bank Service
- Request for a New e-Ticket
- Request for receiving the Current Waiting time of an e-Ticket
- Management of the locally stored e-Tickets

5.5 SMS Server Application

The m-Services have SMS based tools. Any mobile phone can be used to receive this kind of service and the only thing the user has to be aware of is the available services and the telephone number providing these services. It is worth mentioning that unlike the alternatives, SMSs constitute a proof of the entitled service themselves. The users only have to bring their mobile phones along. The following table shortly presents the available services, with the parameters that they receive and a short description for each one of them.

For the service of a new ticket we use as parameter the bank name, for example tickets of the National Bank of Greece are offered (with coding ETE) or Agricultural Bank of Greece (with coding ATE) etc. The city name is used as a parameter as well, for example Patras (with coding PAT) or Athens (with coding ATH) etc., the address, for example Corinthu (with coding KOR) and finally the date. In the case of the city we consider the city of Patras to be the predefined one and in the case of the date, when one is not provided by the user we assume the current date is the desired one. The process of using the services and corresponding answers are presented in Fig. 9.

Table 1. Services offered through SMS

Service Code	*Service Name*	*Service Parameters*	*Details*
NEW	New Ticket	<Bank> <City> <Address><Date>	Returns a new Ticket Number, for a specific Bank Branch as well as a specific date
TIM	User's Waiting Time	<Ticket Code>	Returns the waiting time for the specified user, whose Ticket Code is provided as the service parameter in the SMS sent by the user.
TAV	Average Waiting Time of a Branch	<Bank> <City> <Address><Date>	Returns the Average Waiting Time for a specific Bank Branch, which is specified by the user.

6 Evaluation Process and Outcome

In this section we present the results of validation of the proposed system in a realistic scenario. For this purpose, we have used the services provided by the secretariat of the department of Computer Engineering and Informatics instead of a bank as presented above. The evaluation is small scale and includes two services; applying for a new certificate and applying for a grading sheet. The proposed approach can scale up to include all the available services of the secretariat either in a single queue, in case there is only one employee interacting with the students or more than one queues to support time-consuming procedures seperately from the others aiming at .

The overall satisfaction grade for the services (on a scale of 5 = very satisfied to 1 = very dissatisfied) ranged from 4.0 to 5.0, with an average value of 4.8 and provided some noteworthy results. The users evaluated the provided feedback for their waiting time to be very helpful. Many of them stated that the implemented integrated system effectively meets an important need of theirs and in less time than before. Within the campus, it is often usefull to be aware of the waiting time for different services in order to be able to schedule liabilities in a less tedious way. This is a feature that can be even more important in a greater extend, e.g. in a city center. Moreover, a notable proportion of the participants mentioned that it is importanted to have different means of access to the system available, in order to spend affordable amounts of money by allowing utilization of technology any user can have access to.

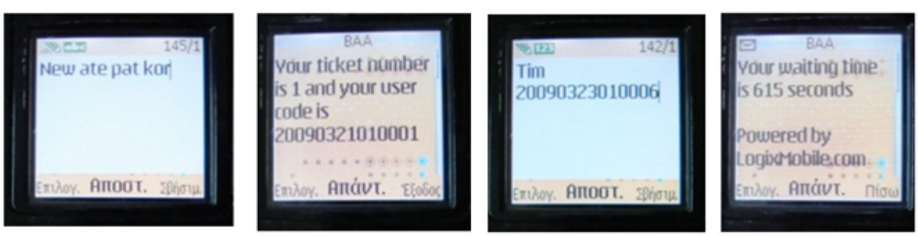


Fig. 9. Snapshot of the SMS services

7 Conclusions and Future Work

In conclusion, the objective of the proposed work constitutes the study, planning and implementation of techniques and algorithms for more efficient management of applications and services of electronic government with use of techniques for network-oriented information systems and mobile phones in particular. We have proposed an integrated queue management with live feedback directly to the citizens mobile phone or web access device. Businesses could benefit further from the proposed approach while using it for implicit developing a customer database and or combining it with loyalty card system besides customer satisfaction. The presented prototype serves as a proof of concept that the approach is feasible and efficient. The key feature of the solution is that it is designed to take advantage of all different mobile phones already existing and maximize the effects of their wide penetration. The abstract design of the approach allows its use in a wide variety of e-gov and e-services applications.

Future steps include enhancing the alert mechanisms using Multimedia Message Systems (MMS) in order to be able to give voice updates which would be especially usefull for visually impaired and elderly users. Moreover non-GSM based solutions are also interesting such as local bluetooth transmission, though there is an extra research issue as bluetooth applications have not received wide acceptance by the public. Interesting is the case of multiple queue management and workflow optimization using the mobiles in order to generalize the case presented in [2].

Acknowledgements. This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. “Investing in knowledge society through the European Social Fund”.

References

1. Pedone, F.: Optimistic Validation of Electronic Tickets. In: Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems, pp. 110–119 (2001)
2. Miwa, K., Takakuwa, S.: Simulation modeling and analysis for in-store merchandizing of retail stores with enhanced information technology. In: Mason, S., Hill, R., Mönch, L., Rose, O. (eds.) Proceedings of the 40th Conference on Winter Simulation (Miami, Florida), pp. 1702–1710 (2008)
3. Kieslinger, M., Polazzi, L.: Supporting time-based coordination in everyday ser-vice interactions: the fluidtime system. In: Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, DIS 2004, Cambridge, MA, USA, pp. 225–232. ACM, New York (2004)
4. Jouini, O., Dallery, Y.: Predicting queuing delays for multiclass call centers. In: Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools (2006)
5. Southard, P.B., Siau, K.: A Survey of Online E-Banking Retail Initiatives. Communications of the ACM 47(10), 99A–102A (2004)

6. Tiwari, R., Buse, S., Herstatt, C.: Mobile Services in Banking Sector: The Role of Innovative Business Solutions in Generating Competitive Advantage. In: Proceedings of the International Research Conference on Quality, Innovation and Knowledge Management, New Delhi, pp. 886–894 (2007)
7. Nie, W.: Waiting: integrating social and psychological perspectives in operations management. Omega 28(6), 611–629 (2000)
8. Pearce, P.L.: Towards the better management of tourist queues. Tourism Management 10(4), 279–284 (1989)
9. Cheung, M.F.Y., To, W.M.: Customer involvement and perceptions: The moderating role of customer co-production. Journal of Retailing and Consumer Services 18(4), 271–277 (2011)
10. Stolletz, R.: Analysis of passenger queues at airport terminals. Research in Transportation Business & Management 1(1), 144–149 (2011)
11. Kushchu, I., Kuscu, M.H.: From e-Government to m-Government: Facing the Inevitable. In: Proc. of the 3rd EU Conf. on e-Government, pp. 253–260 (2003)
12. Tripathi, A., Suman Kumar Reddy, T., Madria, S., Mohanty, H., Ghosh, R.K.: Algorithms for validating E-tickets in mobile computing environment. Information Sciences, Including Special Issue on Chance Discovery - Discovery of Significant Events for Decision 179(11), 1678–1693 (2009)
13. Kreitzman, L.: The 24 hour society. Profile Books, London (1999)
14. Boukas, L., Kambourakis, G., Gritzalis, S.: Pandora: An SMS-oriented m-informational system for educational realms. J. Network and Computer Applications 32(3), 684–702 (2009)
15. Q-Matic (October 14, 2011), `http://www.q-matic.com/en/Int/What-we-do/About-CFM/`
16. Nemo Queue (October 14, 2011), `http://www.nemoq.com/`

Enabling Probabilistic Process Monitoring in Non-automated Environments

Andreas Rogge-Solti and Mathias Weske

Business Process Technology Group,
Hasso Plattner Institute, University of Potsdam, Germany
{andreas.rogge-solti,mathias.weske}@hpi.uni-potsdam.de

Abstract. Business processes are crucial for every organisation as they represent the core value generating processes. Managing business processes is important to be efficient and to compete with a globalized market. Business process monitoring is an essential means to understand and to improve working procedures. It helps detecting deviations from planned procedures and brings transparency into the state and progress of running process instances. However, without automated execution of business processes via a workflow engine, the absence of execution information hampers monitoring. Often, the automated execution of business processes is neither feasible nor desirable. However, a few monitoring points can be used, when process participants interact with IT-systems.

In this paper, we propose a novel approach to business process monitoring using probabilistic estimations to fill information for missing monitoring points. The applicability of the approach is evaluated with a case study in a German university hospital.

Keywords: business process monitoring, probabilistic analysis, sparse events.

1 Introduction

Monitoring of business processes and process activities [1] emerged as an instrument in business process management to gain insights into running process instances. It allows for live evaluation and keeping track, whether the process executes as planned. It can also be used to predict resource shortages and provide decision support for timely operational changes in process instances [2].

However, current business process monitoring solutions require detailed log files containing entries for each monitored activity in the process, often realized by workflow enactment components that emit events for each activity.

Another method that can be used for correlating live event streams to process models is process mining and it does not depend on complete log files and can deal with noise, i.e., missing or multiple events. However, it is used mainly for conformance checking between process models and logs, by computing *fitness* and *appropriateness* [3]. The difference to our approach is that it does not deal with untracked process fragments, i.e., parts of processes that *never* occur in the logs.

We want to address this gap and provide a monitoring approach for such use cases. More concretely, we enable monitoring of untracked parts. The approach is based on

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 226–240, 2012.

probabilities; it allows to compute the probable state of the unmonitored parts of the process for a given time. We resort to an analytical model, i.e., calculating the expected behaviour of the process parts, where no events are available that indicate process progress. The question we want to answer in this paper is: *Which activities in the process are probably being executed at a given time?*

If we know the answer to this question, we can also derive an answer to the question: What is the probability of finishing the process in a given time duration? It also might help to know that there is a (high) chance that a process instance will be performing a task in near future that requires a specific resource.

The applicability of the approach is evaluated in a real world case study in a university hospital in Germany. In hospital environments, or more generally, if people perform manual activities in processes, the utilization of workflow enactment systems is neither feasible nor desirable [4]. However, some important events might be available in IT-systems nonetheless. This is the case in the liver transplant surgery procedure in the Jena University Hospital, where several important events during the surgery, e.g., performing the cut, discharge from the room, etc. are recorded in the hospital information system.

The remaining paper is organized as follows. Sect. 2 introduces preliminary definitions and notions required for describing the scenario and approach formally. Next, in Sect. 3, we describe the related work with focus on analytical models, process mining and monitoring systems. Sect. 4 describes the proposed probabilistic approach to monitoring. In Sect. 5, the applicability of the approach is evaluated with a case study in the Jena University Hospital. Sect. 6 concludes the paper and discusses limitations and future work.

2 Preliminaries

In this section, we give a formal description of the setting in order to describe the monitoring approach explicitly. Therefore, we first explain the model level by introducing the notions of process model, activity lifecycle, and events. Then, we relate these notions to the corresponding instance level, i.e., the notion of process instances and events.

The elicited process models for the clinical pathways contain information about the activities that need to be performed by the clinical personnel. The process flow can be sequential, or split into mutually exclusive or parallel branches at gateways that can also be used to merge alternative or parallel flows. Thus, we use the following definition of a process model. (Let $\dot\cup$ be the disjoint union.)

Definition 1 (Process Model). *Let $\mathcal{P}$ be the set of process models. A process model $P \in \mathcal{P}$ is a connected, directed, acyclic graph (N, E) where $N = A \,\dot\cup\, G \,\dot\cup\, \{n_{in}, n_{out}\}$ is a finite set of nodes, with activities A, gateways $G = G_{XOR} \,\dot\cup\, G_{AND}$, with exactly one start event n_{in} and one end event n_{out}, and $E \subseteq (N \setminus \{n_{out}\}) \times (N \setminus \{n_{in}\})$ is a set of edges connecting the nodes.*

Each node $n \in N$ lies on a path from n_{in} to n_{out}. There are no loops allowed in the process model (acyclic property) and there is exactly one edge entering and leaving an activity, i.e., $\forall a_i \in A, \forall n_i, n_j \in N : ((n_i, a_i) \in E \wedge (n_j, a_i) \in E) \Rightarrow$

$n_i = n_j \wedge ((a_i, n_i) \in E \wedge (a_i, n_j) \in E) \Rightarrow n_i = n_j$. *Gateways can be* either *of branching or of merging nature.*

We assume that the process models are sound*, c.f., [5].*

The branching and merging nature of gateways is just a syntactical restriction, because every gateway that is both, can be split into two sequential gateways, one gateway merging the branches and a second one splitting them.

This set of modeling capabilities suffices for modeling most processes [6] and more complicated modeling constructs are not necessary to illustrate the proposed techniques.

Each activity $a_i \in A$ follows an activity lifecycle described by several states and state transitions. While detailed activity lifecycles were introduced in literature [7,8], we resort to a basic one for our monitoring purposes.

Definition 2 (Activity Lifecycle Model). *The activity lifecycle model is described by the states $S = \{init, enabled, running, terminated\}$, where $init$ is the start state, and the corresponding set of transitions $T = \{begin, enable, terminate\}$ change the states of the activity lifecycle according to the transition functions $t : T \times S \rightarrow S$. Where $t(enable, init) = enabled$, $t(begin, enabled) = running$ and $t(terminate, running) = terminated$.*

The state diagram in Fig. 1 depicts the possible states and transitions in the activity lifecycle model.

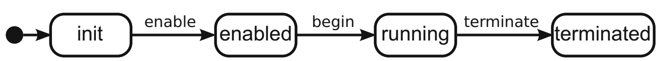


Fig. 1. State transition diagram of the activity lifecycle

Definition 3 (Possible Monitoring Point). *The set of possible monitoring points PMP for each process model P is defined as follows: $\forall P \in \mathcal{P} : PMP(P) = A \times T$, s.t. A is the set of activities in P and T is the set of transitions in the activity lifecycle. This means that every state transition of each activity in a process model is a possible monitoring point.*

We assume that the traversal of edges and gateways (except the joining gateway of parallel branches) does not take time. Therefore, the $terminated$ state of activities is linked to the $init$ state of the immediately following activities. Immediately following means either there is an edge between two activities a_1 and a_2, or there is a path from a_1 to a_2 on which there are no activities, but gateways. The semantics of this link is, that once the $terminated$ state of one activity is reached, the linked activities enter their $init$ state. Fig. 2 shows the combined activity lifecycle of two sequential activities a_1 and a_2. The link is depicted as dotted arrow between a_1 and a_2. Fig. 2 also shows the possible monitoring points at the state transitions of the activities lifecycle.

In order to specify the events triggering the state transitions in the activity lifecycle of a certain process instance, we first need to introduce the notions of event type and process instance.

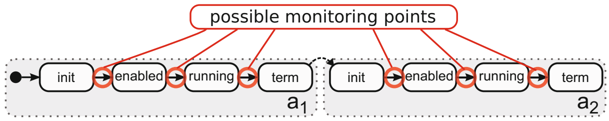


Fig. 2. Possible monitoring points in the process

Definition 4 (Event Type). *Let $et_1 \ldots et_n \in$ EType be a finite set of event types.*

An example for this set is EType $=$ $\{et_1$=*"new patient entry stored"*, et_2=*"surgery scheduled"*, et_3=*"anestetic given to patient"*, et_4=*"patient enters surgery room"*, et_5 = *"patient left surgery room"*$\}$ as in Fig. 3.

Definition 5 (Process Instance). *Let $\mathcal{I}$ be the universe of process instances. An instance $i \in \mathcal{I}$ represents a concrete case.*

We assume that events are characterized by various *properties*, e.g., an event has a time stamp and has a certain event type; it is executed by a particular person and is associated with a process instance. We do not restrict the set of properties, but assume, that the properties time stamp, event type, and process instance identifier exist for all events, c.f. [9].

Definition 6 (Event, Property). *Let $\mathcal{E}$ be the event universe, i.e., the set of all possible event identifiers and $\mathcal{T}$ the time domain. There are property functions $prop_T : \mathcal{E} \rightarrow \mathcal{T}$ assigning time stamps to events, $prop_{EType} : \mathcal{E} \rightarrow$ EType assigning event types to events and $prop_I : \mathcal{E} \rightarrow \mathcal{I}$ assigning instances to events.*

In the case that a workflow system [10] is executing a process model, i.e., the model is instantiated and traversed step by step, the generation of events marking all the state transitions of every visited activity can be done by the workflow system automatically. Thus, the alignment of these generated events to the model is straight-forward and monitoring, which activity is currently running becomes a trivial task. If, however, no workflow system is executing the model, the first challenge is to align corresponding process related events to the model. This is done through the definition of monitoring points in the model.

Definition 7 (Monitoring Point). *Let P be a process model, $PMP(P)$ represents possible monitoring points in P. Monitoring point $m : PMP \nrightarrow$ EType is a partial function mapping a possible monitoring point in the process model P to an event type.*

Note that m is a partial function, i.e., only defined for a subset of the input, because usually not every transition in the activity lifecycle of activities in a process is captured by events in underlying systems.

The idea is that the monitoring system subscribes to all events defined corresponding to the monitoring points. Let P be a surgery process model with activities $A = \{a_1 =$ *"Schedule surgery"*, $\ldots$, a_n = *"Perform stationary care"*$\}$ as in Fig. 3. The monitoring points function m in the example in Fig. 3 maps the $terminate$ transition of the activity lifecycle of activity *"Schedule surgery"* to event type *"surgery scheduled"* = et_2, i.e., $m((a_1, terminate)) = et_2$.

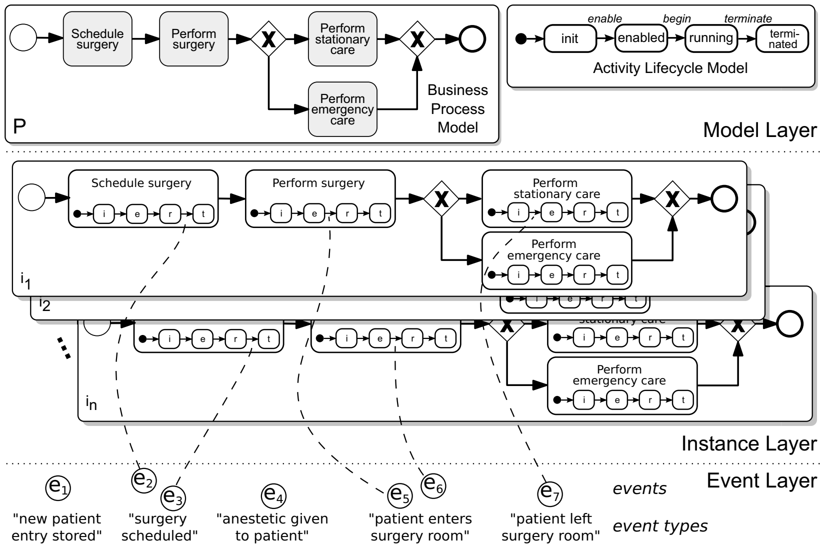


Fig. 3. Simple treatment process model with activity lifecycles for each activity. Dashed lines indicate the monitoring points available in the model, that are fed by events in the surrounding IT-environment.

In this case, the monitoring system subscribes to the event types et_2, et_4, and et_5. Whenever an event $e_i \in \mathcal{E}$ with one of these types occurs in the event layer, the corresponding process instance with identifier $i = \text{prop}_I(e_i)$ is updated by firing the state transition of the activity specified by m. This is illustrated in Fig. 3 by the dashed lines connecting activity state transitions of process instances with events. Note, that not every event is mapped to an activity state transition and neither do events for every activity state transition exist.

For probabilistic monitoring, we further need a model for the behaviour of the system over time, i.e., we need a way to express when activities will be completed. As we cannot know that exactly, we could specify a range over time based on our observations, or more generally specify a distribution of activity durations. In our model, we assume that activity durations, i.e., the duration from *enable* to *terminate*, can be characterized by the normal distribution, as in [11]. Even if, as indicated in [12], time durations are rather log-normally distributed (or follow another distribution type), the proposed concepts and computation mechanisms of the durations remain the same. However, calculating the sum of log-normally distributed activities can then only be approximated numerically [13].

Definition 8 (Activity Duration). *Each activity $a \in A$ in the process model P is annotated with an estimated normal probability distribution $\delta(a_i) = \mathcal{N}(\mu, \sigma^2)$, with the probability density function*

$$f_a(t) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(t-\mu)^2}{2\sigma^2}} \tag{1}$$

The cumulative distribution function $F_a(t) = \int_0^t f_a(t)\mathrm{d}t$ *captures the probability of the activity* a *being already completed at time* t*. For activity* a *with the normal probability density function* $f_a(t)$*, the cumulative probability function is*

$$F_a(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \mathrm{d}t \tag{2}$$

Note, that the used activity lifecycle allows the distinction between preparation time (init to enabled)*, idle time* (enabled to running)*, and execution time* (running to terminated) *of an activity. For simplicity reasons, and without restricting the applicability of the approach, we further do not distinguish between these three states and use the term activity duration for the sum of preparation time, idle time and execution time.*

The distributions of the activity durations have to be assessed initially by medical experts, which tend to outperform automatically estimated durations [14], and can then be aligned to historical data to better reflect reality. However that alignment is out of scope of this paper and we will address that in future work. Further, we assume that the distributions given in the model reflect reality.

We assume, that activity durations are independent and can be simulated by independent random variables with given probability distributions δ_i.

Thus the probability density function $S_i(t)$ of the sum of several consecutive sequential activities' durations $\delta_i(t)$ can be calculated by applying the *convolution operation* (cf. equation 3) successively on their probability density functions:

$$S_i(t) = \begin{cases} \delta_i(t), & \text{if } i = 0 \\ \int\limits_{-\infty}^{\infty} S_{i-1}(u)\,\delta_i(t-u)\,\mathrm{d}u, & \text{if } i > 0 \end{cases} \tag{3}$$

We assume that probabilities are assigned to each outgoing edge of a splitting XOR-gateway. Estimation of these probabilities can be done by experts or also mined from execution logs [15]. Further, we assume faithful execution of the process models, a restriction that we aim to get rid of in future work.

3 Related Work

This paper is located in the discipline of business process intelligence [16], where techniques of analysis, prediction and monitoring play a key role.

3.1 Statistical Process Analysis

Some early statistical analysis techniques for graph based models were developed for project planning purposes, e.g., critical path method [17] or PERT [18]. Martin presented an algorithm to compute the total time in a directed acyclic network with times based on independent random variables [19]. He described the convolution operation algorithm for piecewise polynomial distribution functions. However, he did neither address exclusive parts in the network, nor updating mechanisms for different places in the net.

A successor of PERT is called GERT and supports a quite detailed modeling spectrum [20] including statistical estimations of task durations with deterministic or probabilistic edges. The difference to the approach presented in this paper is that we deal with continuous updates of the model where parts are not monitored. That means, we can do more precise estimations of what has happened by looking backwards (c.f. Section 4.4).

3.2 Simulation Based Systems

De Vin et al. show how simulation models can be used for decision support for virtual manufacturing, when enriched with historical data and snapshot data [21]. The line of thought is similar and the authors stress that an integration of different data sources in a common information center is necessary. However, they do not provide a concrete framework, of how to achieve the ideas, but only sketch possible use cases and outline a necessary architecture with the focus on information fusion [22].

3.3 Process Mining

If no workflow system supports the execution of a process, information can still be derived from enterprise resource planning systems, hospital information systems and the like. The term *process mining* specifies the extraction of information, e.g., process models, from event logs that typically exist in these systems [23,24].

Besides extraction, process mining can also be used as conformance checking method [25,3] and extension of existing models with additional (mined) information like branching probabilities [15] or duration of activities [26,9]. In their work [9], van der Aalst et al. describe a system that learns remaining time durations from logs and enriches the traces in the log with the remaining duration. They assume normally distributed times for activities, but they do not consider activities that are not in the log, but only in the model. They describe an algorithm that abstracts from single activity durations and focus on remaining cycle time. In this paper we describe a method to calculate finer grained estimations of probabilities of single activities being active in one process instance.

3.4 Other Monitoring Approaches

In the integrated European project SUPER[1], an approach to monitoring based on domain specific languages was proposed by Gonzáles [27]. In his PhD thesis, he designed an architecture and domain specific language for monitoring processes. However, he does not deal with missing information, that can be probabilistically estimated. Also from the same project is the work of Pedrinaci et al. [28], where semantic concepts are introduced into business process monitoring. However their focus is on ontologies and they use them to integrate different sources of monitoring [28], and align and evaluate goals [29]. They also considered the combination with process mining in [30], where the challenges for semantic enrichment of process mining are outlined.

[1] http://www.ip-super.org

In a white paper, DeFee and Harmon presented an idea to integrate simulation and business activity monitoring [31], however, they used discrete event simulation and did not address the use case of missing monitoring points in the model.

In the same vein, Kang et al. [32] recently proposed a probabilistic business process monitoring system that updates its estimates upon arrival of a new event. We also advocate the idea to use a probabilistic approach to monitoring. However, their focus is on estimating failure probabilities of instances and they use support vector machines to achieve that, while in this paper we propose a finer grained prediction for activity instance durations and their lifecycle states.

4 Monitoring with Missing Monitoring Points

In this section, we address the question raised in Sect. 1: *Which tasks in the clinical process are probably being executed at a given time?*

To answer this question for a particular process with process model P, we need access to the occurring events that are correlated with the model through monitoring points. For a particular instance $i \in \mathcal{I}$ of P, we require at least one event $e \in \mathcal{E}$ with $\mathit{prop}_I(e) = i$. Further, there has to be a monitoring point defined for P that maps to the event type of e, i.e. $\exists\, a \in A, \exists\, t \in T : m((a,t)) = \mathit{prop}_{\mathit{EType}}(e)$.

The idea is to calculate the probability distribution functions for each possible monitoring point after activity a's state transition t, i.e., each following activity lifecycle state transition. Since we assume independence of the monitored events, we can use the convolution operation on the distributions of two successive events, to derive the distribution of the combined event of both events having happened.

4.1 Monitoring Sequences with Missing Monitoring Points

We demonstrate the approach with normally distributed random variables for predicting time durations for activities in the model. Thus the computation of the probability density function of the sum of two sequential activities $a_1, a_2 \in A$ with $\delta(a_1) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $\delta(a_2) = \mathcal{N}(\mu_2, \sigma_2^2)$ can be done with the following formula, since normal distributions are closed under convolution:

$$f(a_1 + a_2) = \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) \tag{4}$$

With the density function given, the cumulative distribution function can be derived through integration of the density function. The cumulative distribution function captures the probability $P(E)$, that an event E, e.g., the termination of an activity, has already happened at a given time. The complementary event $\bar{E}$ of event E captures that the termination of the activity has not happened yet, i.e., that the activity is *still active*.

$P(\bar{E})$ can be calculated by $P(\bar{E}) = 1 - P(E)$. Let two consecutive points of interest p_1 and p_2 ($p_2 > p_1$) have corresponding cumulative distribution functions F_1, F_2. We get the probability of being in the state between these two points of interest by subtracting the two probability distributions of $P(\bar{E}_2) - P(\bar{E}_1)$ at given point in time.

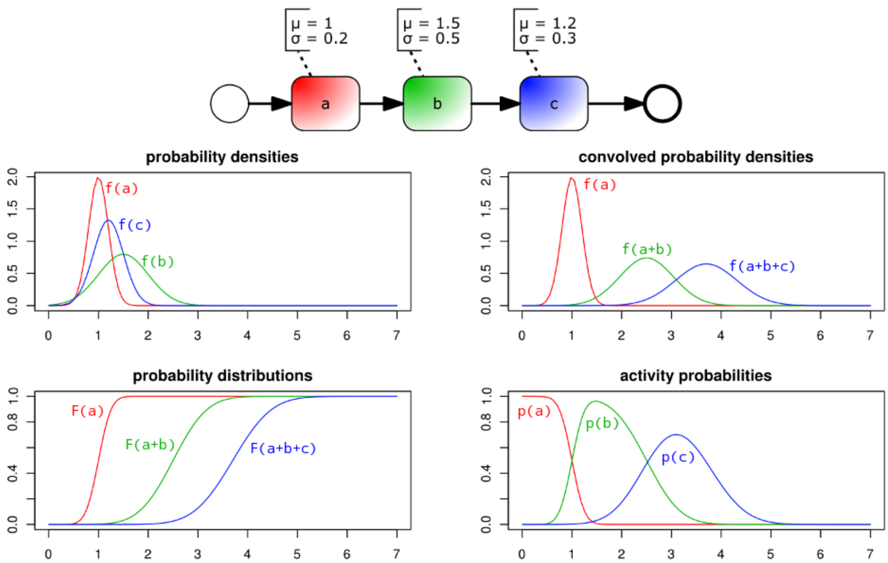


Fig. 4. Process model on top shows sequence of three activities a, b and c with duration parameters of the normal distribution (μ and σ). From top left to bottom right: 1) density functions of the activities, 2) convolved density functions, 3) distributions of end events, 4) probabilities, that activities are being performed at time *t*.

Fig. 4 visualizes the idea in a simple sequential example. The last graph shows the subtracted cumulative probability functions indicating the probabilities $p(a)$ of activity a currently running (red curve), $p(b)$ (green curve) and $p(c)$ (blue curve) as functions over time t. We can use these functions to predict the probabilities of executing the corresponding activities at time t.

With the calculation of probability distributions defined for sequences, we turn to the remaining control flow constructs in the next subsections.

4.2 Monitoring Choices with Missing Monitoring Points

We assume, that for each outgoing edge of each branching exclusive gateway $g \in G_{XOR}$ in process model P a probability is given that indicates which fraction of cases follows that direction. We introduce a branch probability factor $\gamma \in [0, 1]$ that scales the probabilities in a branch after a split. For instance, in Fig. 5 the process model contains two branches after activity $a1$ with branching factors $\alpha = 0.4$ for the upper branch and $\beta = 0.6$ for the lower branch. Computation of the probabilities of $a2$ and $a3$ are similar to the computation of activities in sequence, with the difference, that the scaling factor γ sets an upper limit to the probabilities. In the example in Fig. 5, the probability of activity b, can be computed by the formula $p(b) = \gamma \cdot (F(a) - F(a+b))$ with $\gamma = \beta = 0.4$ To calculate the branch probabilities of the outgoing branches of an XOR-Gateway, one has to multiply the initial γ branch probability with the probability of the outgoing edge.

At the joining counterpart, the incoming branch probabilities are summed up again to get the probability of the outgoing branch. The probability function of the joining XOR-gateway is computed by the weighted sum of the preceding probability functions on the incoming edges, where the weights are the branch probability factors.

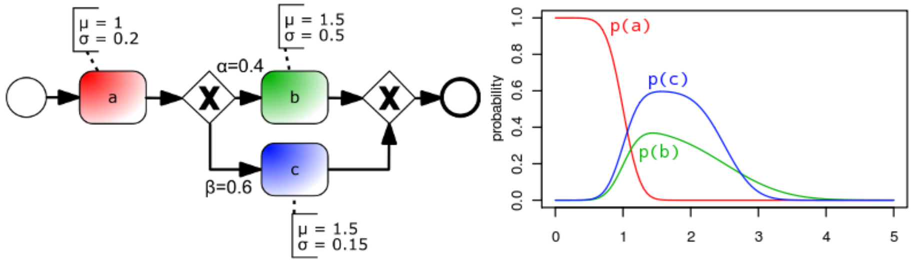


Fig. 5. A choice construct and their respective probabilities

4.3 Monitoring Parallelism with Uncertainty

The calculation of the probabilities of parallel tasks is straight forward. The procedure is the same as for calculating activities in sequence. We just have do that for each outgoing branch after a parallel split $g \in G_{AND}$. The branch probability γ of the incoming branch of the parallel split g is assigned to each of the outgoing branches.

When synchronizing parallel branches, the assumed soundness of the monitored process model assures, that the joined branch probabilities $\gamma_1, \ldots, \gamma_n$ on the synchronized n branches are equal. Otherwise the process would contain a potential deadlock. Thus, the branch probability of the outgoing edge is equal to the incoming ones.

The outgoing *cumulative* distribution function at the synchronization point is the product of the incoming *cumulative* distribution functions, since both branches need to finish before continuing after the joining gateway. Thus to get the probability distribution function, an implementation needs to integrate the probability distribution functions, multiply the integrals and differentiate the result.

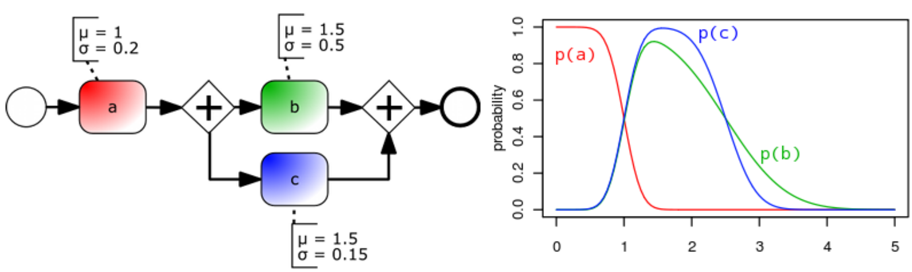
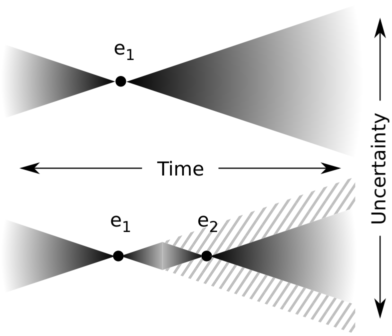


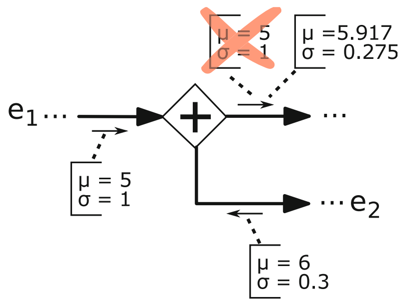
Fig. 6. A parallel construct with three activities and their respective probabilities

4.4 Uncertainty Update

Since the application of the convolution function always adds uncertainty to the result of two random variables, i.e., the squared standard deviations are added, the further the model looks ahead, the more uncertain the estimations become. The upper part of



(a) Uncertainty over time upon arrival of events e_1 and e_2



(b) once second estimation from e_2 is available, the two estimations are combined to get a more accurate estimation for the upper branch.

Fig. 7. Uncertainty update upon arrival of second event. The striped uncertainty area can be eliminated by estimating in both directions from e_2.

Fig. 7(a) shows this graphically. It depicts the uncertainty starting from one monitored event e_1. If, at a later point in time, a second event is detected for a monitoring point, we can update uncertainties in both directions from e_2. An update of the probability distributions in a backward direction becomes useful, if there are parallel branches starting between the two monitoring points, as depicted in Fig. 7(b). The new parameters of the merged estimation can be calculated by multiplying the two normal distributions and rescaling the product to have an area of one. The scaled product of two normal distributions is again a normal distribution with the following parameters:

$$\sigma_{1,2} = \sqrt{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \text{ and } \mu_{1,2} = \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \qquad (5)$$

5 Applicability Evaluation

We have made two assumptions regarding the activity durations. We assumed, that they are independent and that they can be characterized by a distribution function, i.e., they follow a distribution function instead of being completely random. In the following, we show, that these assumptions hold in a real world use case.

For the motivating use case from Sect. 1, the liver transplant surgery in the University Hospital of Jena, detailed process models exist. Fig. 8 shows a part of the surgery process with some monitoring points marked with ovals. The surgery process has been modeled in a detailed way for various reasons, including education of young doctors. However, for the monitoring, the details of the process model are not relevant and the modeling capabilities defined for this paper suffice.

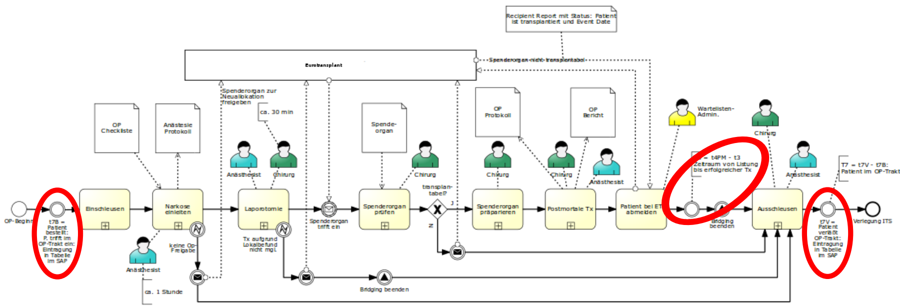


Fig. 8. Part of the liver transplant surgery model with 3 monitoring points (marked with red ovals) modeled in BPMN as annotated Events

We have access to protocol data of surgeries performed from 2009 to 2011. A total of 187 complete cases, each with 11 time stamps for specific events in the surgery, were tracked. The data fits the process models, i.e., can be correlated with activities.

To test for independence, we calculated the correlation values for the durations between these time stamps. The results were low correlation values with a mean of 0.08 and and maximum of 0.24. This indicates independence of activity durations.

In order to analyse the data for characteristic distributions, we have statistically investigated the durations between successive events. Figure 9 shows the duration from begin of the anesthesia procedure of the patient to the approval of the anesthesia team for surgery. The normal distribution fits the durations quite nicely, as one can see in the linear alignment of the QQ-plot on the right.

Literature characterizes the duration of surgeries in hospitals as log-normally distributed [12] with a skewed right tail for cases that take longer due to complications. To test the data for this claim, we used a maximum likelihood estimation for our sample durations from "cut" to "suturation". And although we have only a small sample, the p-value of a fitted log-normal function to the duration between cut and suturation yields p=0.157 for the Cramer-von-Mises test, which is a strong indicator that the data is indeed log-normally distributed. The histogram of the data is shown in Fig. 10 with the estimated distribution and the QQ-Plot. The data contains a a second heap at around 8-9 hours as seen in the histogram in Fig. 10.

From analysis of this data, we can argue, that the assumptions were reasonable, though further analysis with greater sample sizes and more data would be necessary to bolster them.

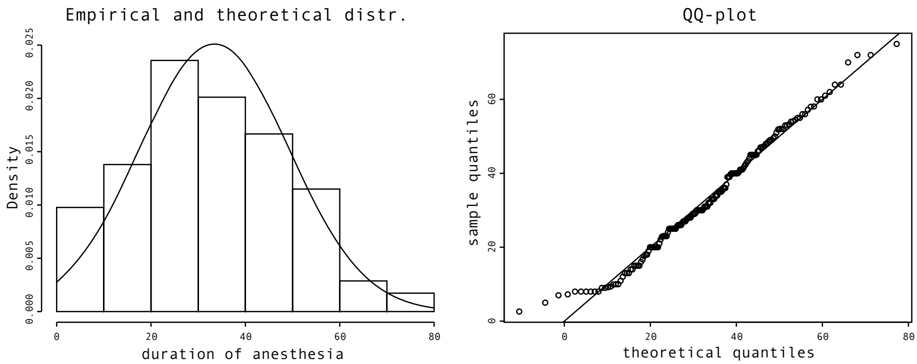


Fig. 9. Duration from “anesthesia begin” to “anesthesia approval”, fit with normal distribution

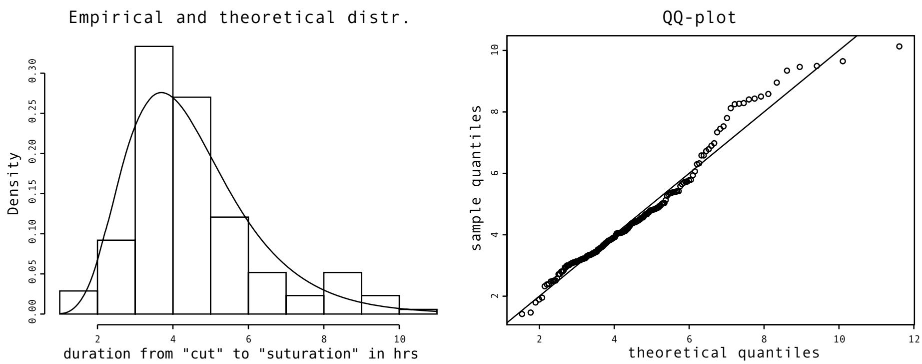


Fig. 10. Duration from “cut” to “suturation”, fit with log-normal distribution

6 Conclusion

In this paper we have investigated the scenario of monitoring a process that is executed without a workflow engine, but described by a detailed process model, which we encountered in a hospital setting. To answer the question, which activities are probably being executed at a given time, we presented a method to calculate probabilities for a core set of modeling capabilities and discussed a special case, when backward estimation yields improved monitoring results for parallel branches.

By applying the proposed techniques in a model where scattered monitoring points exist, relevant and interesting monitoring questions can be answered probabilistically and the estimations get better, the more events are detected at monitoring points.

We evaluated the applicability of our approach by analyzing real durations from a liver transplant surgery and investigated, whether this data fits into characteristic probability distributions and is stochastically independent.

One limitation of the approach is, that we assume faithful execution of the process models, which might not hold in health care scenarios, where exceptions are common [4]. We limited the allowed modeling constructs to a minimum and disregarded loops in the model, but we want to address these limitations in future work. Also the quality of the predictions in unmeasured branches heavily depends on the estimations provided by experts. However, these can be improved by real measurements and alignment to the durations between the monitoring points.

The work presented here is motivated by a hospital scenario, but is is applicable to other domains, too. It is applicable, when information on process progress can be made available as events in an IT-infrastructure and when detailed process models exist that describe truthfully what actually happens. The gain in process transparency should always justify the additional costs of implementing this approach.

References

1. McCoy, D.W.: Business Activity Monitoring: Calm Before the Storm. Gartner Research Note LE-15-9727, vol. 15 (2002)
2. Dahanayake, A., Welke, R.J., Cavalheiro, G.: Improving the Understanding of BAM Technology for Real-Time Decision Support. International Journal of Business Information Systems 7, 1–26 (2011)
3. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33, 64–95 (2008)
4. Lenz, R., Reichert, M.: IT Support for Healthcare Processes Premises, Challenges, Perspectives. Data & Knowledge Engineering 61, 39–58 (2007)
5. van der Aalst, W.M.P.: Verification of Workflow Nets. In: Application and Theory of Petri Nets 1997, pp. 407–426 (1997)
6. zur Muehlen, M., Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 465–479. Springer, Heidelberg (2008)
7. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
8. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York Inc. (2007)
9. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time Prediction Based on Process Mining. Information Systems 36, 450–475 (2011)
10. Hollingsworth, D., et al.: Workflow Management Coalition: The Workflow Reference Model. Workflow Management Coalition (1993)
11. Liu, X., Chen, J., Yang, Y.: A Probabilistic Strategy for Setting Temporal Constraints in Scientific Workflows. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 180–195. Springer, Heidelberg (2008)
12. Strum, D.P., May, J.H., Vargas, L.G.: Modeling the uncertainty of surgical procedure times: comparison of log-normal and normal models. Anesthesiology 92, 1160–1167 (2000)
13. Wu, J., Mehta, N., Zhang, J.: A Flexible Lognormal Sum Approximation Method. In: GLOBECOM 2005, pp. 3413–3417 (2005)
14. Wright, I.H., Kooperberg, C., Bonar, B.A., Bashein, G.: Statistical Modeling to Predict Elective Surgery Time – Comparison with a Computer Scheduling System and Surgeon-Provided Estimates (1996)

15. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006)
16. Grigori, D., et al.: Business Process Intelligence. Computers in Industry 53, 321–343 (2004)
17. Kelley Jr., J.E., Walker, M.R.: Critical-Path Planning and Scheduling. Papers Presented at the Eastern Joint IRE-AIEE-ACM Computer Conference, December 1-3, pp. 160–173. ACM (1959)
18. Moder, J.J., Phillips, C.R.: Project Management with CPM and PERT (1964)
19. Martin, J.J.: Distribution of the Time through a Directed, Acyclic Network. Operations Research 13, 46–66 (1965)
20. Neumann, K.: Recent Advances in Temporal Analysis of GERT Networks. Zeitschrift für Operations Research 23, 153–177 (1979)
21. de Vin, L.J., Ng, A.H.C., Oscarsson, J.: Simulation-based Decision Support for Manufacturing System Life Cycle Management. Journal of Advanced Manufacturing Systems 3, 115–128 (2004)
22. de Vin, L.J., Ng, A.H.C., Oscarsson, J., Andler, S.F.: Information Fusion for Simulation based Decision Support in Manufacturing. Robotics and Computer-Integrated Manufacturing 22, 429–436 (2006)
23. Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 469–483. Springer, Heidelberg (1998)
24. Weijters, A., van der Aalst, W.M.P.: Process Mining: Discovering Workflow Models from Event-Based Data. In: BNAIC 2001, Citeseer, pp. 283–290 (2001)
25. van der Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. Requirements Engineering Journal 10, 198–211 (2005)
26. van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P.: Cycle Time Prediction: When Will This Case Finally Be Finished? In: Meersman, R., Tari, Z. (eds.) OTM 2008. LNCS, vol. 5331, pp. 319–336. Springer, Heidelberg (2008)
27. González, O.: Monitoring and Analysis of Workflow Applications: A Domain-specific Language Approach Ph.D. thesis, Universidad de los Andes (2010)
28. Pedrinaci, C., et al.: SENTINEL: A Semantic Business Process Monitoring Tool. In: Proceedings of the First International Workshop on Ontology-Supported Business Intelligence, pp. 1–12. ACM (2008)
29. Pedrinaci, C., Markovic, I., Hasibether, F.: Strategy-Driven Business Process Analysis. In: Business Information, pp. 1–12 (2009)
30. Alves de Medeiros, A.K., Pedrinaci, C., van der Aalst, W.M.P., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In: Meersman, R., Tari, Z. (eds.) OTM 2007 Ws, Part II. LNCS, vol. 4806, pp. 1244–1255. Springer, Heidelberg (2007)
31. DeFee, J.M., Harmon, P.: Business Activity Monitoring and Simulation. BP Trends Newsletter, White Paper and Technical Briefs, pp. 1–24 (2004)
32. Kang, B., Kim, D., Kang, S.: Periodic Performance Prediction for Real-time Business Process Monitoring. Industrial Management & Data (2011)

Raw Materials for Business Processes in Cloud

Marite Kirikova, Ilze Buksa, and Ludmila Penicina

Institute of Applied Computer Systems, Riga Technical University,
1 Kalku, Riga, LV-1658, Latvia
marite.kirikova@cs.rtu.lv,
{ilze.buksa,ludmila.penicina}@rtu.lv

Abstract. The commonalities and differences between business processes of enterprises depend on the level of abstraction at which the processes are represented. At the lower levels of abstraction the processes are complex and their variations cannot be shown by business process frames of higher levels of abstraction. However there are some parts of processes that are common to many enterprises or several units in an enterprise, for instance, the process parts which must conform to particular regulations. The paper proposes an approach for managing regulation dependent business process parts in cloud. The purpose of the approach is to minimize the total time, which enterprises use for incorporation of regulations in their business processes.

Keywords: business process, regulations, services, regulations graph.

1 Introduction

Different types of regulations [1] are to be taken into consideration when organizing enterprise business processes. These can be governmental laws and corresponding to them regulations, particular industrial standards, and also particular norms and regulations that are issued locally in the enterprise. Many regulations are mutually related and are subjects to more or less frequent changes. When a particular regulation (e.g., governmental law) changes; all companies operating under it have to change their business processes accordingly. For each enterprise implementation of these changes costs time and effort in indentifying and understanding differences with respect to previous regulations and incorporating them into business processes. In case organizational business process models are documented, the incorporation of changes includes the changes of business process models.

In this paper we envision a solution that may help to save organizational effort in their adjustment to changing regulations. The research aim is to find organizational and technical solution that would provide enterprises with business process "spare parts" or raw materials, which comply to valid external (issued outside the enterprise) and internal (issued inside the enterprise) regulations. These "spare parts" would be made available in regulations aware process cloud and could be pushed to and pulled by their users with the help of a number of services tailored to different levels of process maturity.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 241–254, 2012.

The paper is structured as follows. In Section 2 the nature of regulations is discussed and extendable model of regulations landscape proposed. Section 3 demonstrates different types of relationships that may exist between business processes and regulations. Section 4 presents the concept of process raw materials in cloud and envisions possible scenarios of their usage. Related work is discussed in Section 5. Brief conclusions are presented in Section 6.

2 Nature of Regulations

In [1] regulations are defined as a directives published by a legislature. Compliance of business process models to these directives is mandatory. In this paper we use softer interpretation of the term "regulation". We consider regulation as a directive or guidelines that are mandatory for or are chosen to be followed by an enterprise. This complies with the definition of a regulation given in [2]: "A regulation is a document written in natural language containing a set of guidelines specifying constraints and preferences pertaining to the desired structure and behavior of an enterprise. A regulation specifies the domain elements it applies to. Examples of regulations are a law (e.g., the health care law ...), a standardization document, a contract, etc." At a high level of abstraction regulations can be divided into the following categories: *mandatory regulations*, which are issued by governing bodies; "*good to have*" non-mandatory regulations such as various industry standards; and *internal regulations*, which are chosen to be followed by an enterprise. From the enterprise point of view, the first two types of regulations are regarded as external regulations. Internal regulation may depend on (or mirror) external regulations as well as they may be independent of external regulations. Fig. 1 illustrates abovementioned types of regulations.

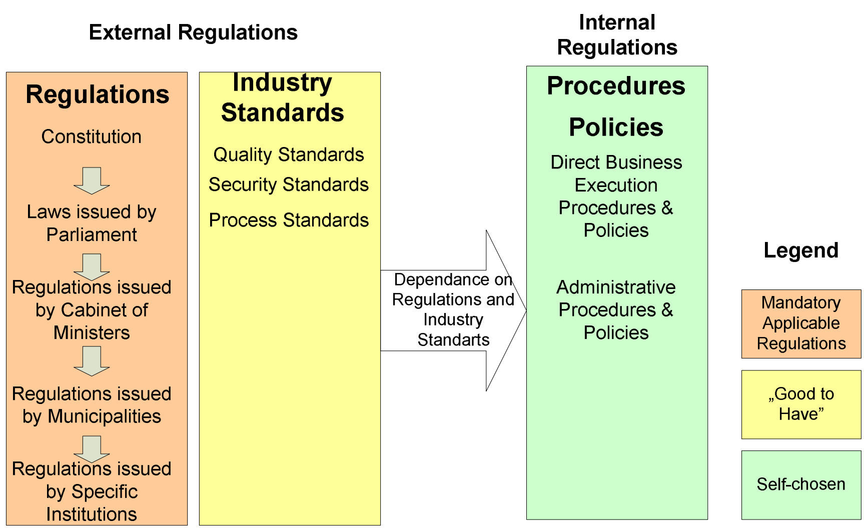


Fig. 1. A landscape of regulations

Regulations are quite difficult to manage due to complexity of their structure, conceptual linkages within one regulation and across several regulations, their legal hierarchy, possibility of multiple interpretations of particular parts of regulations, and relatively frequent changes of contents and linkages of regulations. [1], [3]-[8]. The model/representation of the landscape of regulations has to help to deal with above mentioned difficulties. Currently we are elaborating direct graph based option of regulation landscape representation. In the first glance it might seem that UML class diagram would fit well for modeling the landscape. However, translating the regulations and their relationships into UML cannot be done without additional interpretation of regulations. Any extra interpretation lowers down the usability of the model. Digraph-based approach gives opportunities (1) to avoid extra-interpretations and (2) to use graph algorithms for analysis of the landscape. An abstract example of the use of the digraph-based representation of the regulation landscape is represented in Fig. 2.

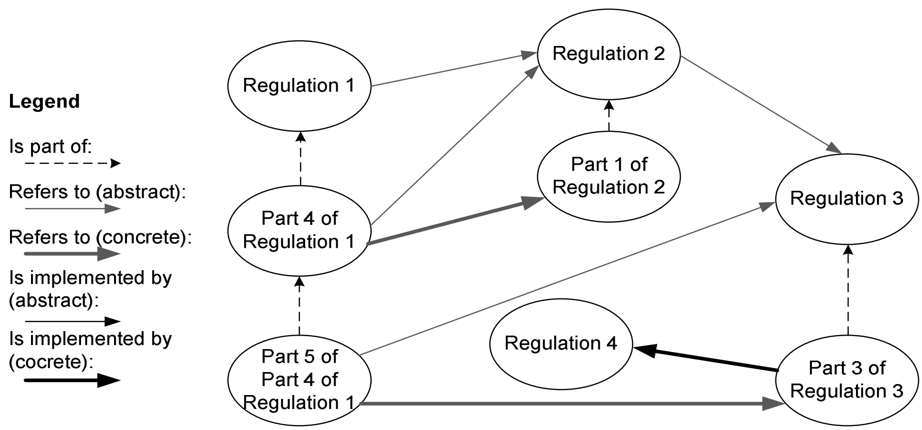


Fig. 2. Regulation digraph

The graph consists of nodes that represent the regulations and parts of the regulations. Directed links between the nodes represent the linkage between the regulations and their parts. We have considered the following four types of links:

- *Is part of (concrete)*: the link shows direct part-of relationships between regulations and their parts
- *Is part of (abstract)*: shows indirect part-of relationships in the regulations landscape
- *Refers to (abstract)*: shows, which regulations (or their parts) refer to which other regulations (or their parts) in general
- *Refers to (concrete)*: shows, which regulations (or their parts) refer to which other regulations (or their parts) in particular
- *Implemented by (abstract)*: shows which regulations define implementation of a particular regulation (or its parts) in general
- *Implemented by (concrete)*: shows which regulations define implementation of a particular regulation (or its parts) in particular

In general, abstract relationships can be derived from concrete relationships if all concrete relationships are defined. However, usually it is not possible to identify all concrete relationships in advance, while abstract relationships are to large extent available in descriptions of regulations (can be retrieved manually or with particular retrieval algorithms) and regulation databases (with already implemented relationships between regulations). Therefore the graph is constructed gradually by extracting the information about regulations and analyzing currently available parts of the graph. There are no restrictions on the size of the graph. For instance for the Law about Accounting [9], which refers to 14 other regulations and is implemented by 18 regulations, there would be 34 *regulation* nodes with corresponding *part of regulation* nodes and at least 14 abstract *Refers to* links and 18 abstract *Is implemented by* links.

3 Relationships between Business Processes and Regulations

The nodes of the regulations digraph may be "populated" (related to) particular business process model parts that represent the regulations in terms of business processes (Fig. 3). These parts of business processes, together with the regulation digraph, may be made publicly available for the enterprises. Using the parts of business process models would prevent enterprises from multiple efforts of translating regulations into business processes.

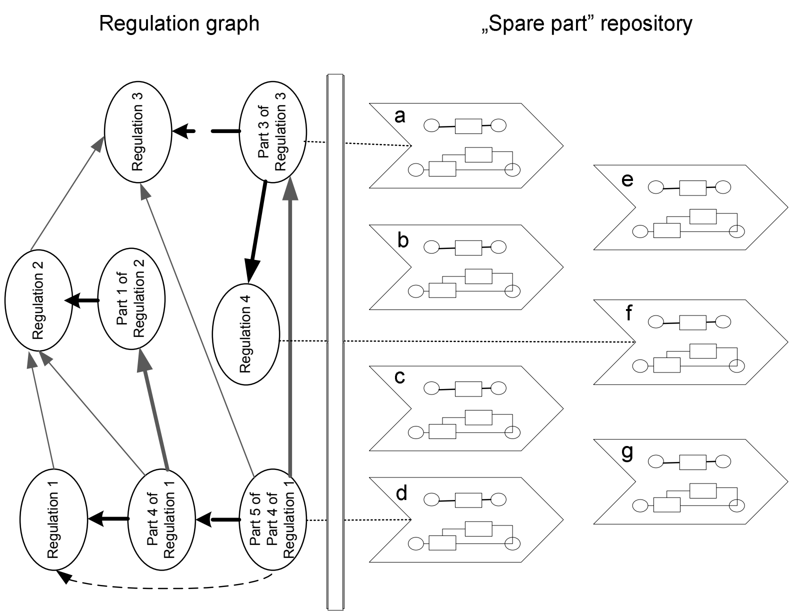


Fig. 3. Regulation digraph with corresponding "spare parts" (raw materials) of business processes

In Fig. 3 there are particular process raw materials that correspond to Part 5 of Part 4 of Regulation 1, Regulation 4, and Part 3 of Regulation 3. In case when an enterprise must comply with Regulation 1; it can use all aforementioned "spare parts" of the processes (d, f, and a).

Regulation digraph can be constructed for external regulations and internal regulations. This is a matter of choice whether an enterprise uses one or several such regulation graphs (each for a particular subclass of regulations as shown in Fig. 1). Enterprise processes may be constructed by combining raw materials from “spare part” repository(ies) and organizational processes that are not prescribed by external or internal regulations. In Fig. 4 an example that illustrates a business process constructed as a combination of ready-made parts from “spare parts” repositories and other enterprise specific tasks is presented. Each task in the figure represents the network of subtasks (i.e. it may consist of several subtasks defined inside the “spare part” or enterprise specific part of the process).

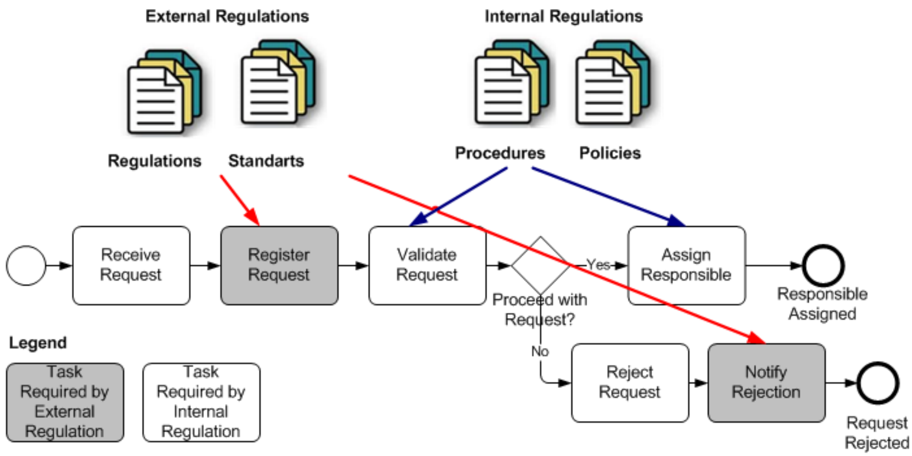


Fig. 4. Relationship between business process and regulations. Each task represents a network of subtasks.

There are domains where regulations are changing rapidly. In these cases process models must be flexible and easy adjustable to changing requirements, since the lack of ability to comply with regulations can lower down the competitiveness of an enterprise. Thus, rapid changes of regulations require rapid changes of related business processes. Regulatory required parts of the business processes must be clear and easy distinguishable from enterprise specific business process parts to give an enterprise a possibility to effectively adjust internal processes to changing economic environment by being free to change internally defined process tasks and still be compliant with regulatory requirements. To achieve aforementioned effectiveness the enterprise models shall be constructed in the way that at any point in process model life cycle it is possible to distinguish, which tasks are required by external regulations, which tasks are required by internal regulations, and which tasks can be freely chosen by their executors. Following such approach gives an opportunity to seamlessly change the “spare parts” in case of changes in regulations by using new “spare parts” from the repository. Therefore business process raw materials corresponding to regulations that are available in the cloud can help to improve agility of enterprises in the rapidly changing economic environment.

4 Providing Process Raw Materials

In this section the cloud solution for provision of regulations compliant business process raw materials is envisioned. Cloud solution is appropriate because of necessity to produce the raw materials in one point only; - this is a key of the proposed approach. The usefulness of the approach is enforced by the fact that cloud solutions are appealing to both large and small businesses [10].

Cloud computing model for managing business process raw materials will provide essential cloud computing characteristics such as on-demand self-service and network access. The cloud infrastructure will be open for use by a specific community of consumers - enterprises subscribed to the cloud.

The use of cloud computing based model for accessing regulations compliant business process raw materials creates new opportunities for enterprises:

- Enterprises will be able to access over network the current versions of regulations and link them with business process models to monitor changes.
- Cloud computing based model allows serving multiple consumers and provides scalability - in the long term more enterprises can be subscribed to the service.
- Regulations are updated in a centralized manner and updates can be sent to the enterprise automatically without requiring human interaction with the service provider.

The proposed cloud computing model is based on the community cloud deployment model described in [11] and is made up of a set of enterprises that consume web-services offering regulation compliant business process parts and monitoring of changes introduced into regulations and corresponding business process parts (Fig. 5).

The cloud computing model depicted in the Fig. 5 is based on the outsourced community cloud scenario described in [11]. This cloud computing deployment model is chosen as a foundation for providing web services offered by the cloud considering the following capabilities of such outsourced site:

- Enterprise can access its private resources such as internal regulations stored on the cloud.
- Multiple enterprises can access the same resources such as external regulations.

The envisioned cloud computing model can be implemented as Software as a Service environment (SaaS). According to [11] in the SaaS environment, a subscriber has control over the application-specific resources that a SaaS application makes available. In this environment software that is supplied to enterprises (subscribers) is a web-service providing enterprises with linking enterprise's business process models with internal and external regulations.

Legal documents such as service agreements and service level agreements will determine terms for organizations subscribing and using web services provided by the cloud. To implement monitoring of the changes made to the externally stored business process raw materials, a web-service must be developed, which continuously checks and harvests for changes and updates introduced to regulations and business process parts; and in case of modifications notifies the enterprise. Monitoring and notification for external regulations is illustrated in Fig. 6. Similar solution can be introduced for internal regulations in an enterprises internal resources or external cloud.

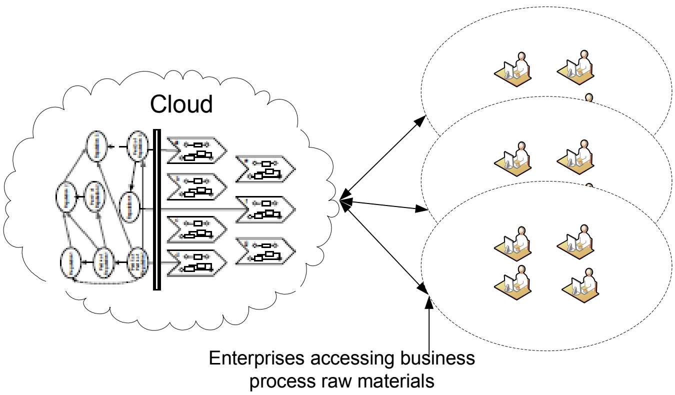


Fig. 5. The general cloud computing model

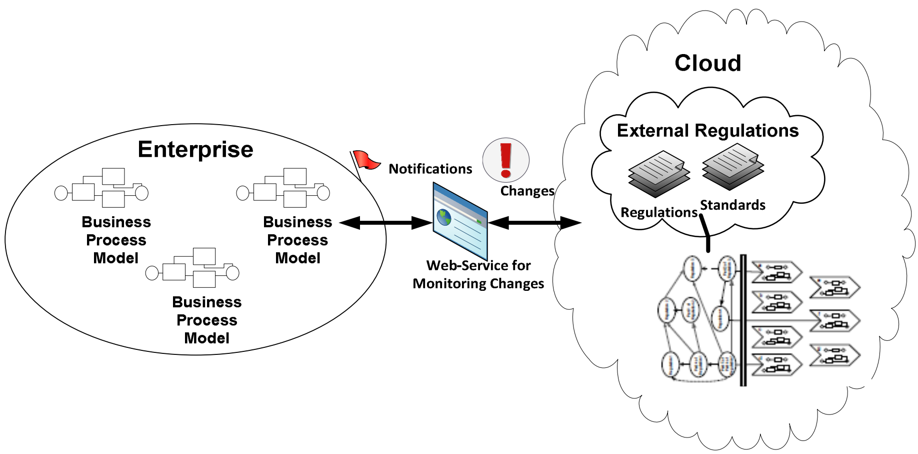


Fig. 6. Change monitoring and notification

Taking into consideration that enterprises have different levels of business process management maturity, the provision of business process "spare parts" can be made in different forms, e.g., these can be business process models as pictures for enterprises that do not use business process modeling tools. For enterprises that do use the tools and adhere to Business Process Model and Notation (BPMN) and technical Business Process Management (BPM) standards - i.e., XML Process Definition Language

(XPDL) [12]; the XPDL can be used to provide design interchange format for business processes in the cloud and in the organizational business process models. XPDL is a standard by the Workflow Management Coalition (WfMC) [12] to interexchange business process description between different business process management tools. XPDL was developed to exchange the business process model graphics and semantics and therefore is proposed to be used as XML schema implemented with a web-service for interchanging business process definition between the enterprise and the cloud. The research by Wil M.P. van der Aalst [13] provides a deep insight into XPDL revealing its limitations - i.e. XPDL is not compatible with all workflow patterns. In this research XPDL is envisioned as a potential standard for business process definition interchange because of the capability of XPDL to store not only the semantics of business process model but also the graphics of the model. The graphics of the business process model will be used when marking the nodes in the process model affected by changes of process raw materials in the cloud. The authors of this paper are aware of existing XPDL limitations and future research will address the issue of using Business Process Execution Language (BPEL) structures when XPDL is not applicable. Using XPDL definition of a business process stored on the cloud, a monitoring web-service shown in Fig. 6 will identify the changes introduced to a model and provide a notification about updates in the business process model. XPDL offers one for one representation of the original BPMN process diagram – it can be written and re-read to recover the original diagram [14]. Every core element of BPMN has a correspondence with a specific element of XPDL specification. XPDL provides vendor and user extension capabilities, so that it is possible to represent new graphical elements by extended attributes embedded in XPDL specification [14].

5 Related Works

The approach described in this paper is cloud based version of automated regulatory compliance management. A conceptual framework and thorough analysis of existing approaches of automated regulatory compliance management are presented in [2]. Relationships between business processes and regulations are analyzed and some suggestions about their monitoring are provided in [3]-[7].

Vendors of several tools (e. g., ARIS [13] and Casewise [16] have made models of enterprise architectures and process frameworks (named as “good to have” regulations in Section 2) and give an opportunity to obtain these models for users of the tools. The cloud solution proposed in the paper does not bind the users of business process raw materials to particular platforms or tools. Potentially, business process frameworks could be one of the groups of regulations that are supported by the cloud solution. Process frameworks provide organizations with reference models of business processes and support optimization and assessment of business processes. Process frameworks are raw materials for business processes that can be stored and managed in the cloud outside the organization, providing the organization on-demand network access to a shared pool of typical default processes. Specific process frameworks include but are not limited to:

- APQC Process Classification Framework (PCF) [17] - offers assessments and best practices for business processes. APQC categorizes a wide range of processes from which organization can identify, compare and optimize its own business processes. APQC PCF is organized hierarchically starting with top level processes such as operating, support and management processes comprising underlying level process categories.
- Supply Chain Operation Reference (SCOR) Model [18] - hierarchical reference model for supply chain processes addressing product movement, interaction with customers and suppliers, etc.
- Value Reference Model (VRM) [19] - addresses planning and governing of value chains in an organization by connecting business processes beyond functional unit boundaries.

Similarly, process management and governance frameworks specify how processes can be managed or governed [20]. Specific regulations and governance frameworks include but are not limited to:

- Basel II [20] - regulation for the banking and financial sectors, which is the required implementation of the EU in all EU countries since January 1, 2007.
- Sarbanes-Oxley Act (SOX) [21] - federal law in the United States of America to ensure the correctness and reliability of published financial data.
- Control Objectives for Information and Related Technology (COBIT) [22] - comprises a range of best practices for IT management and control. COBIT is a top-down approach from the enterprise goals, to the derived IT goals, to their impact on IT architecture [23].

These frameworks have also been considered by tool vendors [13], [16] and compliance analysts.

The notion "regulation" in this paper is used as in [2], However, there exist different interpretations of this term. For instance, regulations are called directives in OMG [24] Business Motivation Model (BMM). Its standard [25] describes the goals of an enterprise and associated implementation strategies. BMM defines the structure of such enterprise governance elements as vision, goals, means the enterprise deploys to meet the enterprise objectives. The means area of BMM is subdivided into missions, courses of action and directives [23], [26]. The directives consist of business policies and business rules. A directive is either a business policy or a business rule [23]. BMM provides an enforcement level for every business rule. This enforcement level can have two characteristics [23]:

- Strict – the business rule must be adhered to. It becomes an organizational procedure.
- Guideline – the business rule should be adhered to, but deviations may occur in justified exceptions.

The following existing enterprise directives are listed in [23] and can be taken into consideration in further research regarding business process raw materials:

- Regulation – a directive published by a legislature, - compliance to it is mandatory
- Self-regulatory rule – standard that enterprises commit to on their own
- Principle – generally acknowledged rule
- Guideline – set of principles
- Standard – set of rules created and published by a standardization organization
- Control model – similar to standard but focuses on the implementation of rules rather than the rules themselves
- Best practice – accepted best approach based on experience
- Organizational control – activity that ensures the directive
- Organizational policy – formal document that describes organization's attitude toward a specific aspect
- Organizational procedure – step-by-step instruction how to implement a task
- Safe harbor – prescribed shortcuts for the adherence to regulations.

The question of the relationship between business process models and business rules (that may represent regulations) is also addressed widely within business process management community. Business rules capture operational decisions while processes capture sequence of execution of activities, events and actors. Operational decisions presented in the form or business rules can be seen as a parts of the business process, which can indicate starting/terminating points of the process, decision points, rules of execution of activities and actors – requirements for process which can be derived from the regulation. To integrate Business process management systems (BPMS) and business rules engines (BRE) the following issues are to be considered:

- Lack of interchange standard between BPMS and BRE to enable integration
- Complexity of combining two approaches: process-driven and rule-driven governance in an enterprise.

The trend across today's businesses is that organizations extract their business rules embedded within the complex application code and spread across multiple systems (making it extremely difficult to introduce changes quickly) from code-based legacy application into BRE. Today BREs provide the ability for a technically competent business analyst, working within business (not IT) to change implicit business rules very quickly [27]. BRE allow the decision logic, which is being used by the process during its execution, to be driven by a central repository where all the rules are stored and managed [28].

Business rules and external business process regulations and standards potentially have such attributes:

- Definition independent of the business process that use them – business rules (as well as regulations and standards) change independently of the business process model [29]

- Maintained and stored in a central location external to the business process model – central location where business rules (and external regulations and standards) are stored provide a way to abstract the decision logic from the applications and helps managing this logic centrally [29].

Following the rapid development of the business rules management BPMN 2.0 [30] version provides four constructs related to business rules modeling within a business process that can be utilized in constructing business process row materials or “spare parts” [29]:

- Business rule task – specific type of a task that evaluates a business rule at a particular point in the process. This type of tasks provides the input data needed to evaluate the rule and receives the result in an output variable (Fig. 7).

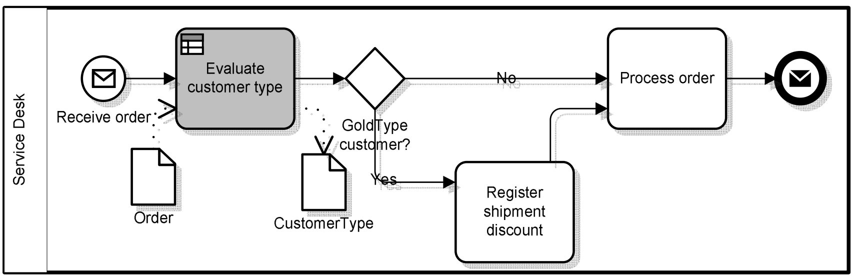


Fig. 7. BPMN Business rule task (marked grey)

- Conditional event – is used when BRE continuously monitors some Boolean data condition and publishes the event when the condition becomes true (Fig. 8).

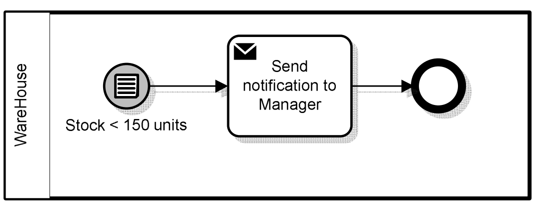


Fig. 8. BPMN conditional event (marked grey)

- Data object – contains the input data passed from business process model to BRE to be evaluated against a ruleset and to send the output data (result) back to the business process.
- Call activity – externally defined subprocess. A call activity represents invocation of either the reusable global task or process. Data must be explicitly passed from the Call Activity to the global task or called process (Fig. 9)

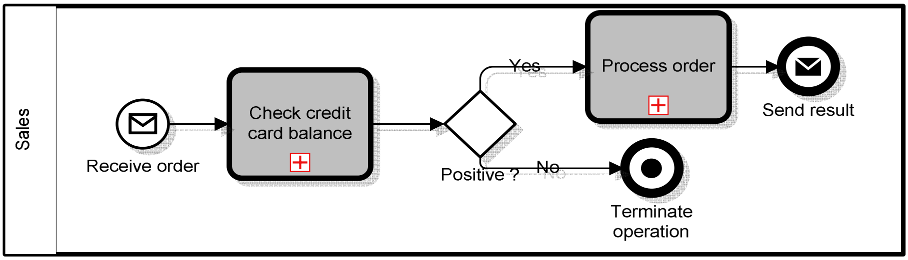


Fig. 9. BPMN call activity sub-process (marked grey)

These main principles of existing BPMN 2.0 solutions help to integrate in the business process model externally located and maintained business rules or global reusable processes used across different business processes. These principles are useful for providing enterprises with combinations of business process “spare parts”.

In its current stage of development, the purpose of the approach described in this paper is not to strive to provide executable business processes. The focus is on business process models that comply with regulations. However, development of cloud solution itself requires implementation of range of processes that have to comply with security and quality regulations.

6 Conclusions

Enterprise business processes must comply with or take into consideration many different regulations. A part of these regulations are common to several enterprises or to several processes in an enterprise. Analysis of these regulations and their inclusion in business process models require large amount of time and effort. In this paper we envision an approach where regulations are translated into business process model “spare parts” or raw materials that can be used by designers of business processes at several enterprises (or several units in one enterprise). The solution is based on the use of the regulation digraph, which is related to the regulations and the business process “spare parts” or raw materials amalgamated in the spare parts repository. This approach can support business process life cycle in the design phase and also in the later phases when changes have to be introduced in the processes due to changes in affecting them regulations. The research presented in the paper only blueprints the approach and has the following limitations:

- It does not discuss how the regulation graph is obtained and does not consider related work on managing relationships in legal documents.
- It does not discuss issues on information representation for enterprises using the business process raw materials.
- It does not discuss navigation rules inside the regulations graph and business process “spare parts” repository.
- It does not describe in detail the architecture of the solution.
- It does not reveal to full extent the complexity of the approach.

Above-listed issues are left out of the scope of the paper, since the number of experiments has not yet reached the stage where reliable conclusions and proven architectures and algorithms can be presented. Nevertheless the approach itself has attracted interest of companies and further research will mainly concern elaboration of concepts presented in this paper.

Acknowledgment. The research presented in this paper is partly supported by Ltd “Komerccentrs DATI grupa”, Latvia in terms of joint research project, contract No. 5/7-2012.

References

1. Basel Committee on Banking Supervision, Basel II Accord, `http://www.bis.org/publ/bcbsca.htm`
2. Kharbili, M.E.: Business process regulatory compliance management solution frameworks: a comparative evaluation. Accepted for The Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012) (2012)
3. Olbrich, S., Simon, C.: Integration of legal constraints into business process models. Transforming Government, People, Process and Policy 1(2), 194–210 (2007)
4. Veal, G.J., Mouzas, S.: Changing the rules of the game: Business responses to new regulation. Industrial Marketing Management 40(2), 290–300 (2011)
5. Boer, A., van Engers, T.: AGILE: From source of law to business process. In: Casellas, N., Francesconi, E.C., Hoekstra, R., Montemagni, S. (eds.) Proceedings of the Third Workshop on Legal Ontologies and Artificial Intelligence Techniques, Held in Conjunction with the Workshop on Semantic Processing of Legal Texts, Barcelona, Spain, June 8 (2009), `http://ceur-ws.org/Vol-465/`
6. Governori, G.: Law, Logic and business processes. In: Proceedings of the 2010 Third International Workshop on Requirements Engineering and Law (RELAW), pp. 1–10 (2010)
7. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. Journal of Visual Languages and Computing 22, 30–55 (2011)
8. Kharbili M.E.: Business process regulatory compliance management solution frameworks: a comparative evaluation. Accepted for the Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012) (2012)
9. Law About Accounting (in Latvian) (2011), `http://www.likumi.lv/doc.php?id=66460`
10. Marston, S., Bandyopadhyay, S., Ghalsasi, A.: Cloud computing — The business perspective. Decision Support Systems 51, 176–189 (2011)
11. Badger, L., Grance, T., Patt-Corner, R., Voas, J.: DRAFT Cloud Computing Synopsis and Recommendations, `http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf`
12. XML process definition language (XPDL) specification, Workflow Management Coalition, `http://www.wfmc.org`

13. van der Aalst, W.M.P.: Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language, Eindhoven University of Technology, http://www.workflowpatterns.com/documentation/documents/ce-xpdl.pdf
14. Stephen, A.W.: XPDL and BPMN, SeeBeyond, United States (2003)
15. ARIS Platform, http://www.softwareag.com/corporate/products/aris_platform/reference_content/archimate/default.asp
16. Casewise products, http://www.casewise.com/products/models-and-frameworks
17. APQC's Process Classification Framework (PCF), http://www.apqc.org/knowledge-base/documents/apqc-process-classification-framework-pcf-cross-industry-pdf-version-520
18. Supply Chain Operation Reference Model, http://supply-chain.org/f/SCOR-Overview-Web.pdf
19. Value Reference Model, http://www.value-chain.org/value-reference-model/
20. Basel Committee on Banking Supervision, Basel II Accord, http://www.bis.org/publ/bcbsca.htm
21. Sarbanes-Oxley Act of 2002, Public Law 107-204 (116 Statute 745), United States Senate and House of Representatives in Congress (2002), http://www.soxlaw.com/
22. Control Objectives for Information and Related Technology (COBIT), http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx
23. Weilkiens, T.: OCEB Certification Guide: Business Process Management - Fundamental Level. The MK/OMG Press (2011)
24. Object Management Group, http://www.omg.org/
25. Business Motivation Model (BMM), http://www.omg.org/spec/BMM/
26. Model Systems: Overview of OMG Business Motivation Model: Core Concepts, http://www.omg.org/oceb/BMM_Overview-Core_Concepts_[081208].pdf
27. Jeston, J., Nelis, J.: Business Process Management: Practical Guidelines to Successful Implementations. Elsevier Ltd. (2008)
28. Juric, M.B., Pant, K.: Business Process Driven SOA using BPMN and BPEL. Packt Publishing (2008)
29. Silver, B.: BPMN Method and Style. Cody-Cassidy Press (2009)
30. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard. Computer Standards & Interfaces 34, 124–134 (2012)

Model-Driven Strategic Awareness: From a Unified Business Strategy Meta-Model (UBSMM) to Enterprise Architecture

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

[1] Department of Computer and Systems Sciences, Stockholm University
Forum 100, SE 1640-40 Kista, Sweden
{constantinos,jelenaz}@dsv.su.se
[2] PReCISE Research Center, Computer Science Department, University of Namur
Rue Grandgagnage 21, B-5000 Namur, Belgium
mpe@info.fundp.ac.be

Abstract. Business strategy should be well understood in order to support an enterprise to achieve its vision and to define an architecture supporting that vision. While business views are identified in many Enterprise Architecture (EA) proposals, business strategy formulations from the area of Strategic Management are overlooked. Thus, IT solutions cannot be traced back to business strategy in a clear and unambiguous way. Our intended proposal, a Unified Business Strategy Meta-Model (UBSMM), aims at establishing such a link. UBSMM is a formalization of the integration of known business strategy formulations with precise semantics enabling its model-level usage to provide strategic awareness to Enterprise Architecture. In this paper we present the development process of UBSMM, and further, we propose conceptual relationships towards Enterprise Architecture (EA).

Keywords: Business strategy, model-driven engineering, meta-model, enterprise architecture.

1 Introduction

Business strategy requires a continuous management of the resources of an enterprise to ensure its realization. Focusing on IT, enterprises fail to establish traceability from business strategy towards IT operations thus hindering their optimal utilization.

Possible ways to address the lack of such links include business-IT alignment approaches, whether that is alignment between business strategy and distinct enterprise models, or business strategy and Enterprise Architecture (EA). However, alignment approaches are falling short in two ways. From one side, business strategy formulations from *Strategic Management*, such as *Strategy Maps and Balanced Scorecards* (SMBSC) [1], Porter's *Value Chain* [2], and others, are overlooked in [3, 4, 5, 6]. On the other hand, approaches referring to such business strategy formulations focus on particular enterprise models [6, 7, 8, 9]. Consequently,

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 255–269, 2012.

alignment approaches establish links between business and IT within particular context, thus not addressing alignment overall in an adequate manner [10].

Business strategy formulations used in Strategic Management are traditionally natural language-based, usually accompanied by schematic representations. In such a form, establishing meaningful traceable links towards IT, as expressed by enterprise models and enterprise architecture, is unattainable due to the ambiguity of the formalisms. Therefore, business strategy formulations need to be formalized, thus transforming their notions and rules from natural language to a process-able form. The degree of formalization may vary according to the purpose of use, from manual to fully automated. Business strategy formulations, such as Strategy Maps and Balanced Scorecards (SMBSC) [1], and the Value Configuration (VC) [2, 11] have been formalized in [12, 13], and [14] respectively, providing unambiguous descriptions of their concepts.

On the other hand, EA provide the principles, methods, and models used to design and realize an enterprise's organizational structure, processes, information systems and infrastructure [15], and are widely present in the industry. EA proposals such as Zachman [16], ARIS [17], TOGAF [18], and Archimate [19], to name a few, include business elements without linking them to business strategy formulations.

The need to establish meaningful links between business strategy and EA to improve practitioners' understanding of their business strategy has been long acknowledged, in the aforementioned EA proposals and the Enterprise Business Architecture proposed in [20]. Yet recent evidence indicates a lack of business strategy insights in EA. Among 176 practitioners of a 2011 EA webinar, answering whether they understand their business strategy, 1% stated they had no business strategy, 6% didn't know the state of their business strategy, 16% stated business strategy is not clearly communicated, 44% stated business strategy is not understood or supported, and only 33% stated their business strategy is well understood [21].

The goal of this paper is to provide a unified business strategy meta-model (UBSMM) which does not currently exist, allowing it to (i) serve as a pivot model between business strategy formulations, and (ii) provide strategic awareness to EA via model-to-model linkages. The purpose of UBSMM is:

- to add precision to business strategy formulations through formalization,
- to contribute and complement business alignment, allowing an enterprise to align various business strategies, or to integrate with the strategies of others,
- to serve as a pivot model between business strategy formulations; an enterprise can be modeled from a resource-based view and through UBSMM to get a competition-view of its current strategy, and vice-versa,
- to enhance strategy communication among actors, when more than two actors are involved, one-to-one mappings between strategy formulations each actor uses are inefficient. Thus, UBSMM can become the common point of reference for such mappings,
- to be extendable to embed more business strategy formulations, therefore, aiming at not being static and at being updated and enriched, thus supporting up-to-date mappings.

The scope of this paper encompasses the development of UBSMM through rigorous schema integration, and further, identification of conceptual relationships towards EA, for which the ISO/IEC/IEEE 42010-2011 standard is used [22]. The paper is structured as follows: Section 2 presents the schema integration process followed to build UBSMM; Section 3 elaborates the conceptual relationships to EA; Section 4 presents related work on EAs using business elements, and Section 5 concludes the paper and discusses future work.

2 Schema Integration (UBSMM)

Unifying meta-models of business strategy formulations can be achieved through schema integration [23], where schema integration refers to both view integration and database integration. However, for the scope of this paper the schema integration process followed is rooted in the foundational work of [24] which identifies four phases: *pre-integration, schema comparison, conforming to schemata,* and *merging and restructuring*. Due to space limitations, this paper provides an illustration of the integration process followed, while a more detailed presentation can be found in [25].

2.1 Pre-integration

During this phase the selection of schemata to be integrated and their representations takes place. According to [25] this selection is based on relevance, completeness and reliability, the integration order, and assignment of preferences and strategic decisions for integration, e.g. the involvement of users or designers along with relevant information collected for an integrated set of constraints depending on the view (user, designer, etc.).

Select Schemata for Integration. The schemata chosen for integration are the Strategy Maps and Balanced Scorecards meta-model (SMBSC) [12, 13] and the Value Configuration meta-model (VC) [14] because in terms of relevance, completeness and reliability they are complete conceptualizations of the aforementioned business strategy formulations validated through correct instantiations of the meta-models as well as through their ontological formalization capable to instantiate each business strategy formulation.

SMBSC (Figure 1) formulates strategy upon establishing four perspectives of an organization, where goals are identified for each perspective and they are altogether related via cause-effect links, the strategy map [1]. Goals are then extended to a set of targets using measures to evaluate their achievement. Initiatives are identified to achieve the targets, the balanced scorecard.

The VC (Figure 2) refers to the Value Chain [2], the Value Shop, and the Value Network [11], formulating strategy based on a setup of value activities and margin aiming at a unique value proposition. Value activities are all the activities a company performs to create value for its buyers, divided into primary, and support, while margin is the difference between the total value and the total cost of performing the value activities. Primary activities capture the activities that bring value to the VC and vary between value chain, value shop, and value network. Support activities aim at supporting the primary activities.

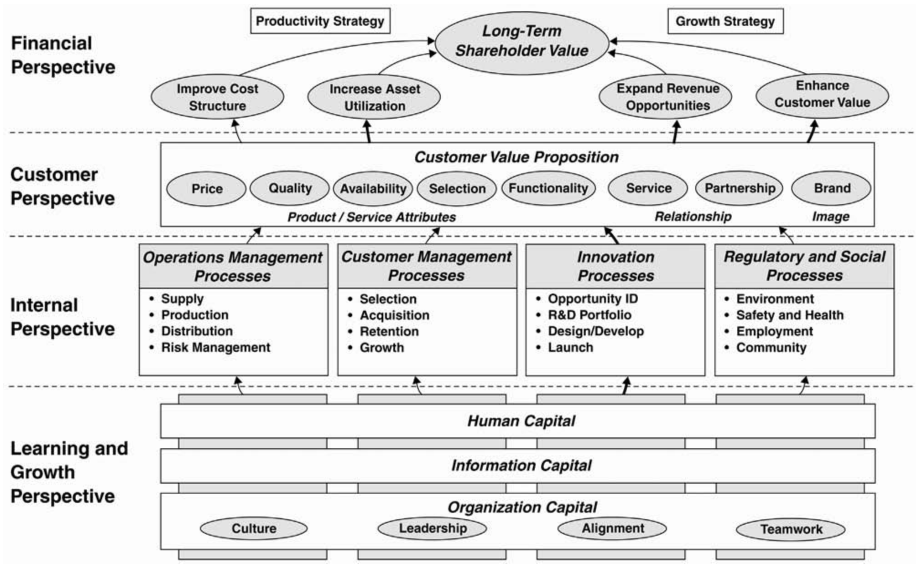
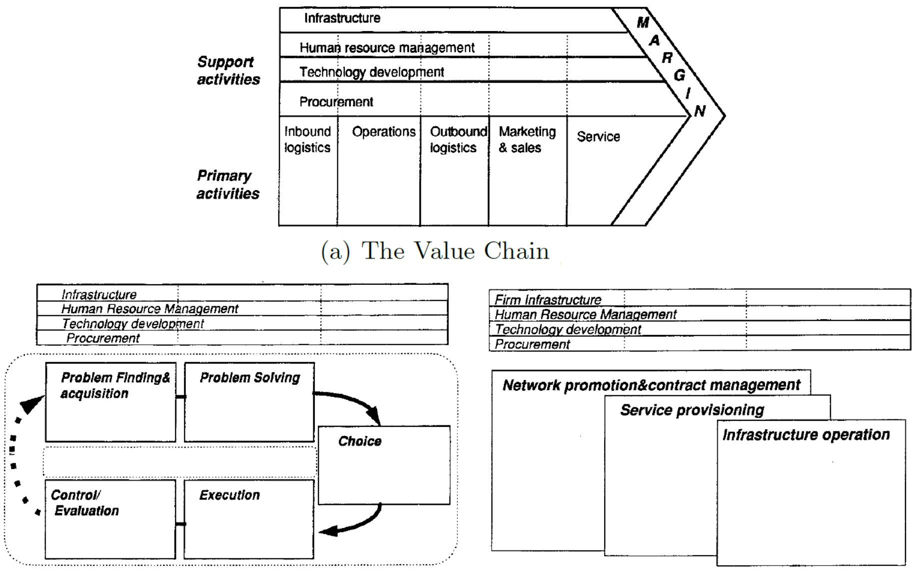


Fig. 1. The Strategy Map template [1]

Select Schemata Representation. Both business strategy formulations are conceptualized and represented as UML conceptual models accompanied with constraints expressed in statements [12, 13, 14].

Select Integration Process Strategy. There are four possible variations grouped into *binary*, for integrating two schemata at a time, and *n-ary*, for integrating *n* schemata at a time [24]. Binary strategies can be divided into *ladder*, when two schemata are integrated and another schema is integrated with the intermediate result, and *balanced* when schemata are divided into pairs and integrated symmetrically. For UBSMM a binary, ladder integration process is adapted aiming at progressive and gradual unification of business strategy formulations.

Assigning preferences is relevant mostly in n-ary integration strategies. However, for binary integration strategies is it efficient to consider preferences before choosing component schemata. For UBSMM, there are two pragmatic reasons why VC and SMBSC were preferred; a) to the best of the authors' knowledge no other business strategy formulations have been formalized, thus not available, and b) based on citations and literature search these two are well established in Strategic Management literature [10].



(a) The Value Chain

(b) The Value Shop

(c) The Value Network

Fig. 2. The Value Configuration [2, 11]

2.2 Schema Comparison

During this second phase schemata are analyzed for correspondences to be identified, and compared for conflicts and inter-schema properties to be discovered, as well as to collect other relevant information.

Both schemata are annotated with acronyms; for Strategy Maps and Balanced Scorecards, *SMBSC* is used as a prefix, while for the Value Configuration, *VC*. Classes are presented in the form of schema.class, attributes are presented in the form of schema.class.attribute, and associations are presented capitalized as they appear in the schemata. Due to space limitations, examples of schemata comparisons are provided while the complete list of correspondences can be found in [25].

For schema analysis, schemata have been analyzed, correspondences have been identified and then schemata have been compared for conflicts. Additionally, inter-schema properties have been discovered; semantic relationships holding between a set of objects in one schema and a different set of objects in another schema. Example correspondences include: *VC.Strategy* with *SMBSC.StrategyMap*, *VC.Strategy.Type* with *SMBSC.CustomerValueProposition, VC.Strategy.Goal* with *SMBSC.Goal,* and *VC.ValueActivity* with *SMBSC.Initiative*.

These classes were then compared for identifying naming conflicts, aiming at identifying synonyms and homonyms between the two schemata, and structural conflicts. The aforementioned correspondences revealed:

- a naming conflict between *VC.ValueActivity* and *SMBSC.Initiative* as they are synonyms due to the former capturing the distinct activities used in a VC, thus all activities that support the strategy and the latter capturing activities identified to be required towards the achievement of an objective, derived by a goal, thus all activities that support the strategy.

- a naming conflict between *VC.Strategy* and *SMBSC.StrategyMap as they* are homonyms due to both classes referring to a strategy, but with different meaning, using different names.
- a structural conflict between *VC.Strategy.Goal* and *SMBSC.Goal* as *VC.Strategy.Goal* is an attribute of *VC.Strategy* capturing the superior long-term return on investment generating real economic value and *SMBSC.Goal* is a class capturing all goals set across all four perspectives of SMBSC interrelated through causality relations, thus including goals of long-term shareholder value.
- an inter-schema property between *VC.Strategy.Type* and *SMBSC.CustomerValueProposition*, where the former captures three generic strategies: cost leadership, differentiation, and focus which reflects on the aggregation of *VC.ValueProposition* (*PriceRange*, *NeedType*, and *CustomerType*), and the latter captures four customer value proposition types within the customer perspective (low total cost, product leadership, complete customer solution, system lock-in) as a specialization of *SMBSC.Group*.

2.3 Conforming to Schemata

This phase of schema integration entails resolving the conflicts identified previously to align schemata for merging and restructuring. Therefore, semantic relationships between concepts involved in conflicts need to be identified as *identical*, *equivalent*, *compatible* and *incompatible* [24].

Concepts are considered *identical* when the same modeling constructs are used across schemata to represent the same concepts. *Equivalence* consists of three types: (i) *behavioral*; when corresponding instantiations of concepts can be queried and retrieved, (ii) *mapping*; when concept instances correspond one to one to each other, and (iii) *transformational*; when a concept is transformed to preserve equivalence with a correspondent concept. Concepts are *compatible* when they are neither identical nor equivalent and their modeling constructs, design principles and constraints are not contradicting each other's. Concepts are *incompatible* when their specification is contracting each other's [24].

Consequently for the examples presented in schema comparison semantic relationships were identified and resolutions have been provided:

- Synonyms *VC.ValueActivity* and *SMBSC.Initiative* are identical; therefore, the latter is renamed into *SMBSC.ValueActivity*.
- Homonyms *VC.Strategy* and *SMBSC.StrategyMap* can be transformed to preserve equivalence; therefore, they are renamed into *VC.StrategyPlan* and *SMBSC. StrategyPlan* respectively.
- For structural conflict between *VC.Strategy.Goal* and *SMBSC.Goal*, transformation can preserve the equivalence; therefore, attribute *VC.Strategy.Goal* becomes a class. *VC.Strategy Includes* exactly 1 *VC.Goal* and

VC.Goal BelongsTo exactly 1 *VC.Strategy*, which becomes homonym to *SMBSC.Goal* as they have different constraints. Consequently, they are both renamed to *VC.StrategicGoal* and *SMBSC.StrategicGoal*.

2.4 Schema Merging and Restructuring

The conformed schemata are merged and restructured to embed the inter-schema properties identified earlier through various types of operations, such as transformations that produce common generalizations, joins that produce common subtypes, aggregation, attribution creation, etc. [24].

The example of inter-schema property between *VC.Strategy.Type* and *SMBSC.CustomerValueProposition*, is addressed as follows:

- *UniqueValueProposition* is introduced (Figure 3), carrying and attribute with Type: *LowTotalCost*, *ProductLeadership*, *CompleteCustomerSolution*, and *SystemLock-In*. The three generic strategy types of VC correspond to the customer value proposition types in SMBSC which includes a forth. Price corresponds to low total cost, need corresponds to product leadership, customer corresponds to complete customer solution, and system lock-in is also added as used in SMBSC.
- *UniqueValueProposition* is associated with a 1..1 association to *StrategyPlan* for equivalence with *VC.Strategy.Type*.
- *PriceRange*, *NeedType* and *CustomerType* are parts of *UniqueValueProposition* through aggregation associations for equivalence with *VC.ValueProposition*.
- *UniqueValueProposition* is a specialization of *Group* allowing the representation of *StrategicGoal* as goal on the customer value proposition as subgroup of the customer perspective for equivalence with *SMBSC.CustomeValueProposition*.

The outcome of merging and restructuring of the two schemata is presented in figure 3. Due to space limitations, *StrategyPlan* and *StrategicGoal* class and constraint descriptions are presented, aligned with the correspondences presented in previously. A list of class and constraint descriptions can be found in [25].

Class *UBSMM.StrategyPlan* captures the strategy of an actor and carries a Type attribute, which indicates the business strategy formulation modeled as a list:

A *UBSMM.StrategyPlan* of *Type*: *StrategyMap*:

- *Includes* (exactly) one copy of each of the four predefined perspectives of the strategy map template.
- *Includes* at least one goal in each perspective, thus at least four goals.

A *UBSMM.StrategyPlan* of *Type*: *ValueConfiguration*:

- *Includes* (exactly) one copy of the three predefined primary activity groups, ValueChainPrimary, ValueNetworkPrimary, ValueShopPrimary in accordance to the *Type* of *ValueConfiguration IsBasedOn*.
- *Includes* exactly one goal which does not belong to any group.

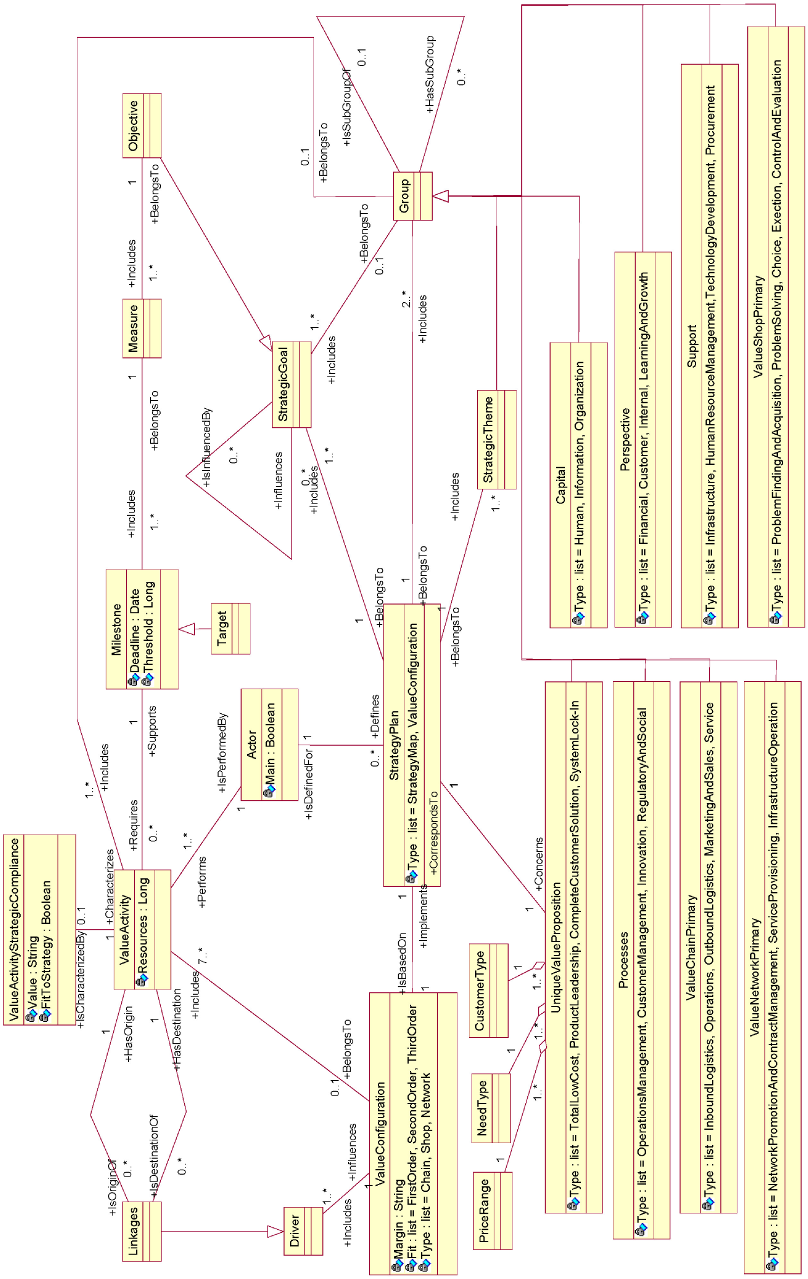


Fig. 3. The Unified Business Strategy Meta-model (UBSMM)

Class *UBSMM.StrategicGoal* captures goals set either across the four perspectives for *SMBSC* or the strategy overarching goal set in *VC* (usually: superior long-term return on investment). Causality relationships between *StrategicGoals* are captured through the self-association *Influences*, *IsInfluencedBy*.

A *UBSMM.StrategicGoal BelongsTo StrategyPlan* of *Type*: *StrategyMap*:

- included in a *StrategicTheme* is also included in the *StrategyPlan* to which the *StrategicTheme BelongsTo*.
- belonging to *Perspective* of *Type*:*Financial* which is a *Group* does not derive any *ValueActivity* because *Target* captures the results of *ValueActivity* from the other perspectives.
- belonging *Perspective* either of *Type*:*Financial* or *Type:Internal* which is a *Group* may influence another *StrategicGoal BelongsTo StrategyPlan* of *Type*: *StrategyMap* that *BelongsTo* either the same perspective or above.
- belonging to *Perspective* of *Type*:*LearningAndGrowth* which is a *Group* can only be *InfluencedBy* another *StrategicGoal BelongsTo StrategyPlan* of *Type*: *StrategyMap* that *BelongsTo* the same perspective (there exists no one below).
- belonging to P*erspective* of *Type*:*Financial* which is a *Group* can only *Influence* another *StrategicGoal BelongsTo StrategyPlan* of *Type*: *StrategyMap* that *BelongsTo* the same perspective (there exists no perspective above).
- must *Influence* another *StrategicGoal BelongsTo StrategyPlan* of *Type*: *StrategyMap*, except if it BelongsTo P*erspective* of *Type*:*Financial* which is a *Group* where a top-goal may exist.
- belonging to a *Group* must belong to the same *StrategyPlan* in which this *Group* belongs to.

3 Aligning UBSMM to Enterprise Architecture

Enterprise architecture provides holistically the methods, and models used to realize an enterprise's organizational structure, processes, information systems and infrastructure [15]. When addressing the alignment of business strategy to EA, UBSMM captures business strategy providing a common interface towards EA, where a common interface is desirable as well; for that, we consider the ISO/IEC/IEEE 42010-2011 standard [22], as explained in what follows.

3.1 Conceptual Relationships between UBSMM and Enterprise Architecture

ISO/IEC/IEEE 42010-2011 (figure 4) describes software system architectures through a set of generic concepts and terms of reference accepted for an architecture description, as well as a conceptual model of a system of interest [22].

Enterprise architecture frameworks such as TOGAF [18], etc. are aligned with the concepts of the architecture description model provided above. Therefore, when considering an enterprise as a system-of-interest, thus aiming at an architecture description of an enterprise, conceptual relationships between UBSMM and ISO/IEC/IEEE 42010-2011 models are identified thus allowing the consequent identification of an enterprise architecture description based on business strategy.

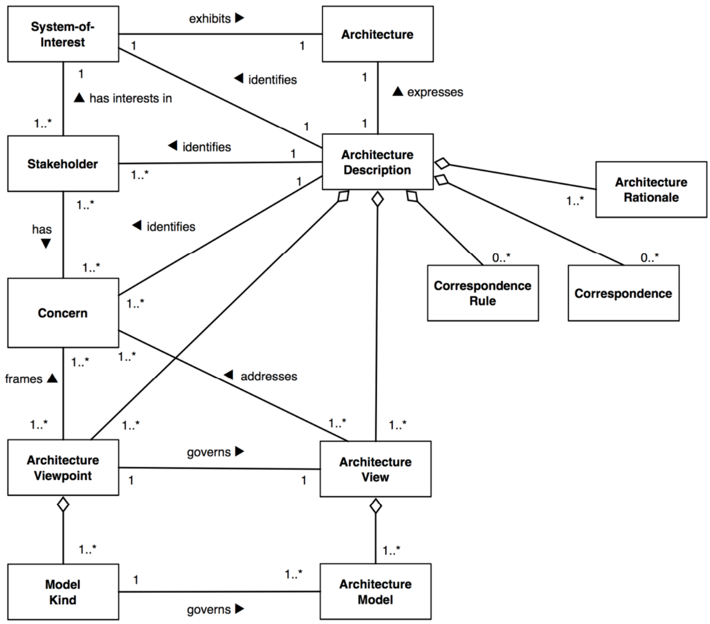


Fig. 4. The ISO/IEC/IEEE 42010-2011 Meta-model [22]

An *Architecture Viewpoint* (Table 1) frames an enterprise's concerns, which when considered holistically they constitute its strategic interests and thus constituting business strategy as its Architecture Viewpoint. Based on this proposal more conceptual relationships are identified:

- *Stakeholders*, represented by *Actor* in UBSMM, are all those having an interest in the long-term profitability and continuity of the enterprise, those that share its purpose of existence, thus its mission and vision.
- Business strategy as an *Architecture Viewpoint* frames stakeholders' concerns expressed as the generic strategies of an enterprise: *being low cost*, *being a product leader,* or *being focused* [2]. As such, business strategy governs particular business strategy formulation as Architecture Views.
- Business strategies such as VC and SMBSC are candidates as architecture views as they address stakeholders' concerns, the three generic strategies.
- *Model Kind* for business strategy being an architecture viewpoint is UBSMM with instantiations: UBSMM.SMBSC and UBSMM.VC, as Architecture Models.
- *Correspondences* between *Architecture Description* elements can be assessed to hold or violate the *Correspondence Rules* as defined by UBSMM through constraints to instantiate either SMBSC or VC.

- *Architecture Rationale* captures the justification for choosing SMBSC or VC as *Architecture View*, as well as the justification for instantiations of all relevant strategy concepts of UBSMM.

Table 1. The concepts of ISO/IEC/IEEE 42010-2011 [22] adjusted for an enterprise

ISO/IEC 42010	For an enterprise
System-of-Interest	*An enterprise.*
Stakeholder	*An individual/team/organization with an interest on the Enterprise ascribing purposes to it.*
Concern	*An interest in the enterprise relevant to one or more of its stakeholders.*
Architecture	*Fundamental properties of the enterprise in its environment embodied in its elements, relationships, and in the principles of its design and evolution. An enterprise is situated in an environment, where the environment determines the totality of influences upon the enterprise through its life cycle.*
Architecture Description (AD)	*A work product used to express architecture for an enterprise.*
Architecture Rationale	*Explanation, justification and reasoning for architecture decisions that should be recorded.*
Correspondence	*A relation between AD elements used to express, record, enforce and analyze consistency between AD elements for an enterprise identifying rules governing it.*
Correspondence Rule	*A rule enforcing the application of correspondences between AD elements of an enterprise.*
Architecture Viewpoint	*A work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific enterprise concerns.*
Model Kind	*Includes the languages, notations, conventions, modeling techniques, analytical methods and operations appropriate to the enterprise concerns framed by an architecture viewpoint.*
Architecture View	*A work product expressing the architecture of the enterprise from the perspective of specific enterprise concerns.*
Architecture Model	*A model adhering to a model kind appropriate for the enterprise concerns addressed by the architecture view*
Architecture Framework	*Conventions, principles, and practices for the description of an enterprise architecture established within a specific domain of application and/or community stakeholders.*

For distinct elements of UBSMM for VC and SMBSC different conceptual relationships are also identified for SMBSC (Table 2) and VC (Table 3).

Table 2. Relationships between ISO/IEC/IEEE 42010-2011 [22] and UBSMM.SMBSC

ISO/IEC 42010	**UBSMM.SMBSC**	**Description**
Concern	StrategicGoal	*Set across the four perspectives.*
Architecture Viewpoint	Perspective	*Four viewpoints grouping strategic goals /framing concerns. E.g. Customer Perspective.*
Architecture View	ValueActivity	*The set of ValueActivity derived (governed) from the StrategicGoal can be considered as corresponding views because they address StrategicGoal within a specific Perspective. e.g. for Customer it addresses the goals relevant to the UniqueValueProposition.*
Model Kind	Perspective:Type	*e.g. Perspective of Type:Customer provides modeling conventions appropriate for a UniqueValueProposition which includes the interplay of CustomerType, NeedType and PriceType.*

Table 3. Relationships between ISO/IEC/IEEE 42010-2011 [22] and UBSMM.VC

ISO/IEC 42010	**UBSMM.SMBSC**	**Description**
Architecture Viewpoint	ValueConfiguration	*e.g. Stakeholders in a manufacturing enterprise are concerned with the enterprise's value chain being unique to bring value.*
Architecture View	Primary/Support	*Groupings of primary and support activities (depending on the configuration type), which address the particular concerns of the stakeholders interested in an enterprise. E.g. ValueActivity is structured in the groups of ValueChainPrimary and Support.*
Model Kind	ValueConfiguration:Type	*Selection of type determines the model of strategy. e.g. ValueConfiguration:Type. Chain.*

3.2 Usage Scenarios

Alignment via a unified business strategy meta-model to enterprise architecture has a number of applications.

Different business strategies have different concerns. Given the number of business strategy formulations and enterprise architectures that may exist in a business context, a 1..1 mapping between strategy and enterprise architecture as presented in the previous section is desirable to avoid numerous pairs of mappings.

Different business strategies across enterprise units (such as for different local markets) may share a unique enterprise architecture, such as TOGAF [18], therefore; the use of UBSMM provides them benefits in the unification of strategy terms, maintenance, and compatibility; further, it becomes possible for every local enterprise unit to map directly their Strategy terms through the proposed conceptual relations to the terms used in the EA.

Communication and understanding between enterprises merging, or establishing partnerships, can be enhanced using a single business strategy meta-model like UBSMM when it comes to understanding and relating each other's' EA.

4 Related Work

The need for aligning EAs to business strategy has been argued in [20] stressing the need for thinking holistically. There exist proposals providing links between strategy and goals for information system (IS) architecture as part of a model-supported alignment framework between IS architecture and its surrounding organization [26]. But there also exist widely applied EAs in the industry. The Zachman Framework [16] includes the notion of business model aiming to capture what's important to the business through a set of three models focused on data, function and network, where strategies are perceived as means towards business objectives (ends). TOGAF [18] includes concepts of business strategy, technology strategy, business principles, objectives and drivers as part of the architecture vision as well as goals, objectives and measures as part of the business architecture. Archimate [19] includes a business layer with concepts addressing information, behavior, structure and motivation. Strategy constitutes a concern for the different viewpoints identified. Part of ARIS's [17] core layer focuses on strategy providing a strategic specification for process design, optimization, controlling and execution. GERAM [27] includes an entity concept that addresses mission, vision, strategy, objectives, etc.

While EAs do not overlook business concepts, they do not relate with business strategy formulations thus resulting into EA being agnostic to them. Moreover, EAs include methods, techniques, and tools used to design, model, develop, monitor, and maintain models and systems requiring concepts to be defined clearly. However, business strategy formulations are traditionally natural language-based, accompanied by schematic representations, where formalization is not seen as a priority.

Such difference also hinders the development of support tools for establishing linkages that can facilitate tracing actions, artifacts and decisions between business strategy and EA. To the best of the author's knowledge the only business strategy formulation formalization efforts that exist are the ones of OMG [28] focused on the balanced scorecards and the ones of [12, 13, 14] focused on SMBSC and VC.

5 Conclusion and Future Work

In this paper, UBSMM, a unified conceptualization of business strategy formulations is proposed, which is aimed to provide model-driven strategic awareness to EAs. UBSMM facilitates the alignment of different business strategies, or integration with strategies of others and can also serve as a pivot model between different business strategy formulations of a single or multiple enterprises.

EA is known to provide the methods and models to design and realize organizational structure, processes, and IS. However, alignment of EA with business strategy is an open issue. In this study UBSMM has been considered as an appropriate solution to provide strategic awareness to EA via model-to-model linkages. Using UBSMM to link business strategy with EA has a number of benefits:

- A strategic view on EA can be established; from UBSMM towards IS development.
- Simplified model mappings - given the number of business strategies and EAs that may exist in a business context, 1-1 mapping between strategy and enterprise architecture through UBSMM and a template EA model (i.e. ISO 42010) eliminates the need for establishing numerous pairs of mappings.
- Communication and understanding within a single enterprise with the units following different strategies, or between enterprises merging or establishing partnerships, can be enhanced using a single business strategy meta-model like UBSMM to easier understand and relate each other's' EA.

In the future, the conceptual relationships identified can be extended through mappings of UBSMM to distinct EAs exploring potential benefits via real cases.

References

1. Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston (2004)
2. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance. Free Press, New York (1985)
3. Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging information technology for transforming organizations. J. IBM Systems 32(1), 4–16 (1993)
4. Luftman, J.N., Papp, R., Brier, T.: Enablers and Inhibitors of Business-IT Alignment. J. Communications of the Association for Information Systems, 1, article 11 (1999)
5. Campbell, B.: Alignment: Resolving ambiguity within bounded choices: In: 9th Pacific Asia Conference on Information Systems (PACIS 2005), pp. 1—14 (2005)
6. Singh, S.N., Woo, C.: Investigating business-IT alignment through multi-disciplinary goal concepts. J. Requirements Engineering 14, 177–207 (2009)
7. Gordijn, J., Petit, M., Wieringa, R.: Understanding Business strategies of networked value constellations using goal- and Value modeling. In: 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 129–138. IEEE Press, New York (2006)
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. Bleistein, S.J., Cox, K., Verner, J.: Validating Strategic Alignment of Organizational IT Requirements using Goal Modeling and Problem Diagrams. J. Systems and Software 79, 362–378 (2006)
10. Giannoulis, C.: Modeling Business Strategy For Business-IT Alignment. Licentiate Thesis, Department of Computer and Systems Sciences, Stockholm University (2011)
11. Stabell, C.B., Fjeldstad, Ø.D.: Configuring Value for Competitive Advantage: On Chains, Shops, and Networks. J. Strategic Management 19(5), 413–437 (1998)
12. Giannoulis C., Petit, M., Zdravkovic, J. Modeling Business Strategy: A Meta-model of Strategy Maps and Balance Scorecards. In: 5th IEEE International Conference on Research Challenges in Information Science (RCIS2011). IEEE, New York (2011)

13. 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)
14. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling Competition-driven Business Strategy for Business IT Alignment. In: 6th International Workshop on Business/IT Alignment and Interoperability (BUSITAL 2011), pp. 16–28. Springer, Heidelberg (2011)
15. Lankhorst, M.: Enterprise Architecture at Work; Modeling, Communication and Analysis. Springer, Heidelberg (2009)
16. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26(3), 276–292 (1987)
17. Scheer, A.-W.: ARIS-Business Process Frameworks. Springer, Berlin (1999)
18. The Open Group: The Open Group Architectural Framework (TOGAF) Version 9. The Open Group, Reading, UK (2009), http://www.opengroup.org/togaf/
19. The Open Group: Archimate 2.0 Specification. The Open Group, Reading, UK (2012), http://theopengroup.org/archimate/
20. Whittle, R., Myrick, C.B.: Enterprise Business Architecture: The Formal Link between Strategy and Results. CRC Press (2004)
21. Burton, B., Allega, P.: Enterprise Architects: Know Thy Business Strategy, Gartner Insight. Technical report (2011)
22. ISO/IEC/IEEE 42010-2011: Systems and Software Engineering – Architecture Description. International Organization for Standardization, Geneva, Switzerland (2011)
23. Boman, M., Bubenko Jr., J.A., Johannesson, P., Wangler, B.: Conceptual Modelling. Prentice Hall, Upper Saddle River (1997)
24. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. J. ACM Computing Surveys 18(4), 323–364 (1986)
25. Giannoulis, C.: Schema Integration Process for UBSMM. Technical Report (2011), http://people.dsv.su.se/~cgia/docs/TowardsUBSMM.pdf
26. Opdahl, A.L.: Model-supported Alignment of Information Systems Architecture. In: Business Strategies for Information Technology Management. IGI Publishing (2003)
27. Bernus, P., Nemes, L., Schmidt, G.: GERAM: The Generalized Enterprise Reference Architecture and Methodology. In: Handbook on Enterprise Architecture. Springer (2003)
28. Business Architecture Working Group: Balanced Scorecard Metamodel. The Object Management Group (OMG) (2010), http://www.omgwiki.org/bawg/doku.php?id=bscmetacurrent (last accessed 2011.12.04)

On Transforming DEMO Models to ArchiMate*

Sybren de Kinderen[1], Khaled Gaaloul[1], and H.A. (Erik) Proper[1,2]

[1] PRC Henri Tudor, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, The Netherlands
{sybren.dekinderen,khaled.gaaloul,erik.proper}@tudor.lu

Abstract. ArchiMate is an established enterprise architecture modelling language that allows organizations to be modelled from a holistic perspective. As a result, its modelling constructs are coarse grained by design and architects may feel that they do not get enough guidance from the language in producing ArchiMate models. To address this issue, we suggest using methods with a more refined semantics and elaborated modelling guidance, as a 'front-end' to ArchiMate. In this paper, we will show how the DEMO method can indeed be used as a front-end to ArchiMate, where we will focus on the automatic transformation of DEMO models to ArchiMate models. This is done by creating a formal link between DEMO and ArchiMate with additional benefits of linking DEMO's key modelling concept of transactions, being as (socio-economic) commitments between actors, to ArchiMate. Specifically, we provide a formal approach that can be used to transform DEMO models into ArchiMate models. In addition, we provide a software implementation of our approach which is illustrated by means of an illustrative case study from the insurance domain.

Keywords: ArchiMate, DEMO, meta-model, model transformation.

1 Introduction

ArchiMate, is an Open Group standard [1, 2] for the modelling of enterprise architectures[1]. Being designed as a general purpose modelling language for enterprise architecture [3, 4], it allows architects to model an enterprise from a holistic perspective, showing amongst others, an organization's products and services, how these products and services are realized/delivered by business processes, and how in turn these processes are supported by information systems and their underlying IT infrastructure. This holistic perspective on an enterprise helps to guide change processes [5], provides insight into cost structures [6], and more [1].

Because of the inherent holistic nature, ArchiMate lacks specificity on how to model the different perspectives *in-depth*. For example, ArchiMate lacks guidelines for process modelling, and lacks expressivity for modelling an enterprise

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

[1] http://www.opengroup.org/archimate/

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 270–284, 2012.

from a value exchange perspective [7]. Moreover, as claimed by [8] ArchiMate lacks conceptual clarity and precision. This “lack” is, however, a direct consequence of the coarse-grained, and holistic, nature of ArchiMate. In that sense this *freedom of interpretation* has been designed into the language on purpose [3, 4]. Nevertheless, as a result, different modellers do indeed create different models. To address the above issues, it has already been suggested that ArchiMate could benefit from the integration with a method such as DEMO to provide it with a more explicit way of working, supporting architects in the creation of models [8]. In this paper, we focus on bridging between DEMO and ArchiMate.

DEMO, short for Design and Engineering Methodology for Organizations, is a method comprising of a comprehensive set of conceptual modelling techniques, in combination with a theory based a way of thinking and associated way of working, focused on modelling/analysing/designing the *essential* aspects of an organization [9, 10]. DEMO uses the word *essential* here to refer to the implementation-independent aspects of an organization. As such, DEMO aims to abstracts away from implementation-specific details, such as the information systems present in a business collaboration. Linking DEMO and ArchiMate would enable architects to use the semantically rich way of thinking of DEMO to create ArchiMate models. These models would then primarily be ArchiMate models providing an essential view of the business processes (business layer) and the information processing (application layer) in the enterprise.

While both DEMO and ArchiMate are the result of academic research, they both have a clear impact in practice. ArchiMate has evolved to become The Open Group's standard for enterprise architecture description [2]. DEMO has not yet evolved to become an internationally accepted standard. However, it has indeed proven its use in practice, by amongst others, (1) aiding in the standardisation of message exchange formats in construction sector [11], (2) acting, in a government agency, as a point of departure for business process modelling [12], and, by (3) fostering, in the aerospace industry, a shared understanding of fragmented strategic concerns, and a link of these strategic concerns to design principles [13].

Next to the fact that a formal linkage of DEMO and ArchiMate would enable architects to produce better ArchiMate models, linking DEMO and ArchiMate formally will also have a second added benefit. As argued in e.g. [14], at the level of an enterprise architecture it is desirable to trace the connection between models at the level of value exchange between the enterprise and its environment, via transactions and services, to its internal processes and implementation. In [14], an illustrative case study showed the added value of being able to trace the connection from value exchanges, via economic transactions and their underlying business services, to the business processes and supporting information systems. ArchiMate on its own only supports part of this chain of models. As already suggested in [14], this would require an integration/linkage of languages such as e^3*value* [7], DEMO and ArchiMate. e^3*value* focuses on modelling an enterprise from a value perspective. It depicts the value exchanges between actors participating in a value web, describing what each actor offers to others, and what it receives in return. So for example, an online music store ships an ‘LP’ to the customer, and receives 'Money' as compensation. We see the work reported on in this paper, also as a first step in more closely integrating the modelling chain from value modelling (e^3value), via transaction modelling (DEMO), to business services and realization (DEMO and/or ArchiMate) and implementation modelling (ArchiMate).

The core contribution of this paper is two-fold: (1) a formal mapping of the DEMO and ArchiMate, specifically of the meta-models underlying these techniques, and (2) a systematic application of these meta-models to map a model created in DEMO to a model of an enterprise architecture in ArchiMate. We use a running example of an insurance scenario to illustrate our ideas. Moreover, we discuss a software tool, based upon our proposed meta-models, that interfaces between two existing software environments for modelling with, respectively, DEMO and ArchiMate. This interfacing software tool provides for partial computational validation of the proposed model transformation.

The remainder of this paper is structured as follows. Sect. 2 illustrates, by means of the case of an insurance company, how we intend to use DEMO as a front-end to ArchiMate. Therafter, we show how to formally transform a DEMO model into an ArchiMate model (Sect. 3). Before concluding, we discuss our implementation of this transformation in terms of a software tool (Sect. 4).

2 Modelling Insurance Transactions and Processes in DEMO

In this section, we create a model according to the DEMO meta-model and its accompanying tools, so that we have a solid basis for creating an Enterprise Architecture model in ArchiMate (which we discuss in Sect. 3). First, in Sect 2.1, we introduce the running insurance case that will be used for illustrating our application of DEMO and, later on, the transformation into ArchiMate. Thereafter, we apply DEMO to our running case. First, we create a high-level DEMO model that focuses on organizational transactions (in Sect. 2.2). Following this, we detail the business processes realising these organizational transactions (in Sect. 2.3).

2.1 ArchiSurance: Selling Car Insurance via Insurance Brokers

For illustration purposes, we use an elaboration of the ArchiSurance case study which as also been used to illustrate the ArchiMate modelling language [5, 15]. The ArchiSurance case study has been accepted in The Open Group as being a realistic case study for providing enterprise architecture example scenarios [16]. In this paper, we extend the ArchiSurance case with area of insurance intermediaries [17] as it allows us to better illustrate the added value of linking DEMO and ArchiMate.

For this paper we actually focus on car insurance, an insurance product that ('our') ArchiSurance sells via *insurance brokers*. The main reason for selling insurance via brokers is to reduce the risk of *adverse risk profiles* [17], incomplete or faulty risk profiles of customers that lead insurance companies to sell inappropriate insurance packages. To mitigate adverse risk profiles, insurance companies may therefore rely on insurance brokers, whose core business it is to match customer profiles to appropriate packages.

2.2 The DEMO′ Meta-model and Transaction Model

We use DEMO to model the sale of car insurance by ArchiSurance. As stated, DEMO aims at modelling the essential, implementation-independent aspects of an organization only. DEMO achieves its focus on the essential aspects by perceiving of an organization as a *social* system of actors, that collaborate to achieve a common goal. Chief to this collaboration are acts: production acts, and communication acts. Production acts bring about (part of) a good or service, and directly contribute to achieving the organization's common goal. In the ArchiSurance case, a production act is for example 'Find matching insurance package', as executed by the insurance broker on behalf of the customer. Communication acts, then, serve to coordinate among the actors that either receive results from, or execute, the production acts. In the ArchiSurance case, 'Apply for insurance' is for example a communicative act used by the customer to indicate to the insurance broker the interest in an insurance package.

In this paper, we use only a subset of the DEMO conceptualisation and techniques, referred to as DEMO', or DEMO derived. DEMO′ borrows from DEMO a subset of concepts, to which we refer as the DEMO′ meta-model, and the DEMO standard transaction pattern.

The DEMO′ meta-model is depicted in Fig. 3. For the ArchiSurance case, Fig. 1 presents an instantiation of this meta-model, referred to as a DEMO′ transaction model. Here, we see the high-level transactions that together implement the selling of car insurance via an insurance broker. First, the transaction 'create customized insurance package', whereby the broker matches a customer profile to a fitting insurance package and, second, the transaction 'contracting', whereby the car insurance department within ArchiSurance - based on the risk profile that it receives from the broker - underwrites the insurance. This (underwriting) means that the car insurance department creates an insurance package for the customer, and calculates the associated premium. These transactions are carried out by physical subjects (e.g. the 'Car insurance department') that carry out an organizational role (e.g. the 'Car insurance department' fulfils the role 'underwriter').

Note however, that a DEMO′ transaction model does not show in detail the business processes that realize the modelled transactions.

2.3 From an ArchiSurance Transaction Model to a Process Model

To elucidate the business processes underlying the DEMO′ transactions, DEMO′ borrows from DEMO transaction patterns. The DEMO′ standard transaction pattern focuses on a process-based pattern of (instantiations of) DEMO′ meta-model concepts, showing the sequence of acts that *always* needs to be executed to realize an economic transaction. So, here we see again DEMO's emphasis

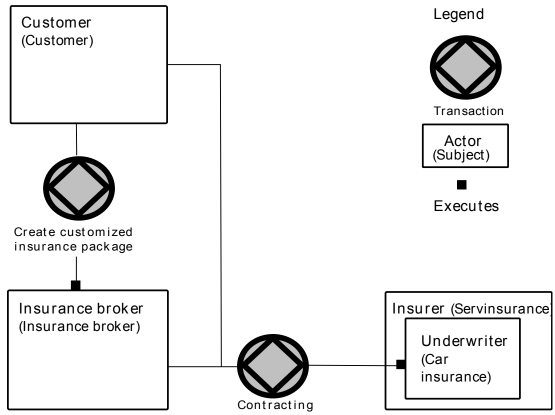


Fig. 1. DEMO transaction model of ArchiSurance

on the essential aspect of an organization: no matter what the domain, if we perceive of an organization as a social entity, then we see a pattern of generic acts that *always* occurs in carrying out a transaction [10]. So, for example, one actor always has to initiate a transaction by performing the act 'request' (which in the ArchiSurance case may translate to the act 'Apply for insurance' as carried out by a customer), while another actor has to always perform the 'execute' act in order to produce the good or service that the initiating actor is interested in (in the ArchiSurance case, this may translate to the act 'Find matching package' which, as mentioned before, is executed by the insurance broker).

Fig. 2 shows an instantiation of the DEMO′ standard transaction pattern for the ArchiSurance case. Note here in particular that the illocutionary acts from the transaction patterns, such as request, promise and accept, aid in detailing exactly what business process steps- or: DEMO′ acts - together realize the transactions in ArchiSurance's transaction model (see Fig. 1). For example, for the ArchiSurance case, the act 'request' translates to the act 'Apply for insurance', an act carried out by the customer to trigger the insurance process, while the act 'execute' translates to the act 'find matching package' as carried out by the insurance broker.

3 Translating DEMO′ Process Models to ArchiMate

In this section, we introduce the ArchiMate modelling language and focus on its business layer meta-model. Thereafter, we present the mapping between the DEMO′ and the ArchiMate business layer meta-models. Subsequently, we apply this mapping to transform the DEMO′ process model of ArchiSurance to an ArchiMate model.

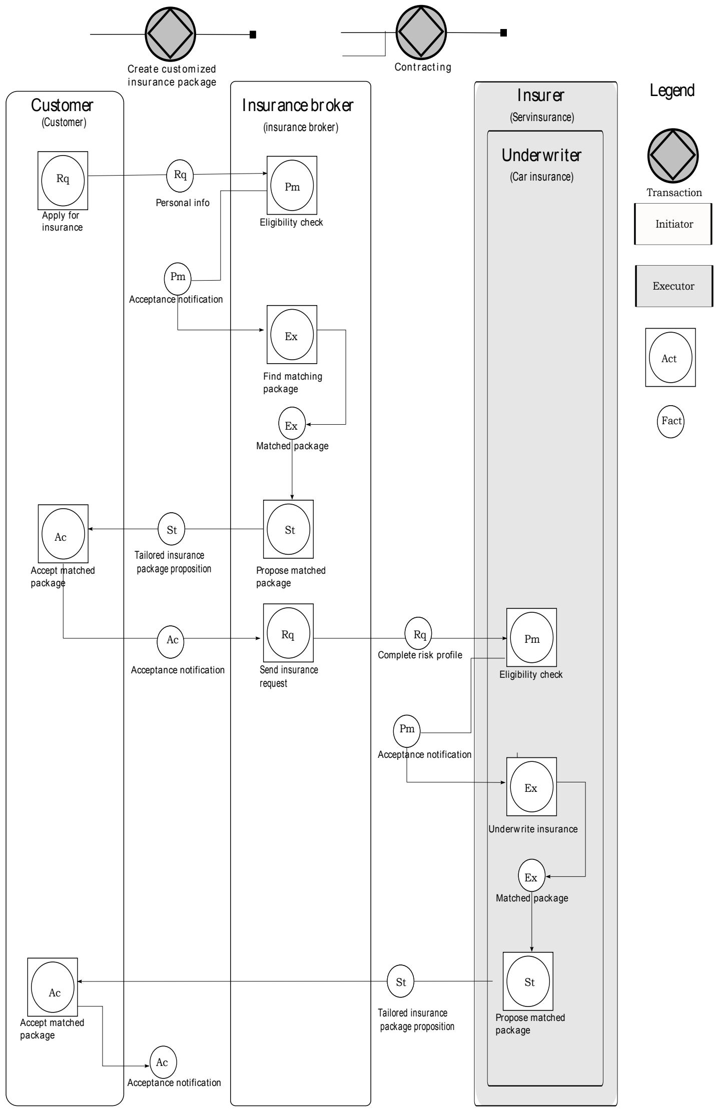


Fig. 2. DEMO Business process model, detailing the ArchiSurance transaction model

3.1 The ArchiMate Business Layer Meta-model

We rely on the ArchiMate modelling language to model the enterprise architecture of the ArchiSurance case. ArchiMate has been transferred to the Open Group, where it is slated to become the standard for architectural description accompanying the Open Group's architecture framework TOGAF [18]. As stated, ArchiMate offers a coherent, holistic, description of the enterprise architecture to enable communication among stakeholders, and to guide change processes within ArchiSurance.

We identify the main concepts for architectural descriptions that can be placed in the business layer of the ArchiMate meta-model. Fig. 3 gives an excerpt (ArchiMate′) of the business layer concepts and their relationships. The business layer refers to the static structure of an organization, in terms of the entities that make up the organization and their relationships [5].

3.2 Mapping the DEMO′ Meta-model to the ArchiMate′ Meta-model

For mapping DEMO′ to ArchiMate, we use the meta-model mapping technique described in [19] where authors distinguished different types of mappings, the most relevant for our work being (1) *class-to-class mappings*, which relates a concept from meta-model A to a concept from meta-model B (e.g., a 'Subject' from DEMO′ relates to an 'Actor' from ArchiMate′). And (2) *relation-to-relation mappings*, which relates concept relationships from meta-model A with concept relationships from meta-model B (e.g., 'performs_role' between the concepts Subject and Actor from DEMO′ relates to the ArchiMate relation 'assigned_to' between the concepts Actor and Business role).

[19] also distinguishes between different types of relations, the most important for us being: equivalence, generalisation of, and its inverse are specialisation of, and no relation.

Now that we have explained the main ideas behind ArchiMate and presented an excerpt of the business layer meta-model, we translate a DEMO process model for ArchiSurance into an ArchiMate business layer model. We do this in two main steps: (1) Translate the concepts from a DEMO process model to an ArchiMate process model, which we can do given that, looking at the concept definitions, the holistic ArchiMate language subsumes DEMO's social perspective, (2) Define a (partial) enterprise architecture model from a business perspective that focuses on the DEMO′ process model. Here, we construct an ArchiMate model from the mapped DEMO′ concepts. As we now actually construct an ArchiMate model, we take here into consideration (a) the difference in abstraction level between DEMO and ArchiMate, and (b) additional ArchiMate constructs not present in DEMO', for example for depicting an IT perspective on the organization at hand.

Step 1: Horizontal Integration via Meta-model Mapping. The first step will apply our mapping between the DEMO′ meta-model and the ArchiMate' business layer meta-model. Here, we make a mapping on a purely *horizontal* level (cf. [19]), meaning that we consider only differences between aspects modelled in DEMO and ArchiMate *on the same abstraction level*. In doing so, we apply the DEMO′ - ArchiMate' meta-model mapping from Fig. 3, and the corresponding rationale of our meta-model mapping (i.e., Table 1). In Fig. 3, we define a specialisation relation between the mapped concepts from DEMO′ to ArchiMate′ concepts. Here, we assume - based on the concept definitions from DEMO and ArchiMate - that the holistic ArchiMate language encompasses the specific social perspective emphasised by DEMO'.

For ArchiSurance, we apply this mapping as follows. For reference, see the ArchiSurance ArchiMate model in Fig. 4, and the ArchiSurance DEMO′ process model in Fig. 2.

- *Subjects from DEMO′ map to business actors in ArchiMate'.* We define a mapping relation from DEMO′ to ArchiMate′ concepts where a subject performing a role in DEMO′ is an actor in ArchiMate'. For instance, we map the subject 'car insurance' (a department within ArchiSurance specialised in car insurances) to the business actor 'car insurance' in ArchiMate'.
- *Actors in DEMO′ map to business role in ArchiMate'.* An actor performing an act in DEMO′ is the associated role to a business actor in ArchiMate'. For instance, we map the actor 'Underwriter', an associated role to the subject 'car insurance', to the role 'Underwriter' in ArchiMate'.
- *An act from DEMO′ is mapped to a set of business behaviour/events in Archimate'.* An act performed by an actor in DEMO′ maps to a business process step in ArchiMate'. For instance, we map the act 'Find matching package' to the same business step in ArchiMate'.
- *Transactions in DEMO′ map to business interactions in Archimate'.* A transaction is a collection of acts in DEMO′ and a business interaction includes a set of business steps performed within collaboration in ArchiMate'. For instance, we map the transaction 'Create customized insurance package' to the business interaction in ArchiMate'.

In addition, we perform relation-to-relation mapping between DEMO′ and ArchiMate (see Table 2). As such, we relate:

- *The relation (Subject) performs_role (Actor) from DEMO′ to the relation (Business actor) assigned_to (Business role) from ArchiMate.* For example, in both DEMO′ and ArchiMate, the department 'Car insurance' performs the role of 'Underwriter'.
- *The relation (Transaction) consists_of (Act) from DEMO′ to the relation (Business collaboration) triggers (Business event/business behaviour) from ArchiMate.* For instance, both in DEMO and ArchiMate, 'Create customised insurance package' consists of the more elementary acts 'apply for insurance' and 'find matching package'.
- *The relation (Actor) performs (Act) from DEMO′ to the relation (Role) assigned_to (Business event/business behaviour) from ArchiMate.* For example, in both DEMO′ and ArchiMate, the underwriter carries out the act 'Underwrite insurance'.

Step 2: Vertical Integration: Defining an Appropriate Abstraction Level in ArchiMate. The second step consists of defining an enterprise architecture model using ArchiMate to represent a DEMO′ process model.

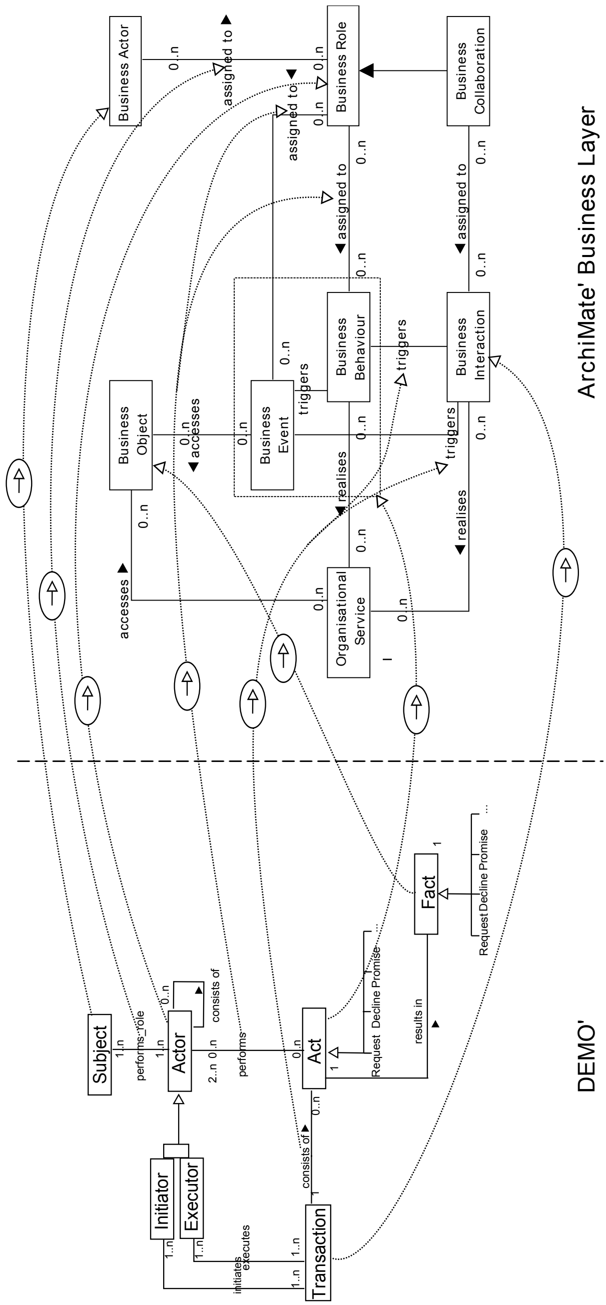


Fig. 3. Mapping of DEMO′ and ArchiMate′ meta-models

Table 1. DEMO′ - ArchiMate′ meta-model concept mapping relations

DEMO′ concepts	ArchiMate′ concepts	Mapping rationale
Actor	Business role	In DEMO, an actor refers to a social role played by a subject in an organization. Such a social role corresponds to the definition of a business role in ArchiMate where roles are typically used to distinguish responsibilities.
Subject	Business actor	A DEMO subject is an organizational entity - person, department or otherwise - that can fulfil an organizational role. This corresponds to a business actor in ArchiMate, which is an organizational entity that performs some behaviour (cf. [5]), thus it can also fulfil a role.
Act	Business behaviour/event	An act is performed by a subject in a social role. Its scope is about contribution/coordination for services. In the ArchiMate context, it corresponds to the realization of an organizational service via a business process or a function (business behaviour) or a business event (e.g., an external request).
Transaction	Business interaction	For DEMO transactions, the initiation and execution are performed by different actors. This emphasises the interaction aspect that we can find in ArchiMate, where a business interaction is carried out by more than one actor.
Fact	Business object	A fact is any object that results from performing an act. In ArchiMate', this corresponds to a business object, which 'represent the important concepts in which the business thinks about a domain' [5].

Table 2. DEMO′ - ArchiMate′ meta-model relations mapping

DEMO′ relation	ArchiMate′ relation	Mapping rationale
performs_role	assigned_to	In both DEMO and ArchiMate, one relates a real world entity (e.g., ArchiSurance) to a role played by that entity (e.g., the role of insurer in the case of ArchiSurance).
consists_of	triggers	As transactions map to business interactions, and acts map to business events and business behaviour, the relation 'consists_of' between transactions and acts in DEMO maps logically to the relation 'triggers' between business interactions and business events/business behaviour in ArchiMate.
performs	assigned_to	While both use different nomenclature, in both DEMO and ArchiMate, a role - not the real-world entity behind it - carries out acts.

First, in addition to the horizontal differences in Step 1, we now consider also the *vertical* differences between DEMO′ and ArchiMate'. This means that we remove from the ArchiMate model any elements that are too detailed for depicting a holistic perspective on the organization at hand. For example: for the ArchiSurance case, we thus remove the business objects 'acceptance notification' (which ArchiMate inherits from the DEMO process model in Fig. 2), since they are too detailed for the high-level model overview provided by ArchiMate.

Second, we supplement the model elements inherited from DEMO′ with ArchiMate constructs. This we do to fully express a holistic perspective on the organization at hand, most prominently in terms of the supporting IT infrastructure. For example, as we can see in Fig. 4, for ArchiSurance we model that the business process activities 'eligibility check' and 'underwrite insurance' are supported by a risk assessment application, and that both the business collaboration 'create customized insurance package' is supported by administrative applications from both the insurance broker and ArchiSurance.

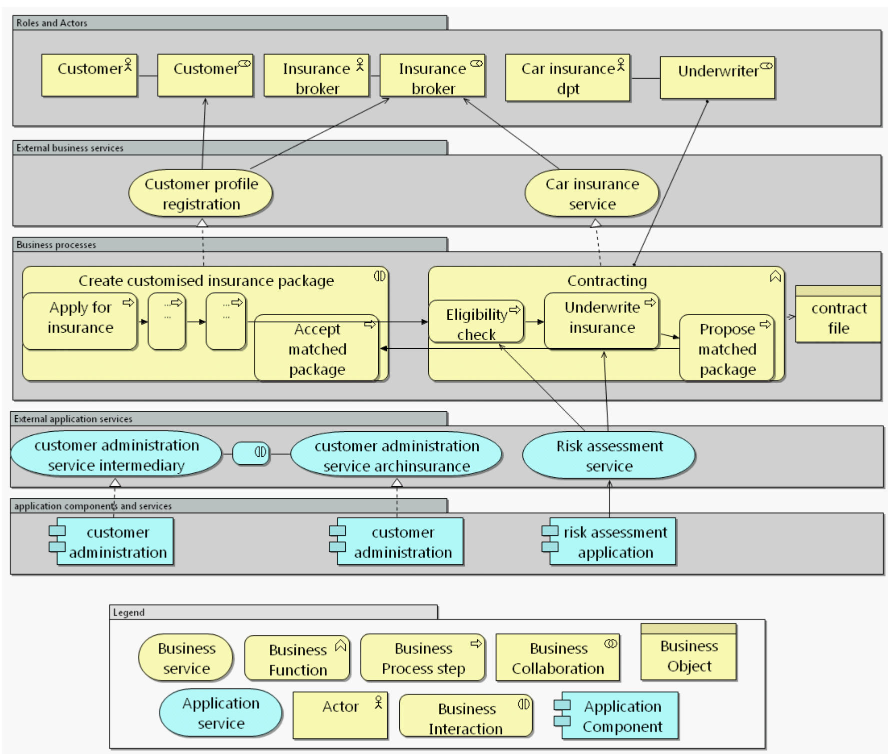


Fig. 4. (Partial) enterprise architecture model based on DEMO process

4 Tool Implementation

We have implemented our DEMO to ArchiMate mapping in ATL[2], an Integrated Development Environment for implementing model transformations that is built on top of the eclipse platform. This mapping conforms exactly to the mapping defined in Fig. 3: no concepts are added, modified, or removed. Table 3 shows a sample of the XML instantiations for the ArchiSurance case, in DEMO (ex-ante model transformation) and ArchiMate (ex-post model transformation). ATL can produce an ArchiMate instantiation in XML, given: (1) the DEMO and ArchiMate meta-models, defined in an ECORE syntax; (2) a meta-model mapping, and (3) an instantiation of the DEMO meta-model, defined in XML.

Table 3. Sample of the XML instantiations for the ArchiSurance

DEMO′ instantation	ArchiMate instantation
<BusinessActor name="Insurance Broker"> <assigned_to name="Insurance Broker Role"/> </BusinessActor> <BusinessActor name="Customer"> <assigned_to name="Customer Role"/>	<Subject name="Insurance Broker"> <performs_role name="Insurance Broker Role"/> </Subject> <Subject name="Customer"> <performs_role name="Customer Role"/> </Subject>

Currently the tool implementation is a proof-of-concept, showing that the mapping that we presented in this paper can indeed by implemented into software. Thus, we showcase the *possibility* for creating a formal tooling chain, whereby models created in a DEMO software tool can be exported to a format interpretable for an ArchiMate tool. However, the actual implementation of such a chain is outside of the scope of this paper.

5 Related Work

The e^3*alignment* approach provides tools for actually creating business-ICT alignment. It does so by ensuring that conceptual models depicting a strategic, value, process and ICT perspectives respectively on the value web at hand are consistent with one another [7]. However, this approach works only on a syntactic level. For instance, if the concept of an actor in e^3*value* and the concept of a swim lane in an UML activity diagram means the same is not a consideration. Derzsi et al. enable profitability calculations of an ICT-infrastructure by providing a meta-model that links an IT infrastructure modelled in UML to e^3*value* [20]. This approach has more formality than e^3*alignment* , yet it focuses on a link between IT and value only. As a result, business processes are not a consideration while these are realistically cost carriers as well.

2 http://www.eclipse.org/atl/

The Business Process Modelling Notation (BPMN) is a standardised business process notation which is defined and specified by the Object Management Group (OMG) [21] and has become the de facto standard for graphical process modelling. BPMN process models are composed of flow objects such as routing gateways, events, and activity nodes. Activities, commonly referred to as tasks, represent items of work performed by software systems or humans. Activities are assigned to pools and lanes expressing organizational institutions, roles and role hierarchies. Moreover, BPMN supports transactions that define a set of activities that logically belong together. However, BPMN is just a language: it does not provide any rules for business process modelling. This is opposed to DEMO, which provides standard transaction patterns that guide modelling.

Ontological merging approaches address the semi-automated integration of system models [22]. System models are created in terms of modelling language, which in itself is based on a meta-model. Syntactic and semantic mapping between pairs of meta-models has been facilitated by the application of existing approaches for ontology mapping [23, 24]. Ontologies improve not only the semantics of a meta-model but also provide a potential way in which these meta-models can be bridged with each other to be integrated within a common context [25]. However, ontology mapping approaches such as [23, 24] focus on providing an approximation of a mapping between two ontologies. Yet, in our research we require a precise mapping. Since our starting point are ontologies, such as DEMO with relatively few concepts (compared to larger ones such as found in the medical domain), it seems better to perform mapping/integration manually and as such, avoid an approximation of a mapping.

6 Discussions and Conclusion

In this paper, we used DEMO as a front end for ArchiMate. Using a case from the insurance domain, we introduced a mapping between DEMO and ArchiMate, and showed how this mapping can be applied to translate a DEMO model into an ArchiMate model. Also, we showed why such a mapping makes sense, in particular by using the transaction patterns from DEMO for constructing a business process that is later transformed into ArchiMate. Finally, we discussed an implementation of our mapping in the model transformation language ATL.

Having this transformational bridge in place, allows architects to start modelling the essential aspects of an enterprise first in DEMO, while then switching to the transformed ArchiMate models to then add more realization and implementation details. This formal linkage also opens up the opportunity to more explicitly link value models (e^3value) of an enterprise's position in a value network, via models of the associated socio-economic transactions it might engage in (DEMO), to business services and realization (DEMO and/or ArchiMate) and implementation modelling (ArchiMate). We see the work reported on in this paper also as a first step in creating such a modelling chain.

Furthermore, our efforts have mainly focussed on the ontological layer of DEMO. In future research, we will also consider the data- and infological layers of DEMO, more specifically how these compare to both the application and technology layers of ArchiMate.

In addition, while our case study suffices for illustrating the potential benefits of our chaining of modelling techniques, it is at the same time *fictituous*. To practically validate our approach, we will therefore also apply it in a real-life setting, while discussing its practical impact on enterprises transformation. Moreover, we will investigate the maturity of such integration and assess the quality of the model transformation.

Finally, we presented a model transformation between DEMO and ArchiMate only, leaving the DEMO and ArchiMate meta-models themselves untouched. We may refer to this as a *federated* approach towards model integration. However, one can also consider merging concepts from the DEMO meta-model (and other meta-models, such as that of e^3*value*) into the ArchiMate meta-model, thus creating a *unified architecture modelling* approach. Such a unified model is then interesting because it enhances the expressivity of ArchiMate with DEMO (and other) constructs. Yet, at the same time, fully merging DEMO (and other techniques) into ArchiMate would make ArchiMate rather 'top heavy', thus making ArchiMate lose one of its main advantages: that of providing a concise, A3-sized, holistic overview of an enterprise. So for future research, we will also experiment with balancing between a federated and unified model approach towards model integration.

References

1. Lankhorst, M.M. (ed.): Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin (2005) ISBN-10: 3540243712
2. Iacob, M.-E., Jonkers, H., Lankhorst, M.M., Proper, H.A.: ArchiMate 1.0 Specification. The Open Group (2009) ISBN-13: 9789087535025
3. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The Architecture of the ArchiMate Language. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 367–380. Springer, Heidelberg (2009) ISBN-13: 9783642018619, doi:10.1007/978-3-642-01862-6_30
4. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The anatomy of the archimate language. International Journal of Information System Modeling and Design (IJISMD) 1(1), 1–32 (2010), doi:10.1007/978-3-642-01862-6_30
5. Jonkers, H., Lankhorst, M.M., van Buuren, R., Hoppenbrouwers, S.J.B.A., Bonsangue, M., Van der Torre, L.: Concepts for Modeling Enterprise Architectures. International Journal of Cooperative Information Systems 13(3), 257–288 (2004)
6. van Buuren, R., Gordijn, J., Janssen, W.: Business case modelling for e-services. In: 18 th Bled eConference eIntegration in Action (2005)
7. Pijpers, V., Gordijn, J., Akkermans, H.: e3alignment: Exploring inter-organizational alignment in networked value constellations. International Journal of Computer Science & Applications 6(5), 59–88 (2009)

8. 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)
9. Dietz, J.L.G.: Enterprise ontology: theory and methodology. Springer (2006)
10. Dietz, J.L.G.: The deep structure of business processes. Communications of the ACM 49(5), 58–64 (2006)
11. Stichting DEMO kenniscentrum. DEMO: The VISI case, http://www.demo.nl/practical-case-studies/why-visi (last accessed on December 5, 2011)
12. Op't Land, M., Middeljans, K., Buller, V.: Enterprise Ontology based Application Portfolio Rationalization at Rijkswaterstaat. In: The 4th Dutch Championship ICT Architecture (2007)
13. Stichting DEMO kenniscentrum. DEMO: The KLM case, http://www.demo.nl/attachments/article/21/080610_Klantcase_KLM.pdf (last accessed on December 5, 2011)
14. Op't Land, M., Proper, H.A., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture – Creating Value by Informed Governance. Enterprise Engineering Series. Springer, Berlin (2008) ISBN-13: 9783540852315
15. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The Architecture of the ArchiMate Language. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 367–380. Springer, Heidelberg (2009)
16. Jonkers, H., Band, I., Quartel, D.: The ArchiSurance Case Study. White paper, The Open Group (Spring 2012)
17. Cummins, J.D., Doherty, N.A.: The economics of insurance intermediaries. The Journal of Risk and Insurance 73(3), 359–396 (2006)
18. Jonkers, H., Proper, H.A., Turner, M.: TOGAF and ArchiMate: A Future Together. White Paper W192, The Open Group (November 2009), http://www.opengroup.org/bookstore/catalog/w192.htm
19. Zivkovic, S., Kuhn, H., Karagiannis, D.: Facilitate modelling using method integration: An approach using mappings and integration rules. In: European Conference on Information Systems (ECIS 2007) (2007), http://is2.lse.ac.uk/asp/aspecis/20070196.pdf
20. Derzsi, Z., Gordijn, J., Kok, K.: Multi-perspective Assessment of Scalability of IT-enabled Networked Constellations. In: Sprague, R.H. (ed.) Proceedings of the 41st Annual Hawaii International Conference on System Sciences, p. 492. IEEE CS (2008)
21. Object Management Group OMG. Business Process Modeling Notation, BPMN (2007), http://www.bpmn.org
22. Devedzić, V.: Understanding ontological engineering. Commun. ACM 45, 136–144 (2002), doi:10.1145/505248.506002
23. Ehrig, M., Staab, S.: QOM – Quick Ontology Mapping. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 683–697. Springer, Heidelberg (2004)
24. Noy, N.F., Musen, M.A.: The prompt suite: interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies 59(6), 983–1024 (2003)
25. Happel, H., Seedorf, S.: Applications of ontologies in software engineering. In: 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), held at the 5th International Semantic Web Conference, ISWC 2006 (2006)

Form-and-Fact Based Modeling

Terry Halpin[1] and Gerald Weber[2]

[1] LogicBlox, Australia and INTI International University, Malaysia
[2] Department of Computer Science, The University of Auckland, New Zealand
terry.halpin@logicblox.com, gerald@cs.auckland.ac.nz

Abstract. A conceptual data model for an information system specifies the fact structures of interest as well as the constraints and derivation rules that apply to the business domain being modeled. Fact-based modeling approaches provide rich graphical and textual languages for specifying conceptual data models, using attribute-free fact structures that enable models to be verbalized and populated in natural sentences that are easily understood by the domain experts best qualified to validate the models. Form-based modeling approaches offer a natural way for domain users to agree upon suitable user interfaces for interacting with the information system. This paper proposes a synthesis of the two approaches, in which prototype forms are used to seed the conceptual data model, which is then used to generate the final user interface. Semantic and practical aspects of form design are discussed, and screen transition diagrams are employed to help visualize and validate the underlying dynamic processes.

1 Introduction

Computer-based information systems model information about the relevant business domain at various levels. For persistent storage, data is often maintained in structures such as relational database tables, XML schema documents, RDF triple stores, or deductive database clause sets. Although such data structures could be directly updated by the technically savvy, practical information systems typically provide higher level user interfaces, often based on screen forms, to enable users to interact with the system without needing to master the technical aspects of the internal storage structures. Many different user interfaces could be designed for the same information. Moreover, information might need to be displayed in different ways for different user groups. Partly for such reasons, data requirements for an information system are often captured in a *conceptual data model* that provides a simpler, more fundamental way of specifying the fact structures and instances of interest as well as the business rules (constraints or derivation rules) that apply to the relevant business domain.

A conceptual model should be cast in terms of concepts that are intelligible to the business users, so that it can be used to validate the model with them. Once validated, the conceptual model may be implemented using a variety of user interfaces and internal structures, including those used for transient storage of the data (e.g. object oriented code structures, or deductive clause sets) and those used for persistent storage of the data.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 285–299, 2012.

Various languages (graphical or textual) are used by modelers to capture or query the conceptual data model. In attribute-based approaches such as Entity Relationship modeling (*ER*) [6] and class diagramming within the Unified Modeling Language (*UML*) [19]), facts may be instances of attributes (e.g. Person.isSmoker) or association types (e.g. Person speaks Language). UML's Object Constraint Language (OCL) [21, 31] provides a textual means to express class diagrams as well as many additional rules.

In *fact-based modeling* approaches, such as *Object-Role Modeling* (*ORM*) [13], all facts are treated as instances of fact types, often represented as typed, logical predicates (e.g. Person smokes, Person speaks Language). Fact instances also involve existential quantification (e.g. **some** Country has CountryCode 'AU'). For an overview of fact-based modeling approaches, see [11]. Our discussion of fact-based modeling focuses on ORM, overviews of which may be found in [12, 13, 14] and a detailed treatment in [17].

Form-oriented analysis [34] is a modeling method that perceives the user interface as a black-box, in the sense that users may input to, and receive output from, the system without knowing how the system processes these interactions internally. This approach promotes a high level of abstraction by separating the analysis model from internal implementation details. Form-based interfaces, particularly if they are well designed and adhere to a canonical submit/response style, provide an ideal basis for such high level modeling because of their simple yet sufficiently universal semantics. The form-oriented methodology as a set of modeling practices can be applied to any conceptual modeling methodology, but in this paper we leverage its specific and natural synergies with fact-based modeling.

The rest of this paper is structured as follows. Section 2 discusses our approach for exploiting prototype forms to facilitate the design of conceptual data models, and identifies some close connections between the two approaches. Section 3 introduces screen transition diagrams for capturing both static and dynamic aspects of the user interface, and illustrates some of their advantages over other diagramming techniques. Section 4 surveys related work on form production, and outlines how ORM models may be used to generate refined versions of forms for the final user interface. Section 5 summarizes the main contributions and outlines future research directions.

2 From Forms to Conceptual Data Models

Information systems may be modeled at different levels. The four levels indicated in Figure 1 are based on a common refinement to the classic 3-schema architecture [28]. Briefly, the external level deals with the user interfaces, the conceptual level portrays the information in simpler structures that are easily conceived by the users, the logical level concerns the general modeling approaches (e.g. relational, object-oriented) used for implementation, and the physical level deals with the detailed internal implementation (e.g. SQL Server databases, Java code). The arrows in Figure 1 depict our viewpoint on the directions in which transformations should be performed between levels. Prototype external level artifacts such as screen forms are used to seed the design of the conceptual data model, which is then used, as far as possible, to generate the logical and physical models, as well as the final version of the user interface.

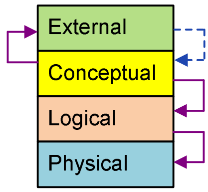


Fig. 1. Modeling at different levels and transforming between them

As an overall approach to developing information systems, we recommend starting with a high level business services model that identifies the main services (what, not how) the system is to deliver. For large systems, we divide the requirements into subareas, and establish dependencies between them, to determine the order in which to build the system components. For each subarea, we then establish the data requirements, and develop a conceptual subschema for the data. The subschemas may then be merged into a global conceptual schema, which underpins the physical and external schemas that are then developed, using automated generation as far as possible.

Fact-oriented modeling approaches such as ORM provide a detailed conceptual schema design procedure (CSDP) [17] for producing a conceptual schema that declares the relevant fact types, constraints and derivation rules that apply to the business domain. The attribute-free nature of fact-based schemas promotes greater semantic stability than that of ER schemas and UML class diagrams (e.g. no remodeling is needed to talk about an attribute), and facilitates model validation by verbalizing business rules in natural language sentences and populating fact types with concrete examples [13]. For data modeling purposes the graphical notation of ORM is far more expressive than that of ER or UML, allowing many more kinds of constraints (both alethic and deontic) to be visualized by the modeler [17]. ORM software tools provide automated support for verbalizing ORM models [15] and for transforming them to other structures (e.g. relational databases, object-oriented code, deductive databases) for implementation [8, 16].

The first step of the conceptual schema design procedure (CSDP Step 1) is to transform familiar examples of required data into atomic facts. This is by far the most important stage of the whole design procedure since it is here that uninterpreted data is assigned basic semantics to transform it into information. Moreover, the information is cast in terms of atomic facts, enabling the information to be viewed in the simplest way possible, with no bias as to how facts might be grouped into composite structures for the internal storage or the external interface.

ORM describes this step as seeding the data model with *data use cases* (cases that illustrate required data being used). Three kinds of data use cases are considered: output reports, input forms, and sample queries. Output reports show examples of data being output by the system, and may appear as tables, graphs, diagrams, or forms of different kinds, typically displayed on screen or in printouts. Input forms populated with sample data show examples of data being input to the system, and may appear on screen or on paper. Input forms are sometimes used to issue queries (e.g. searching a bookseller's website for books by a given author), but other sample queries can be obtained directly from a domain expert (a subject matter expert familiar with the business domain) by asking what kinds of question he/she would like the system to answer. As many information system projects aim to extend or reengineer existing systems, examples of such data use cases are often available. Where such data examples are not readily available, the modeler and domain expert may collaborate to provide them, and use them as a basis for performing CSDP Step 1. In this regard, discussing prototype input forms can serve the dual purpose of determining both data and behavioral requirements.

ORM divides CSDP Step 1 into two phases: (a) verbalize the information; (b) verbalize the information as atomic facts. *Step 1a* is the responsibility of the domain expert. Here, the verbalization may be informal and may involve compound facts. For example, the allergies information conveyed by the form in Figure 2(a) could be verbalized thus: Patient 102 is allergic to penicillin and codeine. *Step 1b* is the responsibility of the conceptual modeler, but requires confirmation by the domain expert. Here each object must be well identified, and each fact must be atomic (an atomic fact cannot be decomposed into two or more simpler facts involving exactly the same object types). The form information stated informally above may be formally verbalized as the following atomic facts: The Patient with patient number 102 is allergic to the Drug named 'Penicillin'; The Patient with patient number 102 is allergic to the Drug named 'Codeine'.

The form must be intelligible to the business user, but by itself is not intelligible to the information system, since it could be interpreted in many ways. The rewording by the modeler in Step 1b is needed to ensure that the information is unambiguously interpreted by the information system, which lacks the subject matter expert's intuitive understanding of the business domain.

The rest of the form is verbalized in a similar way. Informally, we might say that patient 102 is named 'John Smith', has the title 'Mr', is male, and smokes. Though not shown here, a sample form for patient 101 might indicate that patient 101 is female, has the title 'Mrs', doesn't smoke (the Smokes check-box is unchecked), and has no allergies (no entries in the Allergies field). To complete the CSDP, the ORM modeler generalizes such sample fact instances to fact types, enters them on an ORM schema diagram, and then adds the relevant constraints and rules [17]. While ORM schema diagrams are very useful for the modeler to quickly visualize the detailed model semantics, validation of the conceptual model with the domain expert is often performed simply by verbalizing the schema and populating it with concrete examples.

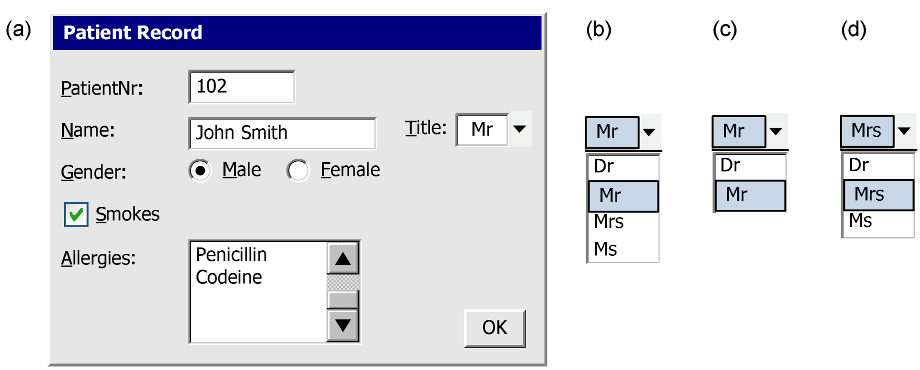


Fig. 2. (a) A simple screen form; (b) drop-down list; (c), (d) gender restricted drop-down lists

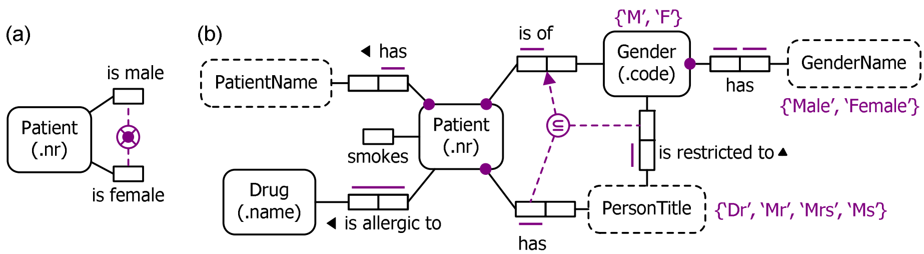


Fig. 3. (a) An ORM schema fragment for part of Figure 2; (b) a detailed ORM schema

Our form example included radio buttons to indicate the gender of the patient. One way of modeling this in ORM is shown in Figure 3(a). A solid, rounded rectangle denotes an entity type, in this case Patient. The parenthesized reference mode "(.nr)" indicates that patients are identified by their patient numbers. The boxes attached to Patient depict roles played by instances of Patient. Logical predicates are depicted as an ordered set of one or more role boxes, together with a predicate reading. Here there are two unary predicates, is male and is female. Constraining predicates to object types results in fact types. Here we have two fact types: Patient is male; Patient is female. The circled, crossed dot between the gender roles depicts an exclusive-or constraint, which verbalizes thus: **Each** Patient is male **or** is female **but not both**.

ORM includes formal procedures for determining equivalence between conceptual schemas [10], as well as heuristics for "optimizing" such schemas to choose a preferred way of modeling domain features [17]. By default, exclusive-or patterns such as that of Figure 3(a) are transformed into a single, functional fact type, such as the Patient is of Gender fact type in Figure 3(b). The bar on the first role of this fact type depicts a uniqueness constraint (**Each** Patient is of **at most one** Gender), and the dot attached to its role connector indicates that the role is mandatory for Patient (**Each** Patient is of **some** Gender). Here Gender instances are identified by gender codes, but also have unique names. The value lists in braces depict value constraints on the possible genders. So the exclusive-or constraint in Figure 3(a) is now captured by constraining each patient to have exactly one of two genders.

Rounded rectangles with dashed lines (e.g. PatientName) denote value types, whose instances are simply typed constants. Predicate readings are read from left-to-right or top-to-bottom unless their reading direction is reversed by adding an arrow-tip (e.g. Patient is allergic to Drug). The spanning uniqueness bar on the allergy fact type indicates that it is a many-to-many relationship. Though the form in Figure 2(a) did not explicitly indicate that the patient number, name, title and gender details are mandatory, our ORM model includes these mandatory role constraints. Sometimes, mandatory fields on forms are marked (e.g. with an asterisk). At any rate, if users attempt to commit a form without completing all its mandatory fields, they are notified of this omission, and returned to the form to complete the data entry.

In Figure 3(b), the circled subset symbol depicts a *join-subset constraint* from the attached source roles to the role-pair attached at the arrow-tip. This verbalizes as follows: **If a** Patient has **a** PersonTitle **that** is restricted to **some** Gender, **then that** Patient is of **that** Gender. For example, assigning a male person the title 'Mrs' would violate this constraint.

Figure 2(c) shows a drop-down list for the Title field. For simplicity, we include just four values here (other values such as ‘Sir’ and ‘Lady’ could also be included). If users are free to select the Title entry before the Gender entry, they have the potential of violating the join-subset constraint. In some systems, such a constraint is not checked (if at all) until the form is committed. In general it is usually better to check a constraint as soon as possible, in order to avoid the need for data re-entry by the user. Where possible, the user interface itself should make it impossible for users to violate constraints. For example, radio buttons or mandatory drop-down lists provide simple ways to enforce exclusive-or constraints.

One way to prevent users from violating join-subset constraints like that in Figure 3(b) is to support *dependent drop-down lists*, whose list options depend on a previous entry. For example, if the system controls form navigation to ensure that the user selects the gender before the title, the drop-down list of titles displayed to the user can be restricted to the titles that are valid for that gender. If the user selects the Male radio button, the drop-down list in Figure 2(c) is displayed, but if the user selects Female radio button, the drop-down list in Figure 2(d) is displayed. Such possibilities reveal one way in which electronic forms provide options beyond that of paper-based forms. While paper forms can include conditional fields and go-to-field instructions, they cannot strictly force the user to act accordingly.

Historically, the earliest forms were fill-in-the-gap legal documents. Typically, users would input facts by entering items such as names or numbers in gaps within open sentences, such as “This contract expires on day ___ of the month _____ of the year ______”. This immediate portrayal of the relevant facts in terms of natural language sentences helped to ensure that the intended semantics was well understood. The fact verbalization process within ORM’s CSDP Step 1 may be viewed as reviving the original notion of forms. In later steps of the CSDP, all relevant constraints and derivation rules are captured.

We distinguish between *conceptual forms* and their visual representations in the user interface. Structurally, a conceptual form is an ORM subschema that captures the intended semantics of a physical form (paper-based or electronic), ignoring layout aspects. We use the term “model” to mean a schema populated with facts. A completed conceptual form includes instance data, and thus corresponds to an ORM submodel.

Apart from its rich graphical notation, ORM includes FORML (Formal ORM Language), which is a controlled natural language that allows ORM models (including fact types, constraints, derivation rules, and fact instances) to be fully verbalized in language that is intelligible to domain experts [18]. The ORM verbalizations in this paper all use FORML. Hence a conceptual form may be fundamentally represented as a set of controlled natural language sentences. Many different physical forms may be designed for the same conceptual form.

When validating models with the domain expert, the models should be expressed in a way that is unambiguous to the domain expert. The use of controlled natural language for verbalizing the model serves this purpose well. Physical forms that are well designed are usually easy for users to understand, and are especially helpful for determining the kinds of facts that need to be maintained by the system. Various underlying constraints and derivation rules might not be obvious from single static forms, but can be exposed by illustrating system responses to errors, including error messages based on verbalizations of the relevant rules.

3 Screen Transition Diagrams

Screen Transition Diagrams are visual models of dialogues, including their dynamic aspects. The screen transition diagram notation used in this paper is new, but is based on an extension to the form chart notation [34] developed by one of the authors. Screen Transition Diagrams focus on a simple but powerful type of user interface (UI) that can be precisely defined and is termed a submit/response style interface. The dynamic aspects of the interface are divided into two types of change, with a clear hierarchy and clear complementary purposes. The semantically most heavyweight type of dynamic behavior is page change. The typical triggers of a page change are mouse clicks, although there are cases such as search forms, where hitting the return key is enough to trigger a page change. A page change is triggered by a single, atomic action of the user, although there might be many options on a page that can trigger a page change.

It is a natural requirement of the principle of conformity with user expectations [19] that a page change should be triggered only if the user expects it; so it must be intentionally triggered by an atomic action. A page change often has heavyweight semantic consequences, such as a purchase or signing a contract. The semantics often require a page change for validity reasons. For example, a contract should only be entered intentionally. It would be unacceptable if a business partner considered a user to have entered a contract at some point simply by browsing long enough through a system. The intentionality is related to one of the lesser known functions of a signature besides authentication, identification and acknowledgement functions: requiring a signature constitutes a warning. We should be alerted that things may get serious if we have to sign something. As a consequence, pleading ignorance about signing something is usually not an accepted excuse. Hence fundamental aspects of this style of user interface can be expected to be here to stay. Interestingly, even single-window based UIs that do not exhibit state changes such as instant stock trading systems will respond with at least a pop-up window in the case of a contract.

Page change is captured as state change, but with a twist. It is the very nature of an interactive system that the system may react in a way that is not always completely predictable. For example, when booking a seat on a crowded flight, the last seat might have just been taken. In submit/response style systems, the system is usually expected to respond on the immediately following page with more than just an acknowledgement of receipt of the form. Hence the system is expected to instantaneously process, assess, follow up on the input, and commit itself, for example by acknowledging that a seat on the plane is now booked and reserved. This means that during the short time in which the system needs to respond, a lot is happening, usually at least a database transaction. On a screen diagram this is indicated by a small symbol indicating an intermediate state.

This intermediate state is called a server action. Since the intermediate server action is a proper state of the system, the state transition is a bipartite state machine. A state during which a page is shown is called a client state, indicating that this state is local to a client and usually has limited access to server data. The system alternates between client pages and server actions. In a high-level model, the server actions are always short lived with negligible duration, and hence are transactional. A page change is therefore a double state transition, and can be decomposed into two transitions, the client/server transition and the server/client transition. If there are several such transitions incident to the same server action, then the whole set of client/server transitions and the server/client transitions incident to the same server action constitute the same kind of page change.

3.1 Page Interaction

A form collates information that is necessary to give the subsequent page change correct semantics. For example, a purchase can happen only if necessary data, such as personal details, have been provided on a form preceding the purchase. The many fine-grained interaction steps that are necessary to do that comprise a page interaction. They are often on a lower cognitive level, for example on the habitual level. Part of our computer literacy is that we can intuitively navigate on a typical form. Often this navigation can be non-linear, first filling out fields that are easier to complete, then turning to fields that are more complex, or require us to lookup data on paper or elsewhere.

A state machine for page interactions may be particularly complex, in fact in some cases non-finite, and at the same time particularly uninforming. This is aggravated by the fact that page interaction is often non-modal, meaning that the user does not perceive any change of affordances while staying on the same page. This in turn means that there is no noticeable state change, and the effort of creating the considerably complex state machine would be of little added value. Since page interaction is non-modal, the graphical depiction of a page is mostly sufficient to explain its function. It is again an instance of the principle of conformity with expectations that if we see a form we should know what we have do with it.

3.2 Screen Transition Diagram Example

Screen transition diagrams depict both page interaction and page change in separate, appropriate ways and yield a comprehensive system model. An example screen transition diagram is shown in Figure 4, for an application that is loosely based on the patient record example used so far.

There are two separate classes of modeling elements in it, the page prototypes and transitions between the pages. The page prototypes depict forms with standard form elements. The way in which a form element is depicted might vary based on which style is assumed to be most easily recognized. The screen prototypes are mostly close to simplified screenshots of a concrete style of interface. Since for example the representation of drop-down lists varies quite a bit between different interface frameworks, there is room for variations of the representation. It is often advisable to stay with interaction elements that are widely in use and have clear semantics. Even among those, there is some potential for reductionism. For example a single selection drop-down list is equivalent to a radio button group, and a multiple selection list is equivalent to a checkbox group. Such semantic minimalism is useful for understanding the mapping between the familiar form elements and the fact-based natural language representation discussed in Section 2. Selection lists warrant mention of another common element of paper forms. Paper forms may contain not only fill-in-the-blank elements but also multiple choice elements, use of which may involve scratching out inapplicable choices, highlighting applicable choices, or a combination of both.

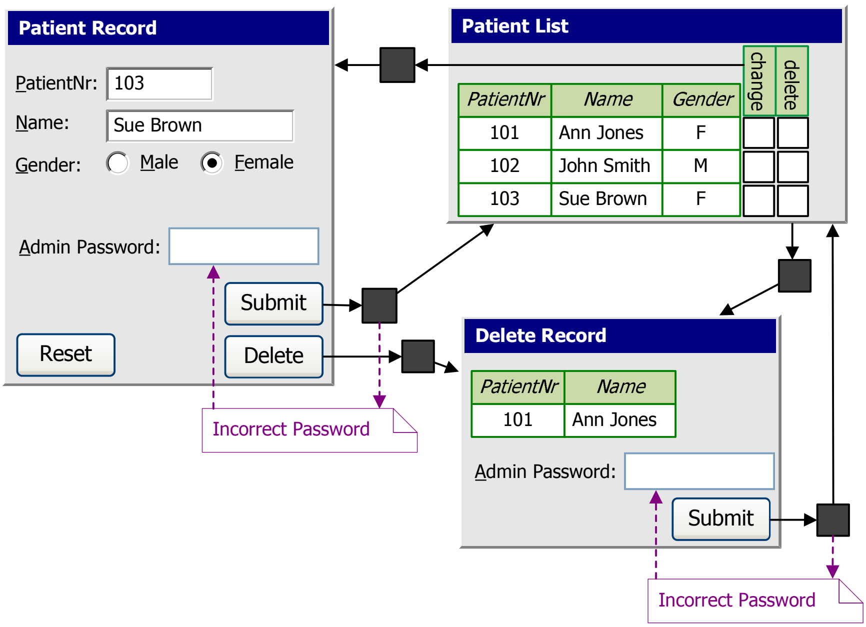


Fig. 4. A sample Screen Transition Diagram

Page prototypes as a whole have a second function in the screen transition diagram. Each page prototype represents a state in a state transition diagram. Each represents the state that the user interface is in when this page is shown to the user. The state is exited on a page change. This leads to the second class of modeling elements in a screen transition diagram, namely the state transition annotation representing page changes. In a screen transition diagram, a page change is always depicted as a two-step process with an intermediate state, the server action. The server action is depicted as a small black square, preferably with no description. This serves several purposes.

First of all, this helps making the diagram visually simpler, since page change arrows are easily distinguished as those arrows that start or end in a server action. This is important particularly for hand-drawn diagrams as they might be developed spontaneously with domain experts. Secondly, often the server action is a branching point in the screen transition diagram, as shown in Figure 4 for the server actions where a branch for the error condition incorrect password is shown. Solid arrows depict normal transitions, while dashed arrows are used to show transitions invoked by an error. As a third motivation, the server action can serve in some instances as a way to simplify the diagram e.g. if the same server action can be targeted by different pages. This is shown for the delete feature. It can be chosen once from the list page and also from the page showing an individual record. Since the server action is a proper state, the screen transition diagram is a bipartite state transition diagram.

Arrows exiting from a server action may point to the subsequent page, or to a specific field on the page if cursor placement is relevant. For example, if a password is incorrectly entered an error message is displayed and the cursor placed in the password entry field.

3.3 Modal Dependencies

There are, however, a few common types of truly modal state changes on a page that are worth modeling and that do not lead to an unacceptable increase in the complexity of the form. These kinds of state change can be expressed largely in terms of the form elements themselves, together with a limited set of behavioral annotations. An example is the gender-dependent dropdown list from Figure 2(a). We call such a dependency a *modal dependency*, and we depict it on a screen transition diagram as an arrow pointing from the form element that changes the state to the dependent form element, as shown in Figure 5.

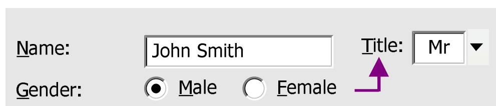


Fig. 5. One kind of modal dependency indicated in the form

Another common case of modal dependency is activation of form fields based on previous entries made on the form. For example, in the fragment of a room booking form shown in Figure 6(a) the details on kind of breakfast and preferred breakfast time are relevant only if the user checked the breakfast required box. On a paper form, these details would often be indented or grouped in a way that suggests their dependency on that check box. On an electronic form, this dependency can be made obvious by greying out or even suppressing the display of the breakfast details (which could be provided in another dialog) unless the relevant check box is activated.

Conceptually, these kinds of dependencies are typically modelled by introducing *subtypes*. For example, the ORM schema in Figure 6(b) models this case by using the subtype RoomWithBreakfastRequest to record those room requests that require a breakfast, and then attaching the specific breakfast details to that subtype (indicating they are recorded only for instances of that subtype). Like UML, ORM displays subtyping relationships as arrows directed from subtype to supertype.

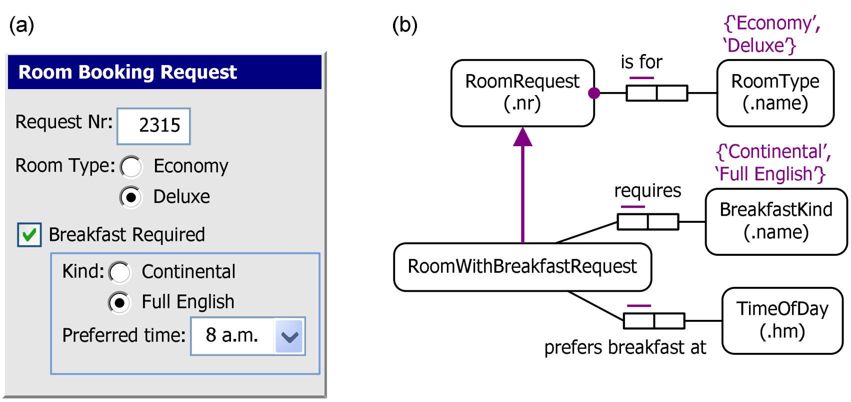


Fig. 6. Modeling another kind of modal dependency on (a) a form, and (b) in ORM

This kind of modal dependency may also be modeled on forms by including instructions to skip parts of the form, or equivalently, jump to a later section, based on entries or answers supplied earlier on the form. Income tax forms are a good example of this, and involve many subtypes. Conceptually these kinds of modal dependencies require that certain kinds of fact are recorded only for specific subtypes (implicit or explicit). In contrast, the gender-restricted title dependency considered earlier simply restricts the population of a given fact type (in this case, Patient has PersonTitle). The same kind of title fact is recorded for males and females, and it is only the possible instances that differ. This difference is clearly indicated by the different conceptual modeling structures used, and not surprisingly leads to different recommended structures in the external forms.

Theoretically, one *could* treat the gender-restricted title dependency like the subtyping model dependencies by creating a distinct fact type for each person title. This would lead to several unary fact types such as Patient has a mister title; Patient has a doctor title; Patient has a mrs title, etc., which could be presented on forms in a basic format such as that shown in Figure 7. However, such an approach consumes more screen space, and rapidly becomes unwieldy as the number of possible titles grows (e.g. consider adding ‘Sir’, “Lady’, ‘Prof.’ etc.). For such reasons, this approach is considered suboptimal at both conceptual and external levels.

o Gender: male
  o Title: Mr
  o Title: Dr
o Gender: female
  o Title: Mrs
  o Title: Ms
  o Title: Dr

Fig. 7. An alternative way of dealing with gender-title restrictions

3.4 Decomposition of Screen Transition Diagrams

In a screen transition diagram all elements on a single screen are known as screen description elements. From a screen transition diagram we can extract the pure state transition diagram for page change by abstracting all screen description elements. Furthermore it is necessary in this diagram to name the intermediate server action, so that the diagram is in a basic way self-contained. The resulting diagram is called a *form chart*.

For large systems there is a need to decompose diagrams into smaller parts. For form-based systems a simple way of decomposition with clear semantics is available, called feature decomposition [34]. It is helpful to describe feature decomposition first on the level of form charts. The decomposition is based on the observation that certain subdiagrams might not describe the whole interaction, even though the subset that they describe is correct. Several subdiagrams are combined by graph union, i.e. no diagram parts are deleted in order to obtain the whole model. This is demonstrated in Figure 8. In this figure, form charts (a) and (b) each model a feature of the system. Form chart (a) allows that a login can fail, while form chart (b) includes a logout option. Both (a) and (b) are combined to yield the system form chart (c).

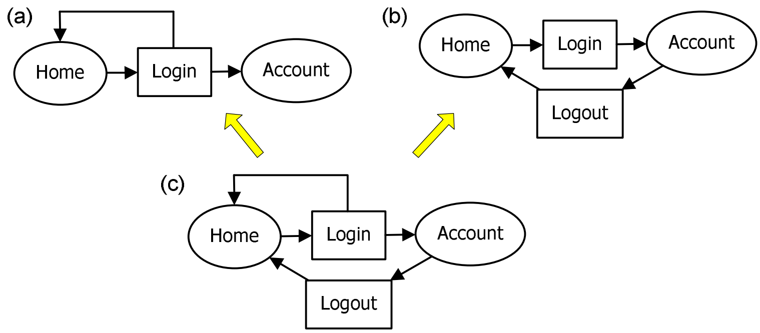


Fig. 8. Three form charts: (a) and (b) are subdiagrams of (c)

Although (c) is unlikely the form chart of a final system, it models already a basic system behavior that makes sense on its own. The decomposition also shows two interesting examples of how the semantics of the system changes from the subsystem to the whole system. In the transition from (a) to (c), a new user option is added, the logout feature. The new option likely does not affect the earlier existing options. In the transition from (b) to (c), however, the change affects the meaning of the original diagram. In (b) it appears that the login always leads to the account, while actually things may go wrong and the login can fail. While this is expected in the case of login, in other cases this can be unanticipated. For example in a webshop, it is possible that purchasing a shopping cart always works (i.e. the items are reserved), but it is also possible that the items are not reserved and the purchase can fail.

4 Mapping Conceptual Models to Forms

Mapping models to user interfaces has been a recurring topic in the application of modeling methods, leading to the development of model-based user interface environments (MBUIDE) [27]. Models for user interfaces or related technologies are a classical application of domain-specific modeling. A specific area of interest has been Web applications [7, 29]. However, typical Web modeling frameworks do not map conceptual models, but instead map implementation models to user interfaces. Other approaches focus on embedding user interface modeling in very specific modeling methodologies [20, 30]. Instead of providing a primarily generative approach, these approaches define frameworks for models that specifically model the user interface, and they provide at best balancing rules which express when a user interface model is compatible with the wider system model. In contrast, our interest is in an approach that generates natural interface models from high-level conceptual models.

The area of MBUIDE is an active research field with new challenges through novel formalisms such as semi-structured data [24]. Also there is interest in model-based development of very specific user interfaces such as tangible user interfaces [25]. Form-based systems have always been on the one hand a core technology, for example in the enterprise application area [9]. On the other hand, they have often been neglected in the modeling community since they superficially have a mismatch with modern object oriented models, while GUIs are classically object oriented. One common modeling area that tangentially deals with form-based systems is Web modeling [5]. However, in many Web modeling methods, the form-based elements of the Web system receive only trivial models, since the dynamic behaviour (i.e. the business logic itself) is not modeled.

Within the fact-based modeling community, a detailed procedure for mapping ORM schemas to external schemas such as form-based screens was first co-developed by Linda Bird (nee Campbell) and one of the authors. This procedure included algorithms for anchoring fact types to major object types (intuitively, the most important entity types), and for constructing an insertion-order graph to determine the order in which facts of different kinds may be entered. These two structures were then used as a basis for grouping fact types into subschemas, each of which provided the conceptual underpinnings of a screen form [3].

Originally, the major object type and anchoring procedure was based purely on an analysis of constraints in the ORM schema. As trivial examples, an object type with a mandatory functional role is more important than an object type that plays only optional roles, and mandatory functional roles provide an anchor for their fact type. This procedure was later extended and used as an abstraction mechanism to recursively refine large schemas into smaller, higher level overviews [4]. The notion of major object type was made relative to the refinement level, the proportion of object type populations participating in optional roles was taken into account, and user-determined weighting factors could be assigned to various constraint categories.

Building on this earlier work, and encouraged by the results of early prototypes, we are currently working on an ORM-to-Forms mapping procedure that outputs a set of candidate forms that can be customized by the forms designer. First, ORM subschemas corresponding to conceptual forms are determined, allowing the designer to override aspects of the default grouping procedure.

In generating labels on forms we adopt the principle that the intended meaning should be clear to users without requiring them to perform complex processing. As a trivial example, our approach ensures that unit prices of line items on a shopping cart Webpage are labeled “Unit Price” along with their currency unit (e.g. USD). In contrast, shopping cart pages of major companies often use the label “Price” (which could be misunderstood as a line total price because the quantity of items is displayed on the same line) without a unit (leading to possible misunderstanding for users of other currencies). If the total price for the shopping cart appears on the same page, the user could perform calculations to determine that a unit price is intended, but it is safer and less burdensome for the user to just use a better label in the first place.

5 Conclusion

This paper proposed a synthesis of form-based modeling and fact-based modeling in which prototype forms seed the conceptual data model, which is then used to refine the final user interface, using screen transition diagrams to help visualize and validate the underlying dynamic processes. Since typical forms, verbalized ORM schemas, and screen transition diagrams are all intuitively understood by business users, this approach facilitates validation by the domain experts at all stages of the information system development. Reflecting the clear and rich semantics provided by ORM models in the user interface provides a disciplined way to ensure that the final forms are both unambiguous and sensitive to the underlying business rules.

Ongoing research aims to extend the software support for the approach. For example, we aim to address the fact that different forms may be required for different roles (i.e. user categories), by utilizing metalevel fact types such as User instantiates Role, and Role has AccessRight to ModelElement, where model elements may be fact types, fact roles, or object types. For the schema in Figure 3(b) for example, assert and retract rights to the fact type PersonTitle is restricted to Gender would typically be granted only to those with an administrator role. For generation of the physical forms, default layout options for various schema features (e.g. radio buttons or dropdowns for exclusive-or constraints) may be either accepted or overridden by the designer. We believe such a semi-automated procedure is currently more practical than relying on full automation.

References

1. Balsters, H., Halpin, T.: Formal Semantics of Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 699–708. Springer, Heidelberg (2008)
2. Bird, L., Goodchild, A., Halpin, T.: Object Role Modelling and XML-Schema. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 309–322. Springer, Heidelberg (2000)
3. Campbell, L., Halpin, T.: Automated Support for Conceptual to External Mapping. In: Brinkkemper, S., Harmsen, F. (eds.) Proc. Fourth Workshop on the Next Generation of CASE Tools, Paris, pp. 35–51 (1993)
4. Campbell, L., Halpin, T., Proper, H.: Conceptual Schemas with Abstractions: making flat conceptual schemas more comprehensible. Data and Knowledge Engineering 20(1), 39–85 (1996)
5. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): A modeling language for designing Web sites. In: Proc. 9th International World Wide Web Conference, Amsterdam (2000)
6. Chen, P.P.: The entity-relationship model—towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976), `http://csc.lsu.edu/news/erd.pdf`
7. Conallen, J.: Modeling Web Application Architectures with UML. CACM 42(10), 63–70 (1999)
8. Curland, M., Halpin, T.: The NORMA Software Tool for ORM 2. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 190–204. Springer, Heidelberg (2011)
9. Draheim, D., Weber, G.: Form-Oriented Analysis: A New Methodology to Model Form-Based Applications. Springer, Heidelberg (2004)
10. Halpin, T.: A Logical Analysis of Information Systems: static aspects of the data-oriented perspective. Doctoral dissertation, University of Queensland (1989), `http://www.orm.net/Halpin_PhDthesis.pdf`
11. Halpin, T.: Fact-Oriented Modeling: Past, Present and Future. In: Krogstie, J., Opdahl, A., Brinkkemper, S. (eds.) Conceptual Modeling in Information Systems Engineering, pp. 19–38. Springer, Berlin (2007)
12. Halpin, T.: Object-Role Modeling. In: Liu, L., Tamer Ozsu, M. (eds.) Encyclopedia of Database Systems. Springer, Berlin (2009)
13. Halpin, T.: Object-Role Modeling: Principles and Benefits. International Journal of Information Systems Modeling and Design 1(1), 32–54 (2010)
14. Halpin, T.: Fact-Orientation and Conceptual Logic. In: Proc. 15th International EDOC Conference, pp. 14–19. IEEE Computer Society (2011)

15. Halpin, T., Curland, M.: Automated Verbalization for ORM 2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1181–1190. Springer, Heidelberg (2006)
16. Halpin, T., Curland, M., Stirewalt, K., Viswanath, N., McGill, M., Beck, S.: Mapping ORM to Datalog: An Overview. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010 Workshops. LNCS, vol. 6428, pp. 504–513. Springer, Heidelberg (2010)
17. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
18. Halpin, T., Wijbenga, J.P.: FORML 2. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 247–260. Springer, Heidelberg (2010)
19. ISO 9241-110:2006 Ergonomics of human-system interaction – Part 110: Dialogue principles. ISO (2006)
20. Object Management Group: UML 2.0 Superstructure Specification (2003), `http://www.omg.org/uml`
21. Object Management Group: UML OCL 2.0 Specification (2006), `http://www.omg.org/spec/OCL/`
22. Object Management Group: Semantics of Business Vocabulary and Business Rules, SBVR (2008), `http://www.omg.org/spec/SBVR/1.0/`
23. Paterno, F.: Model-Based Design and Evaluation of Interactive Applications, 1st edn. Springer, London (1999)
24. Paterno, F., Santoro, C., Spano, L.D.: MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. ACM Trans. Comput.-Hum. Interact. 16(4), Article 19, 30 pages (2009)
25. Shaer, O., Jacob, R.J.K.: A specification paradigm for the design and implementation of tangible user interfaces. ACM Trans. Comput.-Hum. Interact. 16(4), Article 20 (2009)
26. Pinheiro da Silva, P., Paton, N.W.: UMLi: The Unified Modeling Language for Interactive Applications. In: Evans, A., Caskurlu, B., Selic, B. (eds.) UML 2000. LNCS, vol. 1939, pp. 117–132. Springer, Heidelberg (2000)
27. Pinheiro da Silva, P.: User Interface Declarative Models and Development Environments: A Survey. In: Palanque, P., Paternó, F. (eds.) DSV-IS 2000. LNCS, vol. 1946, pp. 207–226. Springer, Heidelberg (2001)
28. van Griethuysen, J. (ed.): Concepts and Terminology for the Conceptual Schema and Information Base. ISO TC97/SC5/WG3. Eindhoven (1982)
29. Vilain, P., Schwabe, D.: Improving the Web Application Design Process with UIDs. In: 2nd International Workshop on Web Oriented Software Technology, Malaga, Spain (2002)
30. Vilain, P., Schwabe, D., Sieckenius de Souza, C.: A Diagrammatic Tool for Representing User Interaction in UML. In: Evans, A., Caskurlu, B., Selic, B. (eds.) UML 2000. LNCS, vol. 1939, pp. 133–147. Springer, Heidelberg (2000)
31. Warmer, J., Kleppe, A.: The Object Constraint Language, 2nd edn. Addison-Wesley, Reading (2003)
32. Weber. G.: Technology-Independent Modeling of Service Interaction. In: van Sinderen, M., Almeida, J.P.A., Pires, L.P., Steen, M. (eds.): Workshops Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference, EDOCW 2008, Munich, Germany (2008)
33. Weber. G: Defining the Paperless Workplace with the Paper Metaphor - Not a Contradiction in Terms. In: Butler-Henderson, K., Sahama, T. (eds.) Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2011), Perth, Australia (2011)
34. Weber. G.: Semantics of Form-Oriented Analysis. Doctoral Dissertation. urn:nbn:de:kobv:188-2003000724, Freie Universität Berlin (2003)

A Framework for Relating Business Constraints to Information Systems

Andrew Carver and Tony Morgan

INTI International University, Malaysia
andy.carver@yahoo.com, goldbasilisk@live.com

Abstract. Many previous attempts at classifying business rules rely on over-simplistic frameworks that conflate business concerns with technical features. Such frameworks hamper traceability between information systems and business needs and can lead to paradoxes that are difficult to reconcile. This paper offers an alternative framework for business constraints, including those that can be embodied in information systems. We assume that such information systems are likely to be automated, but the proposed scheme does not rely on any automation. The paper uses several examples to illustrate the issues that arise with current classification frameworks and the benefits that a more realistic framework can provide.

1 Introduction

The capture, modeling, and management of business rules has been a major concern for two decades. More recently, there have also been several initiatives aimed at defining standards relating to business rules.

A persistent interest has been the desire to classify rules into a number of categories. As was shown in [1], however, a recently popular categorization scheme used in SBVR [14], classifying rules according to the modal-logic categories "alethic" and "deontic", leads to some serious and intractable paradoxes, such as:

- the anomalies of the idea "alethic business rule";
- being forced into the anomalous expedient called "enforcement levels".

This scheme also contributes little help to the problem of providing traceability of business concerns, between different expressions or artifacts classifiable under it.

But it is not only this categorization scheme, as such, that leads to such paradoxes and issues. A deeper analysis shows that an over-simplistic but common assumption about information systems and their development frameworks leads to these problems. On the other hand, a more realistic and flexible approach to these matters effectively resolves these problems.

Section 2 of this paper explores a more generic approach to frameworks and their construction. In section 3 we elaborate consequently a more-flexible framework for business-constraint capture and modeling, which remedies the built-in, over-simplistic assumption inherent in traditional frameworks and causing the above problems. In section 4 we locate, within this framework, the native "levels" or domains of several

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 300–314, 2012.

different types of business constraint, and indicate some natural mappings between some of these rule types. In section 5 we draw out a further, perhaps unexpected benefit of this set of categories, for information-system design. Section 6 presents our conclusions.

2 Towards a More Generic Approach to Information-System Development

2.1 The Organizational Need

In this section we define some concepts that are precursors to the discussions in later sections. This a necessary step, because there is no consistent set of terms used in the available literature. In fact, the opposite is true: the same terms are used in different senses by different authors, making it hard to compare the underlying significance of different approaches. In this paper we are primarily concerned with the construction of computer-based information systems, but we believe that the approach is sufficiently general to encompass other outcomes, such as defining activities that have no corresponding computer implementation.

The general scenario we assume is the following. An organization wants some particular concerns addressed in a systematic manner. These concerns relate to activities carried out by the organization and/or records kept by the organization of its activities. The organization deliberately wants to constrain its activities and/or record keeping so that it is coerced towards operating in a particular fashion. For example, the organization might wish to meet certain goals, or avoid particularly undesirable situations. Recent legislative moves, such as the Sarbanes-Oxley Act [2] have added a further dimension. It is no longer sufficient to state out some directives and assume that they will be followed. Organizations (particularly senior executives in those organizations) need positive assurance that either a directive is definitely followed, or the management chain is alerted to its non-observance so that relevant action can be taken.

Such issues have raised interest in traceability between organizational needs and how those needs are being satisfied. For example, in the context of a computer-based information system we might want to ask the following questions:

(a) considering a particular organizational concern, how is it addressed in the information system?
(b) considering a particular element of the information system, what organizational concern(s) does it address?

We return to the topic of traceability in a later section.

2.2 The Use of Models

Our approach to the situation outlined above is based on the use of models. Here, we define the terms that we use in our descriptions.

We assume that in most cases we will be dealing with a particular set of concerns, rather than the complete concerns of the organization. It is usually impractical to deal with a complete enterprise, for reasons unconnected with any particular modeling approach. The relevant set of concerns can be referred to as the "scope" or "Universe of Discourse (UoD)" of the model. A model is typically considered to be composed of two ingredients: a schema, which describes a particular structure of element types relevant to the UoD, and a population of particular element instances that conform to the schema. Here we will mostly be interested in model schemas.

It has been clear for some time that a particular set of organizational concerns can not be adequately described by a single model. This should come as no surprise, since similar conclusions have been reached in just about every area of human endeavor. For example, a building has separate plans for structure, ventilation, plumbing, etc. An electronic assembly has separate diagrams for logical circuit connections, physical layout, wiring, etc. An aircraft type has separate blueprints for engines, avionics, cabin layout, and so on.

How many models do we require? The correct answer should be "as many as are useful". Several frameworks have tried to be more specific and define a particular set of models. These are often described as different "levels" of model by their proposers. Here are some examples.

- OMG's Model Driven Architecture (MDA) approach [3] originally discussed three levels: Computation Independent Model, Platform Independent Model and Platform Specific Model. Recent OMG publications are less specific.
- Object Role Modeling (ORM) [4] describes four levels: Conceptual (the level addressed by ORM), External, Logical and Physical, but admits the possibility that there could be more.
- The Zachman Framework [5] has six levels: Contextual, Conceptual, Logical, Physical, Detailed and the functioning entity itself. The names of the levels have varied over time, but the number has remained the same.
- RM-ODP – the Reference Model for Open Distributed Processing is an ISO standard [6]. It describes five "viewpoints" (not levels): Enterprise, Information, Engineering, Technology and Computational.
- The National Association of State Chief Information Officers (NASCIO) suggests at least four "architectures" [7]: Business Architecture, Information Architecture, Solution Architecture, and Technology Architecture.
- UML [8] defines thirteen different "diagram types" to cover different facets of software design. Many of these relate to specific technical concerns in OOP, and are not really relevant to reflecting the needs of an organization.
- The ISO/IEC/IEEE 42010 standard [9] (previously IEEE recommendation 1471) assumes multiple "viewpoints" in an architecture description, and offers as example viewpoints: operational, systems, technical, logical, deployment, process, and information. This standard also uses the term "model" in a slightly different way to most of the approaches mentioned here.

Various other Enterprise Architecture frameworks such as TOGAF [10], DoDAF [11], MoDAF [12] and so on concur that multiple representations are necessary and inevitable. The general consensus seems to be that somewhere in the region of three to six different representations might be required to reflect organizational needs, and more if we wish to get down to detailed technical levels that implement those needs.

2.3 A Generic Structure

In order to support later discussions, we describe here a generic approach that avoids many assumptions that are built in to the various frameworks listed above. We assume that the definition of a complete information system will require several models, but we do not prescribe any particular number of models or model types.

2.3.1 Domains

Each model exists in the context of a particular domain. Each domain has:

- a set of elements from which models can be constructed,
- a set of construction rules that govern how the elements may be composed,
- one or more notations that are used to express models in that domain,
- a set of operations that can be applied to model elements.

The first two of these can be thought of as akin to the notion of a "language" and a "grammar", though we wish to avoid linguistic discussion in this paper. The sense of "domain" here is closer to the mathematical usage (e.g. "the set of elements to which a mathematical or logical variable is limited, specifically the set on which a function is defined" [16]).

There may be notational differences between models in the same domain, such as:

- shapes of graphical objects,
- right-to-left text versus left-to-right text,
- formatting properties such as fonts, line styles, colours, etc.

However, all models in the same domain must support exactly the same set of non-notational operations. Obviously, some operations may not be relevant in a particular model if they relate to elements that are not present in that model.

We assume that multiple models will be required to capture the full details relating to a business information system because we need to deal with different kinds of concerns. Unlike most of the frameworks described above we see no reason to prescribe, as universal truth, any specific number of domains. We also avoid specifying layers or levels: in general, domains are not "above" or "below" other domains. Domains may differ from each other in characteristics such as precision, understandability, convenience, etc. Different domains may share the same set of elements and notations, but a different set of operations. For example, two domains may use the same elements and notations but admit different axioms. Notations do not define a domain; it is the set of elements and the set of operations that distinguish one domain from another.

It is important to note that the name of a domain has no semantic content. It is simply an identifying label. We might prefer names that are reminiscent of the nature of the domain, rather than merely codes, but this is a matter of preference. In particular, simplistic or ill-advised choices of labels should not be taken to imply anything about the domain – they are simply poor names. For example, we might label a particular domain as "Conceptual", but this should not be taken to mean that there is only one domain dealing with concepts. Indeed, experience shows that several of the domains required for a comprehensive modeling approach could be considered to be conceptual in nature.

A significant difference between the idea of domains, discussed above, and the different "levels" of model proposed in existing frameworks is that we see the domains as peers by default. The term "level" implies "a position in a scale or rank" [16], which is a meaningless statement when applied to comparisons between most domains. It is certainly possible for domains to be related in a superior/subordinate manner: for example, where one domain is intended to provide a summary of another domain. However, where such relationships exist, we consider that they are best described as domain metadata, rather than being built into a modeling framework as fixed assumptions.

2.3.2 Mappings

Models in different domains are related by mappings. A mapping expresses a relationship between elements in one domain and elements in another domain. Mappings are unidirectional, from source model to target model, and a complete inverse mapping will not necessarily exist. For example, many mappings will not preserve all of the information contained in a source model, so it will not be possible to reconstitute the source model from the target model. This has certain implications for traceability. This use of the term "mapping" here is close to its use in mathematics, but in our case it is unlikely that the mappings would be accomplished by simple functions.

Another implication is that so-called "round-trip" modeling may be of limited value. If we have a mapping from Domain X to Domain Y and another mapping from Domain Y to Domain X, we may find it impossible to update a model in Domain X with the results of operations carried out on a model in Domain Y, because the latter does not fully reflect all of the features of the other domain.

A mapping from one domain to another does not necessarily exist, even if the two domains are precisely defined. For example, one domain may be concerned with spatial features and another with temporal features, so there may be no direct points of correspondence between the two. As with domains, mappings may be named, but the name should not be taken to hold any particular semantic significance.

The following diagram shows a simplistic example.

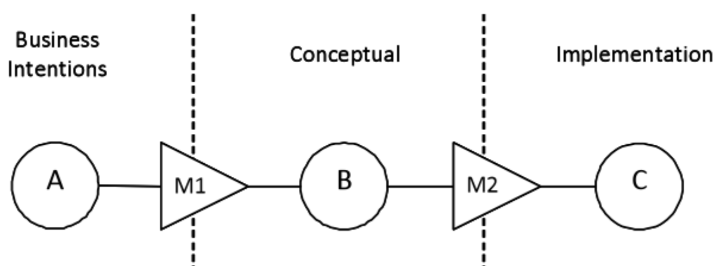


Fig. 1. A simplistic group of domain models and mappings

The diagram shows models in three domains. Model A captures features related to the business intentions for a particular information system. An informal description of such a domain is given in the OMG Business Motivation Model standard [13]. Model B contains conceptual elements related to the business motivation, and is derivable from Model A by the mapping M1. Object Role Modeling provides an example of such a conceptual domain [4]. Model C identifies a specific implementation of the conceptual model, for example, in the form of a relational database system. Model C is derivable from Model B by the mapping M2.

In practice we may find this simplistic approach unsatisfactory. Reasons for needing a more sophisticated approach might include the following

- We may prefer to break a single complex domain (such as "Conceptual") into several simpler domains to make it easier to define and maintain our models.
- We may require models with differing amounts of detail or different notations for different purposes.
- We may want to separate vendor-specific features from vendor-independent features.

Since in our suggested approach we are not limited to any particular number of domains, we can easily accomplish such goals by adding more domains to deal with each specific need. This allows us to select domains that are the most appropriate for the task at hand. For example: we could introduce a mapping from a formal domain – with high precision but low understandability – to a domain using a natural language representation – with low precision but high understandability. This actual approach is used in the NORMA modeling tool for ORM, where logically precise model constructs are mapped to textual descriptions (the process is termed "verbalization" in NORMA). This is also a good example of mapping asymmetry: by and large, a reliable mapping from a logical construct to natural language is straightforward to accomplish, but the reverse is not the case.

In general, a mapping cannot be assumed to take any particular form. For example, the relationships between elements in one domain and elements in another domain are unlikely to be one-to-one. A crucial feature of our approach is that an element in one domain is never the same as an element in another domain, even if they are related by a one-to-one mapping.

It may be convenient to define a mapping from one domain to another via a series of intermediate domains and intermediate mappings, for several reasons.

- It may be simpler to define individual step-by-step mappings than one large complex mapping.
- It may be efficient to re-use mapping steps originally developed for a different mapping chain.
- It may be desirable to carry out some additional operations in intermediate domains, where they are more convenient than in either the source or the target domain.

Mapping defines a relationship between domains, and also implies a set of operations for producing target from source. Here, we're mainly interested in the relationship, not the operations. The mapping from a model in one domain to a model in another domain may use information from a model in a third domain to guide or constrain the mapping. There could be several alternative instantiations of a procedure to map between two domains, each of which instantiations preserves exactly the same relationship. For example, two programs, each of which is written in a different programming language, could carry out the same mapping.

2.4 An Outline Process for Constructing Information Systems

Traditionally, computer-based information systems are developed from user requirements by a series of ad-hoc manual activities to produce an implementation that is intended to satisfy the requirements.. This approach is well known to be error prone, unpredictable and expensive. In addition, it is usually impossible to ascertain whether or not the original intentions expressed in the requirements are actually incorporated into the implementation.

We suggest that this overall process could be improved by considering it as a series of models in different domains linked by appropriate mappings. From this viewpoint, the requirements and the implementation become simply other models linked by relevant mappings. Some mapping steps might remain manual, rather than automated, but this approach should make it possible to provide a more precise description of how the overall process produces its results, and whether those results are the ones intended.

In outline, the overall process would work as follows.

1. Gather information based on stakeholder needs to build source representations in as many domains as are relevant. There may be ways of optimizing this stage in order to minimize cost, time, etc.
2. Transform source domains to target domains, possibly via several intermediate domains, and perhaps with specific operations being carried out in some intermediate domains. Automation should be used wherever feasible.
3. Validate target domains against original stakeholder needs. If discrepancies are discovered, modify source domains appropriately and repeat from 2.

The following diagram shows a theoretical information system development process that illustrates some of the points made above.

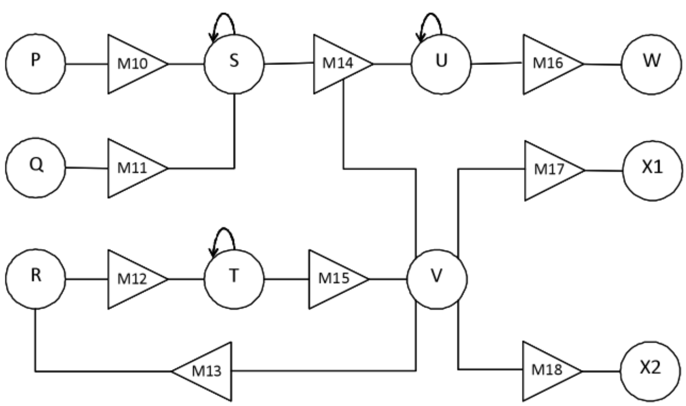


Fig. 2. A theoretical development process

Each model above (represented by circles) is assumed to be in a different domain, except X1 and X2, which are both in Domain X. Each of the mappings, denoted by M*n* are assumed to relate one source domain to one target domain (a one-to-many mapping could always be decomposed into separate one-to-one mappings). Models P, Q and R contain information that express stakeholder concerns. Model S is composed from two of these models. Mapping M14 uses information from Model V to guide the mapping from Model S to Model U. Mapping M13 provides a "round-trip" path from Model V to Model R, but it is unlikely that the source Model R could be reconstructed from a Model V by this route. Models W, X1 and X2 represent implementation artifacts. Models X1 and X2 are in the same domain, Model W is in a different domain. Operations are carried out on intermediate models S, T and U as part of the process. A more realistic example may have many more domains and mappings, but the above should give the general flavor of the approach.

In general, knowledge of source models and mappings alone will not provide reliable forward traceability between a source model and a target model (such as P and W in the diagram above) because of operations carried out on intermediate models (such as S and U in the diagram above). The situation is even worse for reverse traceability, because mappings usually do not preserve source information. A reverse mapping, such as M13 in the example above, can only hope to provide partial information (for example, to update some aspects of Model S to take account of the effects of operations carried out on Model T). The only approach likely to provide traceability in either direction is to record the operations carried out: both inter-model (by mapping processes) and intra-model (by local operations). To make traceability effective, it is obviously essential that these records are strictly maintained. This implies automated recording, because manual recording is unlikely to be sufficiently reliable. Implementing such functionality is not as difficult as it may seem, as we can see by comparison with equivalent functions in other contexts.

- Many complex programs carry out similar recording actions to support an "undo" facility. A "trace" facility would require different types of records and persistent storage, but might be simpler, since we don't want to undo operations: we simply want to record them.
- Relational databases routinely maintain a log of transactions that can be used to rebuild a database in case of system failure. A "trace" facility would require similar logging, but could avoid many of the complications for concurrency and isolation required in databases.

Given the existence of such records of model operations, tracing model features in either direction becomes a straightforward information-processing task.

3 Unpacking the “Levels” for a More-Adequate Modeling of Business Constraints

Let us return to the frameworks which we enumerated in Sec. 2.2: Despite the proliferation of such frameworks, it appears, from anomalies noted in [1], that the modeling (i.e. structured expressing) of business constraints according to such “levels”, and use of such models and structures to relate the business’s concerns to the design of the information system, have been inadequate to support the needed traceability, or even identification and understandability, of the rules. As mentioned in Sec. 1, SBVR’s categorization of business constraints gave rise to some apparently intractable paradoxes. But these paradoxes are, in fact, endemic to a particular flaw that is characteristic of most of these frameworks, as far as they are used for modeling of business constraints. This flaw has been dominant at least as far back as 1982 and the ISO technical paper, "Concepts and Terminology for the Conceptual Schema and the Information Base" [15].

The flaw's main feature is the idea that the implemented constraints and derivation rules (if any) in an information schema are, quite simply, models (i.e., expressions) of rules that structure the UoD existing outside the information system. Thus the overall information system, and not just its fact-type structures and their data-population, is a formal model of the business domain. Although this simple assumption seems plausible on the surface, not only does it naïvely conflate business concerns with technical features, but it thereby hampers traceability between information systems and business needs and leads to various paradoxes such as we have mentioned.

Having stated the faulty assumption fairly simply, let us now unpack it a bit, to see where it goes wrong and what is needed to repair it. The apparent anomaly in the approach we are analyzing is that it assumes that the business may write business rules about anything in its purview, but not about its own information system. After all, it assumes the rule-implementations (the actual "constraints", in the technical sense, triggers, etc.) are all expressions of rules that pertain to business objects outside the record-keeping system. This assumption leaves therefore no scope for any rules the business wishes to express and enforce about the management of the record-keeping system itself. It leaves the business lacking ability to enforce its wishes inside its own system.

For example, suppose the business does not want the record-keeping system ever to record more than one gender (in any state of the system) per customer, given the assumed alethic necessity that, in this domain, no customer will ever have more than one gender at a time. The above, over-simplistic framework-assumption dictates that the "business rule" being enforced by the constraint (be the latter a uniqueness constraint, trigger, or what have you) in the system is that alethic necessity, rather than the business's simple wish not to keep faulty data. That conclusion leads, as mentioned, to the paradox of an alleged attempt to enforce a rule that could never possibly be violated—with or without our attempts at "enforcement". So, either the business would enforce that which needs no enforcement, or it would leave unenforced that which needs enforcement. Either way, the framework would clearly be hampering the understanding or the management of the rule, or both.

Thus, a more-adequate framework must at the very least avoid this over-simplistic assumption. Rather than assuming that all rules implemented in the information system are about things outside the management of record-keeping, it should provide for modeling of rules concerning the record-keeping system itself, and support the capture of any logical, mapping-connections linking them with other rules.

One way to structure a framework in order to support these concerns is to split into two the lone level usually afforded to the capturing of "conceptual" business concerns. That is, we could have one "conceptual" level for capture of business constraints about things external to the record-keeping system, and a separate "conceptual" level for constraints about the performance or management of the record-keeping system itself. (See Table 1.)

Table 1. A better framework for understanding and management of business constraints

Business intentions	
Conceptual-external: Rules (verbalizeable as) about things, events, or processes *outside* the record-keeping system	
Conceptual-internal: Rules (verbalizeable as) about things, events, or processes *inside* the record-keeping system	
Logical-internal design	
Physical-internal design	
...	

We should emphasize that both these "levels" (shown with bold borders in the table) capture concerns of the business; thus, they are both "conceptual", in the sense that this term is used in these contexts. For lack of better terms, we have called them "conceptual-external" and "conceptual-internal" (i.e., "external" and "internal" to the record-keeping system). Additionally, all of these business concerns exist independently of any predicted or actual computerization of the record-keeping system. Finally, it is important to remember that the relative placement of these two "levels" in a tabular depiction of this framework is basically arbitrary: Neither of these levels is inherently "higher" or "lower" than the other, even on a scale of abstraction away from (or towards) implementation-level concerns. And as we emphasized earlier, we see "levels" as domains, and domains as peers by default.

In any case, once one adds the simple distinction between these two levels, one can easily unravel all the otherwise intractable paradoxes and the hampering of one's understanding and/or management of business constraints. We will now work through some examples to illustrate this.

4 Business Constraints and Domains in This Framework

The business constraints we used for illustration in the previous section typify many kinds of rule pertinent to information-system design. A set of categories adequate to these and other examples can be contrived that can predict, for any correctly-categorized rule, not only the sort of verbalization it should have (as to whether it should speak as if about records in the system, or about things outside the record-keeping system); it can also predict what logical, mapping-connections the rule will have to what other rules of concern to the business. Thus it can give strong support to our desired traceability between organizational needs and how those needs are being satisfied. Let us contrive now such a set of categories, localize each of these categories within the framework, and show logical connections that naturally obtain between various rules categories.

First, let us take a common type of constraint implemented in information systems, the so-called alethic constraint. As indicated before, a rule of this sort is based on, rather than identical with, an alethic necessity pertaining to things outside the record-keeping system. For example: each customer has exactly one gender at any given time. This rule is not something that needs enforcing; it is an alethic necessity. So the business would never express and enjoin this as something it wants observed. “Alethic business rule”, taken in this sense, is an oxymoron; no such rule is a concern of the business. Rather, that which it wishes to be observed, and if possible enforced, is the rule that not more than one gender will be recorded, at any time, as the current gender of any given customer. In short, the rule of concern to the business is a rule about the record-keeping, not a rule about the gender-having.

The value of distinction between those two rules might be made clearer by a different example. The business is probably aware of the alethic necessity that every customer has some gender (in every state of the UoD). However, that does not imply at all that the record-keeping system should reflect that necessity. That is, it does not at all imply that the business would or should want to enforce the recording of every customer who is entered in the records. The decision to enforce this, if it happens, will be made entirely at the discretion of the business.

Clearly then, this rule inhabits one of the “conceptual” levels in our framework, but only one. Specifically, it inhabits the level pertaining to rules about the record-keeping system. Thus, contrary to a common assumption, the rule of concern here should be verbalized not as a necessity that “no customer has more than one gender”, but only as a (deontic) rule about the record-keeping, viz. that “no customer should be recorded as having more than one gender”. If the former rule, the alethic necessity, inhabits the framework at all, it inhabits it only in the form of metadata attaching to the latter rule (for explaining the situational background of that business constraint).

What about a rule like, “No more than 30 students shall be enrolled in any class-offering”? This rule is clearly about things outside the record-keeping system (viz., students and class-offerings). However, it may have implications also for the record-keeping system. This depends on whether the record-keeping system is itself the mechanism by which enrollments are effected. If so, then a different but logically-related rule is of interest to the business, viz. that “no more than 30 students shall be recorded as enrolled in any class-offering”. As noted in [1], this situation is due to (what speech-act theorists sometimes call) the self-referentially performative nature of the enrollment-records (they effect that state of affairs which they record).

Here the two rules are clearly distinct, since they are semantically different in propositional content: one speaks only of enrollments, and the other only of recordings of enrollments. However, though non-equivalent semantically, clearly the two rules are, in a sense, equivalent logically: the business will enjoin the one rule if and only if it will enjoin the other. In short, in a case of self-referentially performative records, rules come in pairs, of distinct but logically-related rules: one about some states of affairs effectuated by records, and one about the actual keeping of those records.

What about a rule regarding the derivation of the category of something in the business? For example, should we categorize Customer Y as a “Gold Customer”, or not? Here, the initial rule is a stipulation, by the business, of what conditions shall be necessary and sufficient to qualify someone as a Gold Customer. Such rules are commonly called “stipulative definitions”. Since they are about things outside the record-keeping system (viz. the use of terms, and characteristics that qualify something as an X or a Y), they go in the “conceptual-external” level we’ve defined.

However, this is not to say that it has no direct implications for implementation of the automated categorization of (some) customers as “Gold”. For example, a stipulative definition might map to some feature of the logical-internal information schema, such as a “derived column” that is virtual rather than stored.

This latter sort of relationship between rules at different levels is not exactly that of logical equivalence. Rather, the relationship is somewhat looser: if we have a constraint implemented at such a computation-dependent level, it represents the desires of someone, but it may just be the desires of the database programmer. That is, the desired traceability back to some business concern may not be there. But such a concern, if indeed captured at some computation-independent “level”, would “map to”, i.e. would imply, a “lower-level” (i.e. computation-dependent-level) constraint, assuming the record-keeping system is automated and programmable and supports the programming or automating of that particular rule. The rule that maps to this implementation is usually about the record-keeping system; but sometimes, as in this example, it maps from the other conceptual level, about things outside the record-keeping system.

Also, note that any rule to “derive-and-store” the categorization of customers, i.e. to keep such facts stored and updated on any change to the customer-qualifying facts, is a rule distinct and separate from the stipulative definition of “Gold Customer”. And since it is a rule about the desired operation of our record-keeping system, it would inhabit our “conceptual-internal” framework-level.

Some other rules pertinent to information-system design have regard to the primary scheme for referring to some particular sort of business object. For example, a business constraint might enjoin that no more than one EmpNr (employee-number) be assigned to each employee. Another might enjoin that in a communication regarding any employee(s), the latter will be referred to by their EmpNr (and not by any alternative unique identifier such as their Social Security number). Both of these rules have regard primarily to things outside the record-keeping system. But the latter rule has also some implication(s) for the proper management of the record-keeping system; so it has a corollary rule inhabiting our “conceptual-internal” domain.

A business might have rules about its record-keeping system that are not based on any of the above considerations. For example, a business could try to support the recording of an unlimited quantity of contact phone-numbers for customers; but it might decide instead to save storage space (and perhaps query-overhead) by recording at most one contact phone-number per customer. Such a rule is not based on any alethic necessity, stipulative definition, self-referential record-keeping, or rules about reference schemes; but it is nonetheless the business’s prerogative to make such a decision. Obviously this rule would inhabit the “conceptual-internal” level shown.

Thus we may locate certain types of rules within the two “conceptual” levels. These types and their locations are depicted in Table 2. In this table, mapping-connections between rules of these categories and rules inhabiting other, computation-dependent domains within the framework have been abbreviated to asterisks (*). On the other hand, mapping-connections between the various rule-categories located in either of the “conceptual” levels have been depicted by arrows between them.

Table 2. A populated framework for management of business constraints

Business intentions	
Conceptual-external: Rules (verbalizeable as) about things, events, or processes *outside* the record-keeping system	**RS, *D1, SRP1, PRS1**
Conceptual-internal: Rules (verbalizeable as) about things, events, or processes *inside* the record-keeping system	***AL, *D2, *SRP2, *PRS2, *RR**
Logical-internal design	
Physical-internal design	
...	

*	rules potentially mappable to some feature in the logical-internal schema
->	logical implication of the existence of one type of rule by existence of another
RS	rules about a mapping between business objects and values used to reference them
SRP 1	rules about "something done by way of" self-referentially performative records in a database
SRP 2	rules about the keeping of some self-referentially performative records
AL	data-integrity rules arising from certain alethic necessities in the UoD
PRS	rules about what is the primary reference scheme for a business object
D1	stipulative definitions
D2	rules to derive-and-store (according to some specific D1-type rule)
RR	any other rules about the management of the record-keeping system

5 Value of Such a Categorization-Scheme

We have now found a set of rule-categories that locates a rule within this expanded, rule-management framework, and shows its natural logical connections with, and even entailments of, rules of other sorts and at other levels. This categorization scheme for rules is clearly valuable for its support of traceability, via its indicating trace-paths that travel through the natural logical connections between rules at the computation-dependent levels and those at other "levels".

But there is also some value in the very limitations it applies to what types of rule there are that are relevant for information system design, and in the guidance it gives on exactly which sorts of rule are relevant, and on how they are relevant. It shows in what ways the implemented constraints, in the computation-dependent levels, should be relatable to rules at what other levels. This provides a strong source of guidance, inasmuch as constraints that cannot be traced back through these logical links are probably not traceable at all back to any business intentions or needs.

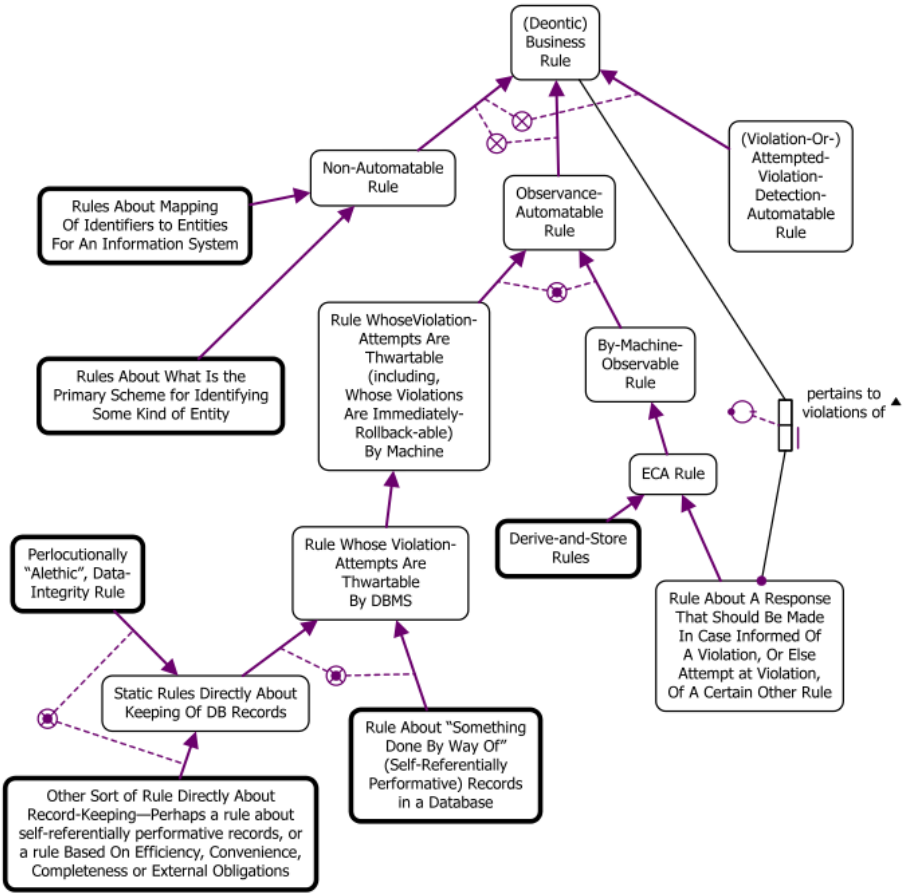


Fig. 3. A fuller categorization scheme for business constraints

The point is that applying a rule categorization scheme such as this forces the system designers and implementers to validate each constraint they put into the system, in terms of business concerns or intentions. In this way, many intuitively practical, but actually pointless, rule-implementations might be avoided.

A fuller rule-categorization scheme, including rules of kinds not implementable, or not pertinent to information system design, is shown in Fig. 3. This diagram uses ORM2 notation [4], except that those rule subtypes pertinent to information systems have been shown with a bold border.

6 Conclusions

An over-simplistic approach to framework construction may have inspired a common assumption that has hampered both the intelligible modeling of business constraints, and the traceability of concerns between different constraints, from initial business constraints to implementations in an information system. This paper has set forward a more generic approach to framework construction, and has illustrated its power by showing how a more generic and flexible framework than is usual for modeling business constraints leads to resolution of the paradoxes and the traceability issues that have hampered more-rigid frameworks.

References

1. Carver, A., Morgan, T.: Characterizing Business Rules for Practical Information Systems. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 443–452. Springer, Heidelberg (2011)
2. US Senate: Public Company Accounting Reform and Investor Protection Act, Pub.L. 107-204, 116 Stat. 745, enacted July 30 (2002)
3. The Object Management Group (OMG): OMG Model Driven Architecture, http://www.omg.org/mda/
4. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann (2008)
5. Zachman International: A Concise Definition of The Zachman Framework (2008), http://www.zachman.com/about-the-zachman-framework
6. International Standards Organization (ISO): Information technology - Open Distributed Processing - Reference model (RM-ODP), ISO/IEC 10746 (1998)
7. National Association of State Chief Information Officers (NASCIO): Enterprise Architecture Development Tool-Kit v3.0 (2004), http://www.nascio.org/resources/EAresources.cfm2004
8. The Object Management Group (OMG): The Unified Modeling Language (UML) Version 2.4.1, http://www.omg.org/spec/UML/2.4.1/
9. International Standards Organization (ISO): Systems and software engineering - Architecture description, ISO/IEC/IEEE 42010 (2000)
10. The Open Group: The Open Group Architecture Framework, TOGAF (2011), http://www.opengroup.org/togaf/
11. The US Department of Defense: DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines (DoDAF) (2007), http://dodcio.defense.gov/docs/DoDAF_Volume_I.pdf
12. The UK Ministry of Defence: MOD Architecture Framework (MODAF) V1.2.004 (2010), http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/
13. The Object Management Group (OMG): The Business Motivation Model (BMM), Version 1.1 (2010), http://www.omg.org/spec/BMM/1.1/
14. The Object Management Group (OMG): Semantics Of Business Vocabulary And Rules (SBVR) Version 1.0 (2008), http://www.omg.org/spec/SBVR/1.0/
15. International Standards Organization (ISO): Concepts and Terminology for the Conceptual Schema and the Information Base, ISO TC97/SC5/WG3 (1982)
16. Merriam-Webster Dictionary, http://www.merriam-webster.com/

Perspectives to Process Modeling – A Historical Overview

John Krogstie

Norwegian University of Science and Technology
krogstie@idi.ntnu.no

Abstract. Processes modeling is done for a number of reasons in relation to enterprise modeling, business process modeling and information systems development in general, and this paper will give an overview of main approaches to different types of process modeling. Modeling approaches are structured according to the main modeling perspective being used. In conceptual modeling in general, one can identify 8 modeling perspectives; behavioral, functional, structural, goal-oriented, object-oriented, language action, organizational and topological. In the paper we will present both historical and current examples of process modeling according to these different perspectives, and discuss what perspectives are most appropriate to achieve the different goals of modeling.

Keywords: Process modeling, conceptual modeling.

1 Introduction

A *process* is a collection of related, structured tasks that produce a specific service or product to address a certain goal for a particular actor or set of actors. Process modeling has been performed relative to IT and organizational development at least since the 70ties. The interest has gone through phases with the introduction of different approaches, including Structured Analysis in the seventies [33], BPR in the late eighties/early nineties [42], and Workflow Management in the nineties [95]. Lately, with the proliferation of BPM (Business process management) [46], interest and use of process modeling has increased even further, although focusing primarily on a selected number of modeling approaches.

Models of work processes have long been utilized to learn about, guide and support practice also in a number of areas. In software process improvement [22], enterprise modeling [32] and quality management, process models describe methods and standard working procedures. Simulation and quantitative analyses are performed to improve efficiency [7, 61]. In process centric software engineering environments [9] and workflow systems [95] model execution is automated. This wide range of applications is reflected in current modeling languages, which emphasize different aspects of the process.

The archetypical way to look on processes is as a transformation, according to an IPO (input-process-output) approach. Whereas early process modeling approaches had this as a basic approach [33], as process modeling have been integrated with other types of conceptual modeling, variants of this have appeared.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 315–330, 2012.

First we describe different reasons for doing process modeling. Then we describe different perspectives to modeling, before we in section 4 provide a brief overview of modeling languages used for process modeling following the different perspectives. Since many of those languages being used in practice are developed a long time ago [20] or are extensions of these, we provide also a partly historical overview. In the conclusion we briefly summarize how modeling according to the different perspectives is beneficial to achieve the various goals of modeling. Since the different goals of modeling require different properties from the modeling language used, it is useful to look more closely on the properties of different modeling perspectives to be able to choose an appropriate modeling approach. Due to size limitation of this paper, this overview will only be on a high level.

2 Application of Process Modeling

According to general model theory [87] there are three common characteristics of models: *Representation, Simplification* and *Pragmatic orientation.*

- *Representation*: Models are models of something else
- *Simplification*: Models possess a reductive trait in that they map a subset of attributes of the phenomenon being modeled
- *Pragmatic orientation*: Models have a substitutive function in that they substitute a certain phenomenon as being conceptualized by a certain subject in a given temporal space with a certain incentive or operation in mind

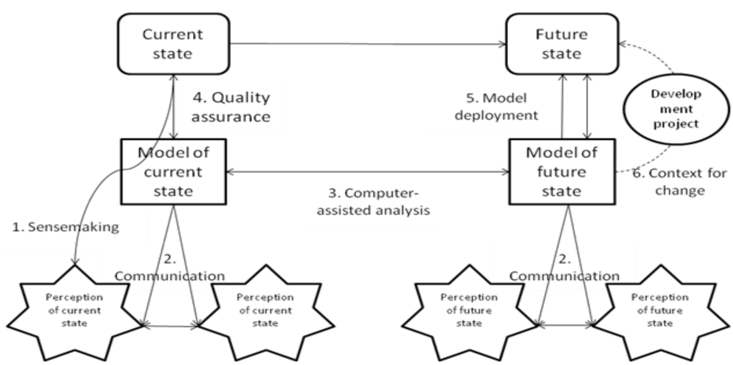


Fig. 1. Organizational application of modeling

Process modeling is usually done in some organizational setting. As illustrated in Fig. 1 one can look upon an organization and its information system abstractly to be in a state (the current state, often represented as a descriptive 'as-is' model) that are to be evolved to some future wanted state (often represented as a prescriptive 'to be' model). Obviously, changes will happen in an organization no matter what is actually planned, thus one might in practice have the use for many different models and scenarios of possible future states, but we simplify the number of possible future states in the discussion below.

The state includes the existing processes, organization and computer systems. These states are often modeled, and the state of the organization is perceived (differently) by different persons through these models. Different usage areas of conceptual models as described in [60, 73]:

1. Human sense-making: The descriptive model of the current state can be useful for people to make sense of and learn about the current perceived situation.
2. Communication between people in the organization: Models can have an important role in human communication. Thus, in addition to support the sense-making process for the individual, a model can act as a common framework supporting communication both relative to descriptive and prescriptive models.
3. Computer-assisted analysis: This is used to gain knowledge about the organization through simulation [6] or deduction, often by comparing a model of the current state and a model of a future, potentially better state.
4. Quality assurance, ensuring e.g. that the organization acts according to a certified process developed for instance as part of an ISO-certification process.
5. Model deployment and activation: To integrate the model of the future state in an information system directly, making the prescriptive model the descriptive model. Models can be activated in three ways:
   a. Through people, where the system offers no active support.
   b. Automatically, where the system plays an active role, as in most automated workflow systems.
   c. Interactively, where the computer and the users co-operate [56].
6. To be a prescriptive model to be used in a traditional system development project, without being directly activated.

3 Perspectives to Modeling

Modeling languages can be divided into classes according to the core phenomena classes (concepts) that are represented and focused on in the language. This has been called the *perspective* of the language [60, 62]. Languages in different perspectives might overlap in what they express, but emphasize different concepts as described below. A classic distinction regarding modeling perspectives is between the structural, functional, and behavioral perspective [74]. Object-orientation analysis appeared as a particular way of combining the structural and behavioral perspective in the late eighties.

Through other work, such as [19], [70], F3 [15], NATURE [51], [57] additional perspectives have been identified, including goal, actor, communicational, and topological. To provide a broad overview of the different perspectives conceptual modeling approaches accommodate, we look on the following:

Behavioral Perspective: Languages following this perspective go back to the early sixties, with the introduction of Petri-nets [79]. In most languages with a behavioral perspective the main phenomena are 'states' and 'transitions' between 'states'. State transitions are triggered by 'events' [21].

Functional Perspective: The main phenomena class in the functional perspective is the 'transformation': A transformation is defined as an activity which based on a set of phenomena transforms them to another set of phenomena.

Structural Perspective: Approaches within the structural perspective concentrate on describing the static structure of a system. The main construct of such languages is the 'entity'.

Goal and Rule Perspective: Goal-oriented modeling focuses on 'goals' and 'rules'. A rule is something which influences the actions of a set of actors. A rule is either a rule of necessity or a deontic rule [58, 75]. A rule of necessity is a rule that must always be satisfied. A deontic rule is a rule which is only socially agreed among a set of persons and organizations. In the early nineties, one started to model so-called rule hierarchies, linking rules of different abstraction levels.

Object-Oriented Perspective: The basic phenomena of object oriented modeling languages are similar to those found in most object oriented programming languages; 'Objects' with unique id and a local state that can only be manipulated by calling methods of the object. Objects have a life cycle. The process of the object is the trace of the events during the existence of the object. A set of objects that share the same definitions of attributes and operations compose an object class.

Communication Perspective: The work within this perspective is based on language/action theory from philosophical linguistics. The basic assumption of language/action theory is that persons cooperate within work processes through their conversations and through mutual commitments taken within them.

Actor and Role Perspective: The main phenomena of languages within this perspective are 'actor' (also termed agent) and 'role'. The background for modeling in this perspective comes both from organizational science, work on programming languages, and work on intelligent agents in artificial intelligence.

Topological Perspective: This perspective relates to the topological ordering between the different concepts. The best background for conceptualization of these aspects comes from the cartography and CSCW fields, differentiating between space and place [28, 45]. 'Space' describes geometrical arrangements that might structure, constrain, and enable certain forms of movement and interaction; 'place' denotes the ways in which settings acquire persistent social meaning through interaction.

4 Perspectives to Process Modeling

We here provide a very brief overview of process modeling according to the different modeling perspectives identified in section 3 above.

4.1 Process Modeling According to the Behavioral Perspective

States (of systems, products, entities, processes) and transformations between states are the central concepts in this perspective. There are two language-types commonly used to model states: State transition diagrams (STD) and state transition matrices (STM). The vocabulary of state transition diagrams is

- State: A system is always in one of the states in the lawful state space for the system. A state is defined by the set of transitions leading to that state, the set of transitions leading out of that state and the set of values assigned to attributes of the system while the system resides in that state.
- Event: An event is a message from the environment or from system itself to the system. The system can react to a set of predefined events.
- Condition: A condition for reacting to an event.
- Transition: Receiving an event will cause a transition to a new state if the event is defined for the current state, and if the condition assigned to the event evaluates to true.
- Action: The system can perform an action in response to an event.

It is generally acknowledged that a large complex system cannot be described in a flat state-model, because of the unmanageable, exponentially growth of states. Hierarchical abstraction mechanisms were added to traditional STDs in Statecharts [43]. Statecharts are integrated with functional modeling (see below) in [44]. Later extensions of Statecharts for object-oriented modeling were developed through the nineties, and Statecharts are the basis for the state transitions diagrams in UML (for the modeling of object-states) [14].

Petri-nets [79] are another well-known behavior-oriented modeling language. Here, *places* indicate a system state space, and a combination of *tokens* located in the places determines the specific system state. State transitions are regulated by firing rules: A transition is enabled if each of its input places contains a token. A transition can fire at any time after it is enabled. The transition takes zero time. After the firing of a transition, a token is removed from each of its input places and a token is produced in all output places. Control-flow aspects like precedence, concurrency, synchronization, exclusiveness, and iteration can be modeled in a Petri-net. There exists several dialects of the Petri net language (going back to [67]) where the transitions are allowed to take time, and these approaches provide decomposition in a way not very different from that of a data flow diagram. Timed Petri Nets [67] also provide probability distributions that can be assigned to the time consumption of each transition and are particularly suited to performance modeling. Other variants are tokens with named and typed variables (Colored Petri Nets), and nets where transitions have pre- and post-conditions in some logic. Colored Petri nets are used in particular for simulation and analysis [52].

Another type of behavioral modeling is based on System dynamics. Systems thinking [85] regards causal relations as mutual, circular and non-linear, hence the straightforward sequences in transformational process models is seen as an idealization that hides important facts. This perspective is also reflected in mathematical models of interaction [93]. System dynamics have been utilized for analysis of complex relationships in cooperative work arrangements [7]. System dynamic process models can be used for analysis and simulation, but not for model activation. A challenge is that it can be difficult to find data to run simulations.

4.2 Process Modeling According to the Functional Perspective

Most popular process modeling languages take a functional (or transformational / input-process-output) approach [20]; although some of the most popular recent languages also include behavioral aspects as will be discussed below. Processes are often divided into activities, which may be divided further into sub-activities. Each activity takes inputs, which it transforms to outputs. Input and output relations thus define the sequence of work. This perspective is chosen for the standards of the Workflow Management Coalition [95], the Internet Engineering Task Force (IETF) [13] as well as most commercial systems [30]. IDEF-3x [50] and Data Flow Diagram (DFD) [33] are paradigm examples of this. DFDs describe a situation using: Processes, data stores, flows, and external entities.

When a process is decomposed into a set of sub-processes, the sub-processes are co-operating to fulfill the higher-level function. This view on DFDs has resulted in the "context diagram" that regards the whole system as a process which receives and sends all inputs and outputs to and from the system. A context diagram determines the boundary of a system. A variant of context-diagrams is Use Case diagrams [14].

DFD and use-cases are semi-formal languages. Some of the short-comings of DFD regarding formality were first addressed in the transformation schema presented by Ward [92] including both data and control transformations, data and event flows (signals, activation and deactivation) (data flows being either discrete or continuous) and variants of stores. A number of the recent process modeling notations typically add control-flow aspects to a transformational approach and combine aspects of the functional and behavioral perspectives. Some examples of this are ARIS EPC, UML Activity Diagrams, YAWL [90], and BPMN.

An Event-driven Process Chain (EPC) [54] is a graphical modeling language used for business process modeling. EPC was developed within the framework of Architecture of Integrated Information System (ARIS) [81] to model business processes. The strength of EPC lies on its simple notation that is capable of portraying business information system while at the same time incorporating other important features such as functions, data, organizational structure, and information resources. However the semantics of an event-driven process chain are not well defined and it is not possible to check the model for consistency and completeness. As demonstrated in [3], these problems can be partly addressed by translating EPC-models to Petri nets since Petri nets have formal semantics enabling analysis techniques.

The UML Activity diagram is one of the three diagram types in the UML for modeling behavior aspect of systems [14]. The most important concepts in the UML activity diagram are activities, decision, start (split) or end (join) of concurrent activities, and start and end states

In 2004, BPMN was presented as the standard business process modeling notation [96]. Since then BPMN has been evaluated in different ways by the academic community [1, 80] and has become widely supported in industry.

The Business Process Modeling Notation (BPMN version 1.0) was adopted by OMG for ratification in February 2006. The BPMN 2.0 specification was formally released January 2011 [76].

Given the extensive use of functional languages, a number of analyses focus on this category [18, 19, 40]. The expressiveness of these languages typically includes decomposition, and data flow, while organizational modeling and roles often are integrated and given less emphasis. In approaches which integrate behavioral and functional aspects, we see also a support for control flow. Aspects like timing and quantification, products and communication, or commitments are better supported by other perspectives. User-orientation is a major advantage of transformational languages, in particular the pure functional ones. Graphical input-process-output models are comprehensible given some training, but you can also build models by simply listing the tasks in plain text, or in a hierarchical work breakdown structure.

4.3 Process Modeling According to the Structural Perspective

The structural perspective has traditionally been handled by languages for data modeling, but also includes approaches from semantic networks and the semantic web. In ER-modeling as described by [17], the basic components are:

- Entities. An entity is a phenomenon that can be distinctly identified. Entities can be classified into entity classes
- Relationships. A relationship is an association among entities. Relationships can be classified into relationship classes
- Attributes and data values. To give value to a property of an entity or relationship. Values are grouped into value classes by their types.

Structural modeling is often perceived to be fundamentally different from functional (process) modeling, since it focus on the static aspects, whereas process modeling focus on dynamics. It is possible to look at processes as entities though (like one have done in object-oriented process modeling discussed below, looking at the process instances as the objects) it which case one can model the situation in a similar way as when doing more traditional data-modeling.

One finds very few attempts on pure structural process modeling in practice, although as we will discuss below, there are approaches to object-oriented process modeling.

4.4 Process Modeling According to the Goal and Rule Perspective

In the workflow area, the use of rules for guiding the workflow is often termed declarative workflow. Constraint based languages [27, 35] prescribe a course of events, rather they capture the boundaries within which the process must be performed, leaving the actors to control the internal details. Instead of telling people what to do, these systems warn about rule violations and enforce constraints. Thus, problems with over-serialization can be avoided [35].

A wide variety of declarative modeling approaches has been specified in business process management, from the use of basic Event-condition-action (ECA)-rules [53] to declarative process modeling languages such as DecSerFlow [4], BPCN [66] and ConDec [78]. In [36] an overview of the most common declarative process modeling languages can be found.

Several advantages have been experienced with a declarative, rule-based approach to information systems modeling [59], but also a number of challenges. Languages representing rule-based process modeling can potentially provide a higher expressiveness than diagrammatic languages (e.g. the ability to specify temporal requirements) [66], but this might result in process models which are less comprehensible [29].

Declarative process enactment guarantees high run-time flexibility for declarative process specifications that contain only the strictly required mandatory constraints. An individual execution path that satisfies the set of mandatory constraints can be dynamically built for a specific process instance. Process compliance is assured when all mandatory rules are correctly mapped onto mandatory business constraints. During the construction of a suitable execution path little support is provided to the end user [94], which could affect the process effectiveness. In [58] differentiating constraints by modality is proposed, recommendations were introduced to guide the user whereas obligations would ensure compliant behavior. Lastly, the increased size and complexity of contemporary process models might decrease the potential for process automation since current declarative workflow management systems might have limited efficiency in when having to take into account a large number of rules according to [5].

A graphic depiction is difficult since it would correspond to a visualization of several possible solutions to the set of constraint equations constituting the model. The support for articulation of planned and ongoing tasks is limited. Consequently, constraints are often combined with transformational models [27, 55, 63]. Alternatively one can have the operational rules related to the process model also linked to goal hierarchies as in [58, 59].

4.5 Process Modeling According to the Object-Oriented Perspective

UML [14] has become both the official and de facto standard for object oriented analysis and design. Consequently, people also apply UML to model business processes. Object orientation offers a number of useful modeling mechanisms like encapsulation, polymorphism, subtyping and inheritance [64, 71]. UML integrates these capabilities with e.g. requirements capture in use case descriptions as described above and behavior modeling in state, activity and sequence diagrams. On the other hand, UML is designed for software developers, not for end users. A core challenge thus remains in mapping system-oriented UML constructs to user- and process-oriented concepts [47]. To this problem no general solution exists [64]. One approach which is somewhat similar to how one can use structural modeling for process modeling is PML [10]. Here one uses object oriented techniques based on looking upon classes in a particular way. Whereas a class is defined by <class name, attributes, methods>, in PML one define this as <process name, methods, resources>. The PML process class describes the process in a generic way. It allows one to define all methods with assurances and resources needed for the process. The instantiation of a process is a project. This means, the instance of a process defines the current occurrence of resources, used data models etc. Regarding connections and dependencies between single process classes, PML features the standard UML-mechanisms of inheritance and associations.

Although with intriguing possibilities, it is safe to say that full-fledged OO process modeling has yet to be taken into use in large scale in practice.

4.6 Process Modeling According to the Communication Perspective

The communication perspective, often termed the language action perspective was brought into the workflow arena through the COORDINATOR prototype [97], later succeeded by the Action Workflow system [69]. This perspective is informed by speech act theory [82], which extends the notion that people use language to describe the world with a focus on how people use language for coordinating action and negotiating commitments. Habermas took Searle's theory as a starting point for his theory of communicative action [41]. Central to Habermas is the distinction between strategic and communicative action. When involved in strategic action, the participants try to achieve their own private goals. When they cooperate, they are only motivated empirically to do so. When involved in communicative action, the participants are oriented towards mutual agreement. The motivation for co-operation is thus rational. Illocutionary logic [26, 84] is a logical formalization of the theory of Searle. The main parts of illocutionary logic are the illocutionary act consisting of three parts, illocutionary context, illocutionary force, and propositional context. The context of an illocutionary act consists of five elements: Speaker, hearer, time, location, and circumstances. The illocutionary force determines the reasons and the goal of the communication. The central element of the illocutionary force is the illocutionary point, and the other elements depend on this. Five illocutionary points are distinguished [83]: Assertives, Directives, Commissives, Declaratives, Expressives

Speech act theory is the basis for modeling of workflow as coordination among people in Action Workflow [69]. The main strength of this approach is that it facilitates analysis of the communicative aspects of the process. It highlights that each process is an interaction between a customer and a performer, represented as a cycle with four phases: preparation, negotiation, performance and acceptance. The dual role constellation is a basis for work breakdown, e.g. the performer can delegate parts of the work to other people. This explicit representation of communication and negotiation, and especially the structuring of the conversation into predefined speech act steps, has also been criticized [16, 23, 88]. Minimal support for situated conversations, the danger that explication leads to increased external control of the work, and a simplistic one-to-one mapping between utterances and actions are among the weaknesses pointed to. On the other hand, it has been reported that the Action Workflow approach is useful when people act pragmatically and don't always follow the encoded rules of behavior [23], i.e. when the communication models are interactively activated.

Some approaches to workflow modeling combine aspects of both the functional and communicative perspective. In WooRKS [8] functional modeling is used for processes and language action modeling for exceptions. TeamWare Flow [89] on the other hand can be said to be a hybrid approach. In addition to the approach to workflow-modeling described above, several other approaches to conceptual modeling are inspired by the theories of Habermas and Searle such as SAMPO [11], and ABC/DEMO [24, 25].

4.7 Process Modeling According to the Actor and Role Perspective

Role-centric process modeling languages have been applied for work-flow analysis and implementation. Role Interaction Nets (RIN) [86] and Role Activity Diagrams (RAD) [77] use roles as a main structuring concept. The activities performed by a role are grouped together in the diagram, either in swimlanes (RIN), or inside boxes (RAD). The use of roles as a structuring concept makes it very clear who is responsible for what. RAD has also been merged with speech acts for interaction between roles [12]. A newer approach in this direction is S-BPM (subject-oriented business process management [31]).

The role-based approach also has limitations, e.g. making it difficult to change the organizational distribution of work. It primarily targets analysis of administrative procedures, where formal roles are important. The use of swimlanes in BPMN and UML Activity Diagrams described above might also have this effect. Some other approaches worth discussing here are REA and e^{3}Value.

The REA language was first described in McCarthy [68]. It has been developed further in [34]. REA was originally intended as a basis for accounting information systems and focuses on representing increases and decreases of value in an organization. REA has later been extended to apply to enterprise architectures [49] and e-commerce frameworks [91].

The core concepts in the REA language are *resource*, *event* and *agent*. The intuition behind this language is that every business transaction can be described as an event where two agents exchange resources. In order to acquire a resource from other agents, an agent has to give up some of its own resource. It seldom happens that one agent simply gives away a resource to another without expecting another resource back as compensation. Basically, there are two types of events: *exchange* and *conversion* [49]. An exchange occurs when an agent receives economic resources from another agent and gives resource back to that agent. A conversion occurs when an agent consumes resources to produce other resources. REA has influence the electronic commerce standard ebXML.

E^{3}Value [39] is an actor/role oriented modeling language for inter-organizational modeling. The purpose of this modeling language is to represent how actors of a system create, exchange and consume objects of economic value, only including value-adding activities. The modeling language focuses on the key points of a business model, to get an understanding of business operations and systems requirements through scenario analysis and evaluation. The purpose of e^{3}value is to determine whether a business idea is profitable or not, that is to say by analyzing for each actor involved in the system if the idea is profitable for them or not. E^{3}value models give a representation of actors, exchanges, value objects of a business system. Modeling at the actor-level is one approach to address BPM-in-the-large [48].

4.8 Process Modeling According to the Topological Perspective

The concept of place can be related to a process, given that a place focuses on the typical behavior in a certain setting rather than where this is physically. Whereas some processes are closely related to place (e.g. what can be done in a certain, specialized factory), more and more tasks can be done in more or less any setting due to the mobile communication infrastructure, thus making it useful to be able to differentiate geographic/topological from transformation-oriented modeling. In certain representations, aspects of space and place is closely interlinked (e.g. in the representation of the agenda of a conference, also taking time into account). Some approaches letting you take the place into account exists, e.g. work on extending UML activity diagrams with place-oriented aspects [3]. An even more topologically oriented approach is to group concepts at the same location [38].

Traditional representations of space such as a map have to a limited degree been oriented towards representation of process knowledge. Some recent approaches do take these aspects more consciously into account, as exemplified by [72], combining conceptual, temporal, and geographic knowledge representation. Other approaches use the topological perspective more as a metaphor [2].

5 Concluding Remarks

We have summarized this high-level overview of the field, looking upon approaches according to different perspectives relative to the different usage areas for process modeling presented in section 2, and also indicated the amount of actual use of the approach in practice.

Table 1. Usage of modeling perspectives

Area (vs. Fig. 1)	1+2	3	4	5a	5b	5c	6
Perspective (vs. 4.1-4.8)	Sense& Com	An al.	QA	Man. Activa-tion	Work -flow	Inter-active activ.	Req. for ISD
4.1 Behavioral	-/o	+/o	-/-	-/-	+/o	-/-	o/-
4.2 Functional	+/+	-/-	+/o	+/o	-/-	o/-	+/o
4.2 Behavioral + Functional	+/o	+/o	o/-	o/-	+/+	+/o	o/o
4.3 Structural	o/-	-/-	-/-	-/-	-/-	-/-	o/-
4.4 Rule/Goal	-/-	o/-	o/-	-/-	o/-	o/-	o/o
4.5 Object-oriented	-/-	o/-	-/-	-/-	-/-	-/-	o/-
4.6 Communicational	o/-	-/-	-/-	-/-	o/-	o/-	-/-
4.7 Actor	+/o	o/-	o/-	o/-	-/-	-/-	o/o
4.8 Topological	o/-	-/-	-/-	o/-	-/-	o/-	o/-

The legend indicates the applicability of the approach / actual use of the approach (relative to the usage of modeling for this task), '+' indicates good applicability or high use, 'o' is some applicability and use, whereas '-' indicate poor applicability and limited use. E.g. o/- under the communicational perspective for sense-making and communication indicates that it has some applicability for this use, but are very little used in practice. Obviously different approaches according to the same perspective can be more or less applicable, and different languages of a certain perspective would score differently based on the concrete expressiveness and level of formality of the language and modeling approach. Due to space limitations, it is not possible to provide the detailed concrete evaluations of all approaches that we mentioned in the previous section. From the table, we see that functional and combinations of functional and behavioral approaches are used the most. All other perspectives have potential for use for certain areas, although this often varies relative to concrete needs in the domain for representing particular aspects (such as topological aspects which in many cases might not be relevant). In particular some of the less traditional approaches appear to have large untapped potential for a richer more appropriate representation of what we term processes and business processes. We work on a longer article including examples to better illustrate the pros and cons of different approaches to process modeling.

References

1. Aagesen, G., Krogstie, J.: Analysis and design of business processes using BPMN. In: Handbook on Business Process Management. Springer (2010)
2. van der Aalst, W.M.P.: TomTom for Business Process Management (TomTom4BPM). In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 2–5. Springer, Heidelberg (2009)
3. van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41, 639–650 (1999)
4. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a truly declarative service flow language. In: Web Services and Formal Methods, pp. 1–23 (2006)
5. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science-Research and Development 23(2) (2009)
6. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russel, N.: Business process simulation. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1. Springer (2010)
7. Abdel-Hamid, T.K., Madnick, S.E.: Lessons Learned from Modeling the Dynamics of Software Development. Communications of the ACM 32(12) (2000)
8. Ader, M., Lu, G., Pons, P., Monguio, J., Lopez, L., De Michelis, G., Grasso, M.A., Vlondakis, G.: Woorks, an object-oriented workflow system for offices. Technical report, ITHACA (1994)
9. Ambriola, V., Conradi, R., Fuggetta, A.: Assessing Process-Centered Software Engineering Environments. ACM TOSEM 6(3) (1997)
10. Anderl, R., Raßler, J.: Computer-Aided Innovation (CAI), Gaetano Cascini. IFIP, vol. 277. Springer, Boston (2008)
11. Auramäki, E., Hirschheim, R., Lyytinen, K.: Modelling offices through discourse analysis: The SAMPO approach. The Computer Journal 35(4), 342–352 (1992)
12. Benson, I., Everhard, S., McKernan, A., Galewsky, B., Partridge, C.: Mathematical Structures for Reasoning about Emergent Organization. In: ACM CSCW Workshop: Beyond Workflow Management, Philadelphia, USA (2000)
13. Bolcer, G., Kaiser, G.: SWAP: Leveraging the web to manage workflow. IEEE Internet Computing 3(1) (1999)
14. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language: User Guide. Addison-Wesley (2005)

15. Bubenko Jr. J. A., Rolland, C., Loucopoulos, P., DeAntonellis, V.: Facilitating fuzzy to formal requirements modeling. In: Proceedings of the First International Conference on Requirements Engineering (ICRE 1994), April 18-22, pp. 154–157. IEEE Computer Society Press, Colorado Springs (1994)
16. Button, G.: What's Wrong with Speech Act Theory. CSCW 3(1) (1995)
17. Chen, P.P.: The entity-relationship model: Towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
18. Conradi, R., Jaccheri, M.L.: Process Modelling Languages. In: Derniame, J.-C., Kaba, B.A., Wastell, D. (eds.) Promoter-2 1998. LNCS, vol. 1500, pp. 27–52. Springer, Heidelberg (1999)
19. Curtis, B., Kellner, M.I., Over, J.: Process Modeling. CACM 35(9) (1992)
20. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58, 358–380 (2006)
21. Davis, A.M.: A comparison of techniques for the specification of external system behavior. Communications of the ACM 31(9), 1098–1115 (1988)
22. Derniame, J.-C., Kaba, B.A., Wastell, D. (eds.): Promoter-2 1998. LNCS, vol. 1500. Springer, Heidelberg (1999)
23. De Michelis, G., Grasso, M.A.: Situating Conversations within the Language/Action Perspective: The Milan Conversation Model. In: ACM CSCW Conference, Chapel Hill, North Carolina, USA (1994)
24. Dietz, J.L.G.: Integrating management of human and computer resources in task processing organizations: A conceptual view. In: HICCS'27, Maui, Hawaii, US, January 4-7 (1994)
25. Dietz, J.L.G.: Enterprise Ontology – Theory and Methodology. Springer (2006)
26. Dignum, F., Weigand, H.: Communication and deontic logic. In: Wieringa, R., Feenstra, R. (eds.) Working papers of the International Workshop on Information Systems - Correctness and Reuseability, IS-CORE 1994 (1994)
27. Dourish, P., Holmes, J., MacLean, A., Marqvardsen, P., Zbyslaw, A.: Freeflow: Mediating between representation and action in workflow systems. In: ACM CSCW Conference, Boston, USA (1996)
28. Dourish, P.: Re-Space-ing Place: "Place" and "Space" Ten Years. In: Proc. of ACM Conf. Computer-Supported Cooperative Work CSCW 2006, Banff, Canada, pp. 299–308. ACM (2006)
29. Fickas, S.: Design issues in a rule-based system. Journal of Systems and Software 10(2), 113–123 (1989)
30. Fischer, L.: Excellence in Practice IV - Innovation and excellence in workflow and knowledge management. Workflow Management Coalition, Future Strategies, USA (2000)
31. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
32. Fox, M.S., Gruninger, M.: Enterprise modeling. AI Magazine (2000)
33. Gane, C., Sarson, T.: Structured Systems Analysis: Tools and Techniques. Prentice Hall (1979)
34. Geerts, G.L., McCarthy, W.E.: An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. IEEE Intelligent Systems 14, 89–94 (1999)
35. Glance, N.S., Pagani, D.S., Pareschi, R.: Generalized Process Structure Grammars (GPSG) for Flexible Representation of Work. In: ACM CSCW Conference, Boston, USA (1996)
36. Goedertier, S., Vanthienen, J.: An overview of declarative process modeling principles and languages. Com. of Systemics and Informatics World Network 6, 51–58 (2009)

37. Gopalakrishnan, S., Krogstie, J., Sindre, G.: Adapting UML Activity Diagrams for Mobile Work Process Modelling: Experimental Comparison of Two Notation Alternatives. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 145–161. Springer, Heidelberg (2010)
38. Gopalakrishnan, S., Sindre, G.: Diagram Notations for Mobile Work Processes Presented at PoEM, Oslo Norway November 2-3 (2011)
39. Gordijn, J., Yu, E., van der Raadt, B.: e-service Design using i* and e3value, IEEE Software (May- June 2006)
40. Green, P., Rosemann, M.: Integrated Process Modeling: An Ontological Evaluation. Information Systems 25(3) (2000)
41. Habermas, J.: The Theory of Communicative Action. Beacon Press (1984)
42. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business (1993)
43. Harel, D.S.: A visual formalism for complex systems. Science of Computer Programming (8), 231–274 (1987)
44. Harel, D., Lachover, H., Naamed, A., Pnueli, A., Politi, M., Sherman, R., Shtull-Trauring, A., Trakhtenbrot, M.: STATEMATE: a working environment for thedevelopment of complex reactive systems. IEEE TSE 16(4), 403–414 (1990)
45. Harrison, S., Dourish, P.: Re-Place-ing Space: The Roles of Space and Place in Collaborative Systems. In: Proc. CSCW 1996, Boston, MA, pp. 67–76. ACM, New York (1996)
46. Havey, M.: Essential Business Process Modelling. O'Reilly (2005)
47. Hommes, B.-J., van Reijswoud, V.: The quality of business process modelling techniques. In: Conference on Information Systems Concepts, Leiden, Nertherlands. Kluwer (1999)
48. Houy, C., Fettke, P., Loos, P., van der Aalst, W.M.P., Krogstie, J.: BPM-in-the-Large – Towards a higher level of abstraction in Business Process Management. Paper Presented at GISP under WCC 2010, Brisbane, Australia (September 2010)
49. Hruby, P.: Model-Driven Design Using Business Patterns. Springer, New York (2006)
50. IDEF-3x Process Modeling Language Specification, Standard NA-94-1422B, Rockwell International (1993)
51. Jarke, M., Bubenko Jr., J., Rolland, C., Sutcliffe, A., Vassiliou, Y.: Theories underlying requirements engineering: An overview of NATURE at genesis. In: Proceedings of RE 1993, pp. 19–31 (1993)
52. Jensen, K., Kristiansen, L.M., Wells, L.: Coloured petri nets and CPN tools got modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transfer 9(3-4), 213–254 (2007)
53. Kappel, G., Rausch-Schott, S., Retschitzegger, W.: Coordination in workflow management systems a rule-based approach. In: Coordination Technology for Collaborative Applications, pp. 99-119 (1998)
54. Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK) (1992)
55. Krogstie, J., McBrien, P., Owens, R., Seltveit, A.H.: Information Systems Development Using a Combination of Process and Rule Based Approaches. In: Andersen, R., Sølvberg, A., Bubenko Jr., J.A. (eds.) CAiSE 1991. LNCS, vol. 498, pp. 319–335. Springer, Heidelberg (1991)
56. Krogstie, J., Jørgensen, H.D.: Interactive Models for Supporting Networked Organisations. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 550–563. Springer, Heidelberg (2004)
57. Krogstie, J., Opdahl, A., Brinkkemper, S. (eds.): Conceptual Modelling in Information Systems Engineering. Springer (2007)

58. Krogstie, J., Sindre, G.: Utilizing deontic operators in information systems specifications. Requirement Engineering Journal 1, 210–237 (1996)
59. Krogstie, J.: Integrated Goal, Data and Process modeling: From TEMPORA to Model-Generated Work-Places. In: Johannesson, P., Søderstrøm, E. (eds.) Information Systems Engineering From Data Analysis to Process Networks, pp. 43–65. IGI Publishing (2008)
60. Krogstie, J.: Model-based Development and Evolution of Information Systems: A Quality Approach. Springer (2012)
61. Kuntz, J.C., Christiansen, T.R., Cohen, G.P., Jin, Y., Levitt, R.E.: The virtual design team: A computational simulation model of project organizations. Communications of the ACM 41(11) (1998)
62. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises. Springer (2008)
63. Lindland, O.I., Krogstie, J.: Validating Conceptual Models by Transformational Prototyping. In: Rolland, C., Cauvet, C., Bodart, F. (eds.) CAiSE 1993. LNCS, vol. 685, pp. 165–183. Springer, Heidelberg (1993)
64. Loos, P., Allweyer, T.: Process orientation and object-orientation - An approach for integrating UML with event-driven process chains (EPC), Germany (1998)
65. Lu, R., Sadiq, S., Governatori, G.: On managing business processes variants. Data & Knowledge Engineering 68(7), 642–664 (2009)
66. Lu, R., Sadiq, W.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
67. Marsan, M.A., et al. (eds.): Proceeding of the International workshop on Timed Petri Nets, Torino, Italy. IEEE Computer Society Press (1985)
68. 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)
69. Medina-Mora, R., Winograd, T., Flores, R., Flores, F.: The Action Workflow approach to workflow management technology. In: ACM CSCW Conference (1992)
70. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., El Boussaidi, G.: Business process modeling languages: Sorting through the alphabet soup. ACM Comput. Surv. 43(1), Article 4 (December 2010)
71. Mühlen, M.Z., Becker, J.: Workflow management and object-orientation - A matter of perspectives or why perspectives matter. In: OOPSLA Workshop on Object-Oriented Workflow Management, Denver, USA (1999)
72. Nossum, A., Krogstie, J.: Integrated Quality of Models and Quality of Maps. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 264–276. Springer, Heidelberg (2009)
73. Nysetvold, A.G., Krogstie, J.: Assessing Business Process Modeling Languages Using a Generic Quality Framework. In: Siau, K. (ed.) Advanced Topics in Database Research, vol. 5, pp. 79–93. Idea Group, Hershey (2006)
74. Olle, T.W., Hagelstein, J., MacDonald, I.G., Rolland, C., Sol, H.G., van Assche, F.J.M., Verrijn-Stuart, A.A.: Information Systems Methodologies. Addison-Wesley (1988)
75. OMG, Semantics of Business Vocabulary and Rules Interim Specification (2006b), `http://www.omg.org/cgi-bin/doc?dtc/06/03/02` (retrieved January 1, 2006)
76. OMG. BPMN v2 Specification. Technical report, OMG (January 2011) (http://www.omg.org/), `http://www.omg.org/spec/BPMN/2.0/`
77. Ould, M.A.: Business Processes - Modeling and Analysis for Re-engineering and Improvement. Wiley, Beverly Hills (1995)

78. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
79. Petri, C.A.: Kommunikation mit automaten (in German). Schriften des Rheinisch-Westfalischen Institut fur Instrumentelle Mathematik an der Universität Bonn (2) (1962)
80. Recker, J., Rosemann, M., Krogstie, J.: Ontology- versus pattern-based evaluation of process modeling language: A comparison. Communications of the Association for Information Systems 20, 774–799 (2007)
81. Scheer, A.-W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management, pp. 301–304 (2000)
82. Searle, J.R.: Speech Acts. Cambridge University Press (1969)
83. Searle, J.R.: Expression and Meaning. Cambridge University Press (1979)
84. Searle, J.R., Vanderveken, D.: Foundations of Illocutionary Logic. Cambridge University Press (1985)
85. Senge, P.: The Fifth Discipline: The Art and Practice of the Learning Organization. Century Business Publishers, London (1990)
86. Singh, B., Rein, G.L.: Role Interaction Nets (RINs); A Process Description Formalism, Technical Report CT-083-92, MCC, Austin, Texas (1992)
87. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Wien (1973)
88. Suchman, L.: Do categories have politics? CSCW 2(3) (1994)
89. Swenson, K.D., et al.: A business process environment supporting collaborative planning. Journal of Collaborative Computing 1(1) (1994)
90. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N. (eds.): Modern Business Process Automation: YAWL and its Support Environment. Springer (2010)
91. UMM - UN/CEFACT Modeling Methodology User Guide (2007)
92. Ward, P.T.: The transformation schema: An extension of the dataflow diagram to represent control and timing. IEEE Transactions on SE 12(2), 198–210 (1986)
93. Wegner, P., Goldin, D.: Interaction as a Framework for Modeling. In: Chen, P.P., Akoka, J., Kangassalu, H., Thalheim, B. (eds.) Conceptual Modeling. LNCS, vol. 1565, pp. 243–257. Springer, Heidelberg (1999)
94. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York Inc. (2007)
95. WfMC Workflow Handbook 2001. Workflow Management Coalition, Future Strategies Inc., Lighthouse Point, Florida, USA (2000)
96. White. S.A. Introduction to BPMN. IBM Cooperation (2004)
97. Winograd, T., Flores, F.: Understanding Computers and Cognition. A-W (1986)

On the Dynamic Configuration of Business Process Models

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

[1] Universidade Federal de Pernambuco - UFPE, Centro de Informática, Recife, Brazil
{ebs,jhcp,jbc}@cin.ufpe.br

[2] University College London - UCL, Department of Computer Science, United Kingdom
a.finkelstein@ucl.ac.uk

Abstract. Business Process Models are a relevant input for the development of information systems. Since processes are performed in increasingly dynamic business environments, the processes are required to be flexible and dynamic as well, adapting to environmental changes. Thus, it is essential to properly represent variability in Business Process Models. Moreover, in order to allow for adaptive and autonomic systems, it is of paramount importance to reason on the variability of a process, being able to select a process configuration for a given context. In this paper, we present an approach for such context-aware reasoning, on which the business process configuration is driven by Non-Functional Requirements. Using independent models for expressing variability representation, configuration knowledge, contextual information, and the process itself, we present algorithms and mechanisms to perform business process configuration at runtime. Furthermore, we describe experiments we conducted in order to assess the suitability of our approach.

Keywords: Business Process Configuration, Non-Functional Requirements, Adaptive and Flexible Information Systems, Context-Aware Information Systems.

1 Introduction

Business Process Management (BPM) is a systematic and structured approach to analyze, improve, control and manage processes with the aim of improving the quality of products and services [1]. In organizations that adopt BPM, the business process models play a central role by capturing the way activities are performed. The processes are becoming increasingly complex and heterogeneous, as they often include activities of different nature involving people, software, and hardware placed in diverse physical surroundings. Moreover, some application domains that are influenced by environmental, geographical and human factors, such as logistics and transportation (e.g., airline companies), have to maintain their processes updated and valid in order to keep running their business properly. More than just changing them, the companies need to be aware of quality constraints that affect their business processes (e.g., security, reliability, performance, and so on).

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 331–346, 2012.

By involving elements of different nature, the processes are increasingly dynamic and therefore more prone to changes [2]. Hence, the flexibility in business process is vital in order to support this heterogeneity. The business process models adapt to changes by providing a description of (i) the parts that can be modified in the process, (ii) the criteria that drive the modifications, and (iii) the mechanisms used to perform adaptation. Business process flexibility can be achieved by several methods [3–6], usually providing ways to represent the variability of business processes and means to perform the configuration of processes to obtain new instances. However, the configuration of business process models currently relies on human experts, such as business analysts, which are often expensive and not always available. In highly dynamic and complex environments that require immediate adaption, this is no longer acceptable. For example, an emergency, such as the volcanic ash cloud crises which massively disrupted air traffic in Europe in 2010 and Latin America in 2011, calls for immediate intervention.

Several works represent variability in business processes [3, 4, 7] and propose mechanisms to modify the process models according to the situation. However, they often lack the necessary guidance to become adaptable to a given context, for example the closure of the air space above a certain height or due to bad weather. In industrial settings the configuration is usually performed on an ad hoc basis, guided solely by the analyst's experience. However, in more dynamic environments, the changes have to be performed more frequently and systematically. Moreover, current approaches [3] that guide the configuration of process models usually just consider high level quality constrains such as cost and performance. Other important quality attributes that could affect the business process [11], such as security and availability, are seldom taken in account.

In previous works we introduced our approach to deal with business process variability and its configuration using NFRs [5] and contextual information [6]. We investigated how to obtain configurations for business process models that are aware of contextual changes and that meet stakeholders' preferences over non-functional requirements. We have proposed a configuration process that relies on contextual information to identify change opportunities. We also claim that Non-Functional Requirements (NFR) [8] can define important constraints that the business process must comply to. Hence, we advocate the use of NFR as qualitative criteria to drive the configuration of business process models and the application context-awareness in order to deal with changes in the environment.

In this paper we break new grounds and present novel contributions. First, we define the metamodel of the proposal based in the conceptual model presented in [6]. The metamodel incorporates a detailed description of the modeling elements including new connections and the linking with another metamodel. It is very important to describe a modeling language and the correspondent tool support. Moreover, it also allows the definition of constraints using a specific language (i.e., OCL). We also improved some steps of our configuration process to include the algorithms used to perform the configuration/generation of process models. More precisely, the last step of our process is detailed to explain the computation necessary to select a configuration and generate a new process model. Last but not least, we present an assessment of our approach using a simulation of business process models execution.

The remainder of this paper is structured as follows. Section 2 introduces the background information and some basic concepts. Section 3 presents our approach. An assessment of our approach is presented in Section 4. We compare our proposal to some related works in Section 5. Finally, in Section 6 we conclude the paper.

2 Conceptual Framework and Background

Our proposal concerns business process configuration, using the notions of NFR and contextual information. We rely on independent models for expressing variability representation, NFRs, contextual information, and the process itself. In the following subsections we present the background on these topics. In Figure 1 we depict the metamodel of our approach. It includes the main concepts used in our approach such as Variants, Variation Point, NFR and Context. Moreover, it also describes the relationships among them and the way how our model is linked to the business process model. Hence, it consists of classes that have attributes and can be linked through association, aggregation and inheritance relationships. The BPMN model is represented using the Eclipse BPMN 2.0 meta-model (package in dark grey). The metamodel was described using the Ecore Language modeling facility of Eclipse modeling framework. The *WorkflowPattern*, *ContributionType*, and *VariationPointOperator* elements are of enumeration type, i.e. they describe a set of values that the attribute of these types can assume. Some relationships are too complex to be expressed by the modeling language. In these cases we have used the OCL (Object Constraint Language) to express constraints and derivations. For example, the constraint involving *VariantsRelationship* and *Variant* classes, which states that a variant cannot requires or excludes itself was expressed in OCL. Due to space limitation we omit the OCL constraints in this paper.

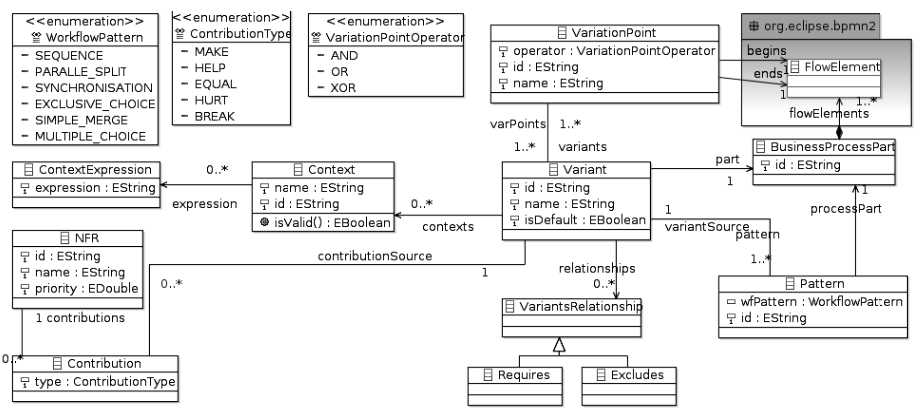


Fig. 1. The Metamodel for our Approach

As a motivating example, let us consider check-in (see Fig. 2) related processes in the airport domain. Check-in is usually the first procedure for a passenger when arriving at an airport, as airline regulations require passengers check-in by certain times prior to the departure of a flight. This duration usually spans from 30 min to 4 h depending on the destination and airline. During this process, the passenger has the ability to ask for special accommodations such as seating preferences, inquire about flight or destination information, make changes to reservations, accumulate frequent flyer program miles, or pay for upgrades. The airline check-in's main function, however, is to accept luggage that is to go in the aircraft's cargo hold.

Several activities related to airport check-in process are relevant and useful for the proper definition of appropriate business process models (such as passenger check-in policies and procedures regarding security, luggage handling, passenger handling and document validation). Check-in options and procedures vary according to the airline as some airlines allow certain restrictions that other carriers have in place, and occasionally the same airline at two separate airports may have different check-in procedures.

These types of process runs in dynamic environment, since the processes may be affected by several factors such as load of passengers, security policies, weather and so on. In the above scenario, it could be helpful to configure the process according the context changes but also considering the quality preferences associated with the check-in process.

Business Process Modeling. A Business Process Model consists of a set of logically ordered activities that are performed to produce goods or services [3]. BPMN is a workflow based language that models business process based on flows of task and data. In Figure 2 we have a sequence of activities that are performed during the check-in and boarding process. The process starts with request of the flight ticket in order to verify the flight information, and then the check-in is performed. After that other airport control checks are executed. The last step is to board the airplane.

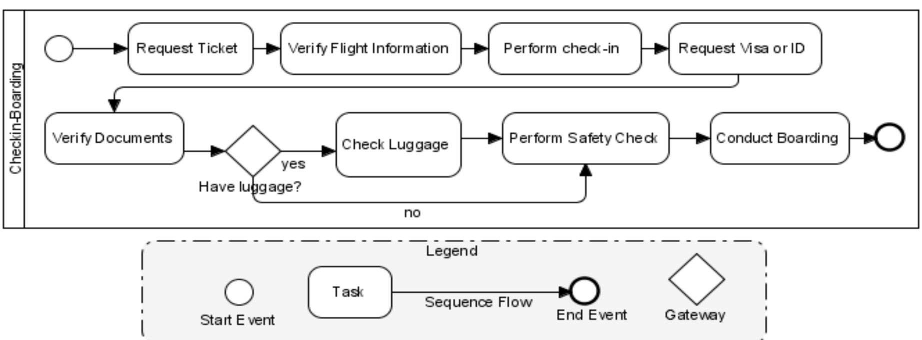


Fig. 2. BPMN model of an Airplane Check-in and Boarding process

Variability modeling in business processes models represents alternative ways of how activities are performed including the actor responsible for performing them, the resources required, and so on. We rely on Variants and Variation Points to describe the desired variability. Variation Points are the subjects of change, while variants are the objects of change [9]. In our case, both Variants and Variation Points are represented by business process model fragments. Observe that Variants can be included or removed from Variation Points. It is important highlight that the variability information is stored in a specific model without extension of the initial business process model. See in Figure 1 that the BPMN is part of another metamodel.

The description of a *Variation Point* includes an identifier (name), an operator (AND, OR, XOR), a point of reference (begin and end) and a list of the *Variants* that can be placed in it. *Variants* can be associated to one or more *Variation Points*. The *Variation Point* in its turn begins and ends in points of the process that can be of any type. Moreover, the *Variants* can interact with each other – e.g., they can require or exclude the presence of other *Variants* on another *Variation Points*.

In our approach, the *Variants* will be related to a *Variation Point* through a pattern. In order to describe the variants we use an identifier, the point where it should be inserted, the dependencies that may be present and a pattern. *Patterns* are used to indicate how process elements will be placed in the resulting business process model. Note that we refer to workflow patterns described in literature [10]. It is important remember that these patterns are specific for workflow languages and differ from the design patterns used in software development. Several types of workflow patterns are available such as sequence, parallel split, exclusive choice, multiple choices and so on.

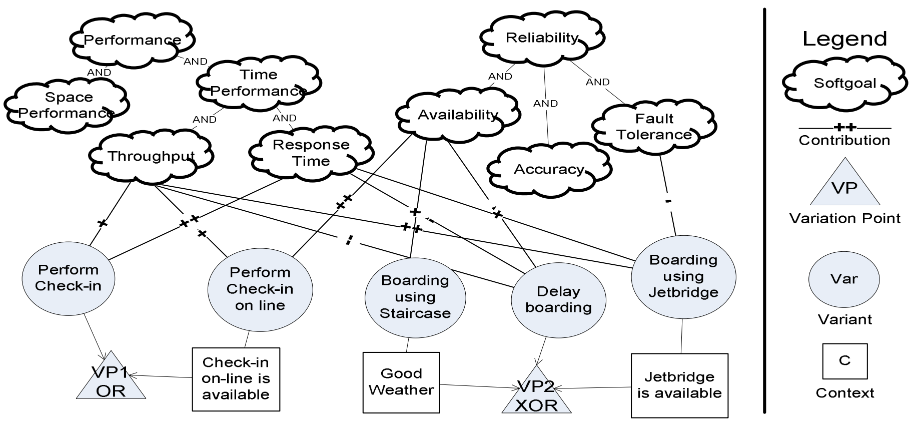


Fig. 3. Relating Variability, context and non-functional requirements

Non-Functional Requirements (NFRs). NFRs are requirements that describe qualities and constraints. Requirements Engineers have long relied on the concept of NFR to describe and analyze the requirements of systems and their relationship with the functional ones. The NFR Framework [8] introduces the concept of Softgoal to represent the NFR as well as means to assess their satisfaction. NFRs can be reused through catalogues that describe how to decompose and operationalize the NFR.

In our work we rely on the NFR Framework [8] to represent the quality attributes required by our approach. Since we are using *NFR* to configure the business process model, it also needs to be linked to the *Variants*. The relationship between the *NFR* and *Variants* is expressed by contributions, which indicate the positive and negative interaction among them. Figure 3 presents a simplified example of catalogues for *Performance* and *Reliability* linked to elements of our approach. A *Variant* can contribute to several *NFR (see Perform Check-in On Line)*, whilst a *NFR* can be contributed by several *Variants* (check *Availability*). However, a *Variant* has just one contribution value to a *NFR* at a time. In our case we adopt a numerical scale, from positive, with maximum value of 1, to negative with minimum value of -1. For example, *Delay Boarding* variant has a very negative (-1) impact on the *Availability* softgoal (constraint).

NFRs are important for business processes modeling. However, they are seldom considered during modeling. Some few approaches apply NFRs during the design by means of extensions of business process modeling languages. For instance, [11] and Pavlovski et al. [12] take the Non-functional requirements into consideration during the software design process. The former by using NFR catalogs during the design and the latter by extending the BPM to incorporate NFR. However, neither considers the variability in their solutions.

Contextualization. A Context is a partial state of world that is relevant to achieve goals [13]. In our case it is relevant information that could affect the business process execution. Contextual analysis is based on context annotations. Annotations are attached to elements of a model in order to indicate the relevant context information that affect that part of the model. Hence, contextual parts of that model can be enabled or disabled. During the analysis the *contexts* can be associated to *facts* and *statements*. Facts can be directly assessed, while *statements* must be decomposed as facts. *Contexts* are linked to sets of *facts* that can be assessed to identify the validity of the *context*. In this paper, we adopted a simplified version of the proposal of Ali et al [13]. A *Context* is described in natural language, and it is composed by *Context Expressions* that allow the computation of validity of a context in a given moment. A *Context Expression* associates the values of the monitorable variables to logical expressions to assess if the context is valid or not.

3 NFR-Driven Configuration of Business Process

Our approach consists of five activities: *Elicit Variability*, *Describe Variability*, *Analyze Context*, *Link NFRs & Variants* and *Perform Configuration*. The first four steps are performed at design time (see Fig 4). While the last step, *Perform Configuration is executed* at runtime (see Fig. 6). Note that the configuration is driven by the stakeholders preferences express in NFR models. Hence the rationale for the selected configuration becomes explicit. Moreover, it can be performed while the processes are running, i.e. it becomes run time adaptable.

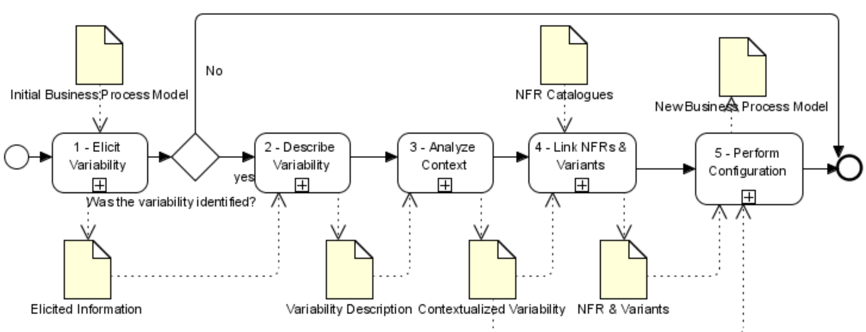


Fig. 4. The process of our approach

Variability Elicitation. The process starts by identifying and discovering possible variations in a model. The objective is to uncover different ways to carry out a process, the effects of the inclusion, change or exclusion of elements of the model. A questionnaire can be used to help to identify different perspectives in the business process models that were not clear in the initial model. In order to perform this elicitation we use an information analysis framework [15] that explores different facets of the information and obtain new data about it. The elicitation produces a list of variations that needs to be represented in order to reflect the structure of a workflow business process model.

Variability Description. Once the possible variants and variation points of the process are identified, they need to be described. Hence, in this step we specify the variations in terms of the standard BPMN notation. As previously explained, we represent the variations using the concepts of Variation Points and Variants. Variation Point is the place where the variation occurs. The variants are described as parts of BPMN and associated with a pattern that indicates how it will be placed in the model.

Figure 5 presents some variants for the check-in. The Check-in can be performed in several ways, it can be performed manually (variant A) in the case where the check-in system is down. It also can be executed at the airport by an operator using a support system (variant B), or by the passenger using an online check-in (variant C). There is also the possibility in combine two ways (variant D), which increases the reliability of the process.

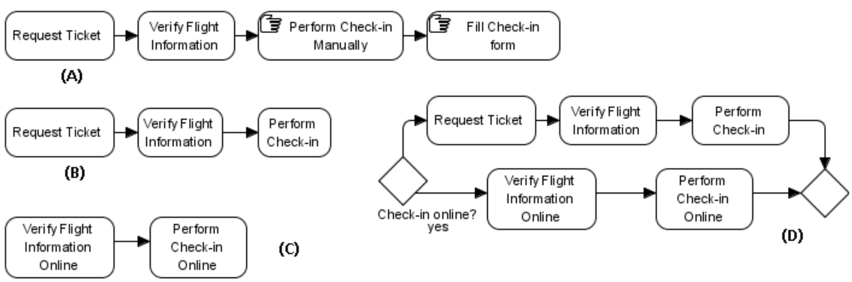


Fig. 5. The Variants for check-in

Context Analysis. The third activity in our approach is to identify the Contexts that could affect the model. It is performed by studying the domain and the relationship of the actors and domain concepts [13]. The user or system states in the environment are described as Contexts. The Context can capture what is going on as well as the location related information (where). It can also include the available resources and other relevant information. A Context consists of expressions and variables that need to be evaluated to check if the context holds.

We need to identify the relationship between the Context and its Variants and Variation Points. Note that the task can only be performed if the Context is valid. Hence, the Contexts represent the monitorable runtime information that will enable or disable possible alternative ways to deal with the process. The evaluation of NFR is performed at design time and the result is used at the runtime. Hence, if the best solution for a NFR is disabled due to the context then another one can be considered.

Linking NFRs and Variants. In this fourth step we identify the NFRs that are critical for the process. Moreover, we define the impact of each Variant to the NFR by means of contribution links. This information can be gathered interviewing experts involved in the business process, using requirements catalogues or any mix of elicitation techniques. Note that the NFR analysis can indicate that several (possibly conflicting) non-functional requirements are to be met.

Once the NFR are identified, we perform the linkage between the process variants and the requirements. These links will be represented using matrices (not shown due to space limitation), which is a usual and scalable solution for representing this kind of information. Moreover, matrices allow the construction of views containing only a partial representation of the variants and the requirements, simplifying its analysis.

NFRs can be used to prioritize the Variants, which lead to the selection of the configuration. Since many alternatives can emerge during the elicitation process, the contribution analysis can be time consuming. However, we claim that the use of NFRs as selection criteria can help to reduce the variability space and thus drive the modification process.

Configuration of Business Process. The configuration of process is a critical step in our approach. All the collected and modeled information is used to obtain new process models. In this last activity we consider the Variation Points and the Variants of the business process, and assess how they impact the non-functional requirements. This information can be used to support the configuration itself (see Fig. 6). It can be performed based on Variants selection or the most critical NFRs.

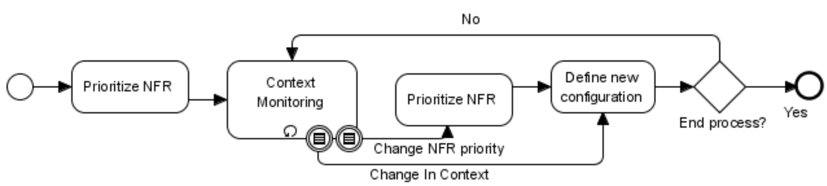


Fig. 6. The Perform Configuration sub-process

Some solutions only rely on expert judgment and NFRs to resolve conflict at design time [8]. Since we are dealing with runtime adaptability, it may not be possible to rely on experts (e.g., they could not be available anymore). In our approach we require the *NFR* prioritization to be conducted before entering in the monitoring loop (see Fig. 6). It is performed by an analyst, who assigns weights to each *NFR* according to their priority weights. Moreover, the *Variation Points* must be associated to a *Context* independent *Variant.*

There are several ways to sort out priorities variants using *NFRs.* A common solution is to rely on weighted averages, where contributions can be counted and weighed according the *NFR*. Although, this method is intuitive it could hide the interaction between the *NFRs*. In order to obtain a global ranking that takes into account the local interactions we adopted a multi-criteria decision making method.

We chose the Analytic Hierarchical Process (AHP) [16] method which generates a global preference measure based on the choice among alternatives. The AHP was selected because it fits well with the structures used in our approach. For example, the hierarchical criteria are represented by the NFR decomposition, while the preferences among alternatives represented by the contributions of Variants to NFR. Moreover, the use of priorities over NFR is also taken in account by the AHP.

According to process described in Figure 6, the next step is to start the Context Monitoring loop that will detect changes in the Context and NFR priorities. Note that if changes in the contexts are detected, a selection of a new configuration is required. Each Variation Point is evaluated to identify the Variant that better fits the non-functional requirement, i.e., the Variants with the highest positive impact on that given NFR. This evaluation is automatically performed.

Let us consider N as the set of *NFRs*, Var as the set of *Variants*, VP the set of *Variation Points*, C as the set of *Contexts*, and $\mathrm{contrib}(\mathrm{v}, \mathrm{n})$: the value of the contribution link from the variant v to the *NFR* n. The contribution function varies in the following range: $1 \geq \mathrm{contrib} \geq -1$. The NFR have weights (w) associated to them to express their different priorities.

```
1 for all c ∈ C do
2    validate(c)
3 end for
4 for all vp ∈ VP do
5   for all v_i ∈ Var, where v is part of vp do
6     if isValid(v_i) then
7       validVar[i] = v_i
8       for all n_j ∈ N do
9           contribMatrix[i][j] = contrib (v_i, n_j)
10      end for
11    end if
12    end for
13  end for
14 ranking[] = AHP(contribMatrix[][], validVar[], w)
15  solve(VP, ranking[ ])
```

Fig. 7. Configuration algorithm

First, the algorithm (see Fig. 7) computes the valid contexts (lines 1-3). After that, the Variation Points are evaluated to identify the valid Variants (lines 4-6). A Variant can be valid for one VP and invalid for another. There are two ways to be valid: being associated to a valid context, or having no context associated to it (default situation). The valid *Variants* are included in a specific set (line 7), and their contributions to the *NFR* are computed (lines 8-10). The set of valid *Variants* and their contributions to *NFR* are the input for the AHP method that will compute a global ranking of preference among *Variants*. In its turn, this ranking is the input to a solver that will derive a valid solution considering the relationships (i.e., exclude and include) among the *Variants*. Another trigger for the changes is priority modification. Note that if the priority of a *NFR* changes it is dealt with similarly to the contextual change.

Once the set of *Variants* is selected they are grouped in a new instance of the business process model. Each *Variant* has its pattern evaluated and the appropriate action is selected. The set of flow elements that is composed in the variants can be placed in parallel to any other variant in the same variation point. The action substitution means that the original task will be replaced by this one. We have implemented these changes in the model using the Query/Validate/Transform-Operational (QVTO) model transformation language. Due space limitation we do not present the model transformations but it and other support material are available at [19]. We have also used the Eclipse platform modeling tools to develop an editor that creates configuration models based on the meta-model of Figure 1. The initial BPMN is inputted and transformations associated with the patterns are applied.

4 Approach Evaluation

In this section we present an experiment that was performed in order to evaluate some characteristics of our approach. An important question to answer is, do the models produced by our approach are more adapt to the environment than a standard process?

The experiment was executed using the simulation feature of the Bonita Open Solution 5.6 business process platform. This environment allows the representation of the process data, simulating the environment resources available to execute the activities, and the configuration of time. Hence, we were able to control the variables of the experiment, allowing for a reproducible study.

The objects of study are *business process configurations* generated using our automatic configuration approach. These models were compared with respect to a *basic* process, using the *same* scenarios. The purpose is to evaluate the business process configurations generated with our approach, verifying if they actually *improve* the process in different context settings. The quality focus is the *performance* of a given process configuration, running in a simulated environment. Two dimensions of performance were considered: the *time* required to execute a process; and the *resources* required to execute a process. The *perspective* is the researcher's point of view. The models used in this experiment were produced by one of the *authors* of this paper, and are an extended version of the process depicted in Fig. 2.

Hypotheses, Variables and Measures. In this experimental study we focused on the following research questions, defining the respective sets of null and alternative hypotheses:

- RQ1: Considering the use of NFRs, do the models produced by our approach consumes less *resources* and require less *time* to execute in comparison with a standard business process configuration?
- RQ2: Considering the adaptation to a context change, do the models produced by our approach consumes less *resources* and require less *time* to execute in comparison with a standard business process configuration?

From these research questions, we generated the following hypothesis:

- $H_0 1$: The business process models produced by our approach, when considering the use of NFRs to configure the models, do not consume less resources or less time when compared the standard process;
- $H_0 2$: The business process models produced by our approach, when considering the adaptation to a context change, do not consume less resources or less time when compared the standard process.

If the null hypothesis can be rejected with relatively high confidence, it is then possible to formulate alternative hypotheses:

- $H_a 1$: The business process models produced by our approach, when considering the use of NFRs to configure the models, consume less resources and less time when compared the standard process
- $H_a 2$: The business process models produced by our approach, when considering the adaptation to a context change, consume less resources and less time when compared the standard process.

The independent variable of this study is the modeling method which can assume one of the values in {*Standard*, *Adaptable*} – standard is a traditional BPMN model, without variability; adaptable is a process model generated with our approach, including variability, contextual and non-functional information. The dependent variable is the performance measured using the execution *time* (in hours) and *resources* consumption (in cost). Resources consumption includes personnel costs, equipment costs, fares, and so on. Both dependent variables were calculated by the simulation engine, considering estimated time and resources required to perform each activity of the process. Also, both dependent variables are inversely proportional to the *performance*, i.e. the lowest the value of execution *time* and of *resources* consumption, the better the *performance*. The measure variables were the average *execution time* by instances in hours and the average *resources cost* by instance in dollars.

Design and Execution. A basic requirement of an experiment is the ability to control the object of study and its parameters. Since our approach allows modifying several parameters first we need to control what parameter will be modified. We designed the experiment to assess the impact of contextual changes and the impact of NFRs priority change. Two scenarios were designed: the case where there are no changes in context (S0) in the environment and the situation where there are changes (S1). In the S0 scenario we block the value of contextual information and change the value of NFRs' priorities. This way we can see the impact of changing the NFRs' priorities over the process. In the S1 scenario, we change the value of a context variable and repeat the simulations to verify the impact of contextual change over the resulting process models.

A standard check-in and boarding process described in the literature [2] was considered the standard business process model (see Fig. 1). Afterwards, we used our approach to design a new business process models based on the change in the S1 scenario. For each, scenario we simulated the case with and without *contextual* change. For instance, the variants *perform on-line check-in* and *use jetbridges to board* can improve the *performance* of the process by reducing the waiting *time*.

However, this type of alternative can increase the *costs* or may be unavailable at the moment when the process needs to be executed. In the scenario with contextual change we consider the *on-line check-in* is unavailable; this change will force the selection of the variant *Perform Check-in* task at the airline counter. We decided to use only a single contextual change in order to isolate the change effect, and avoid uncontrollable results. However, if the contexts do not have cross interaction other context changes could be defined as well.

In order to describe the impact of the context in the process we based our estimate in the data related to the air traffic delay recently experienced in European airports [17]. The simulation data requires the estimative of the tasks duration, the resources costs, and the number of involved personal. For example, the collected information could help to estimate how much time is necessary to perform the task *Conduct Boarding*. As consequence of being based in real information the simulation scenarios are more close to the real situation.

Results and Analysis. The simulation was performed for the 6 processes divided in two groups: 3 processes for scenario S0 and 3 processes for scenario S1. Each process was instantiated 100 times to obtain the average values presented in Table 1 and 2. Table 1 shows the results of the simulation execution for the S0 scenario and the Table 2 the result of S1 scenario. The simulated interval was one month, with working time of 24 hours per day. We consider that this process could be performed at any time of the day since many airports operate 24 hours a day.

Table 1. Simulation results for scenario S0

Modeling method NFR considered	Standard None	Adaptable Performance	Adaptable Reliability
Execution Time by instance	0.916	0.616	1.049
Resources Cost by instance	9.27	7.425	7.175

Table 2. Simulation results for scenario S1

Modeling method NFR considered	Standard None	Adaptable Performance	Adaptable Reliability
Execution Time by instance	0.916	0.666	0.916
Resources Cost by instance	9.321	9.642	9.212

Considering the *Execution time by instance* variable we can identify a very small difference between the average values for the scenario with and without contextual change. This is expected, since in the scenario without contextual change all process variants are available.

Comparing the configurations generated using our (adaptable) approach with the standard configuration in Table 1, we can see that the configuration driven by NFR *reduces* the *execution time*. In the case on which *Performance* is the NFR with higher priority, several variants that contributes positively to improve the process performance reduces the overall execution time – for instance, it is faster to *board with a jetbridge* than using *staircase*. The same goes for the configuration generated prioritizing *reliability*, since in this process configuration there are activities that contribute to reduce interruptions and to improve availability in the process.

If we consider the *resources cost* results, we note that the configurations without contextual change have a smaller cost. This happens because the contextual change suppresses the variant that points to the *Perform On-line Check-in* activity. Moreover, the *on-line check-in* has a smaller cost than a *check-in performed* at the airline counter. In general, the configured processes have different behaviors for the *cost*, for instance, the *performance* prioritized process uses variant that prioritize the *performance* without regarding the *cost* of the alternatives.

Discussions. In this section we relied on simulation as an empirical tool to present a preliminary assessment of our approach. We detected there is a relationship between how the process model behaves according to the changes (i.e., NFRs and Contextual information), and its performance. Considering the resource consumption and execution time we can reject the null hypotheses $H_0 1$ and $H_0 2$, and accept the alternative hypotheses $H_a 1$ and $H_a 2$.

Moreover, we conclude that our adaptable approach has slightly improved the execution time for the check in procedure. Since, this is a process to be executed by all passengers to be flown, the small gains can add up to enormous benefits.

There are threats to the validity of our study. For example, the *reliability* of measures – our measures were compiled from the simulation reports generated by the simulator. In order to deal with this threat we selected just average measures and discarded outliers. There are also *design* threats, such as the interaction among treatments and the mono-method bias. In order to reduce these threats we combine the treatment with different settings including the use of different NFRs such as Performance, Reliability and as parameters for the process configuration, and repeating the simulation with and without contextual changes.

5 Related Work

It is well known that some approaches also rely on Software Product Line principles to deal with variability in BPM. For instance, Schnieders and Puhlmann [7] propose the extension of business process modeling languages to describe variability. However, they do not provide mechanisms to drive the configuration of process models. Neither consider NFRs and contextual information as we do.

There are also some works that provide mechanisms to help to drive the configuration of Business Processes. For example, La Rosa et al. [3] propose a questionnaire based approach that relies on configurable process models to obtain new instances of BPM. The users receive guidance during the configuration by answering questionnaires. Their approach is intuitive but it has a limited application for run-time self-configuration since it requires the user intervention to produce new versions of the model. Note that our proposal also supports user interaction to guide the configuration process, e.g. the change in NFR prioritization. However, human interaction is not required at run-time stage to generate new process models.

Lapouchnian et al. [18] offers a goal oriented approach to configure BPM. They obtain business process models from annotated goal models. Moreover, they configure the process model using NFRs represented by softgoals. Our approach also relies on NFRs. However, we adopt a completely different structure to represent the BPM and variability. We start from a reference process model and represent the variants as process chunks instead of using a goal model. In doing so we maintain the representation in the same abstraction level without the need to annotate or convert the models. Besides, we also use contextual information to support dynamic configurations.

De La Vara et al. [14] proposes an approach for contextualization of BPM. They rely on the context analysis of Ali et al. [13] to represent the contextual information. Additionally, they guide the definition of contexts and inclusion of contextual information in the BPM and allow characterization of variants based on the context information. Our approach also relies on contexts. However we deployed a different strategy. We defined the variants and variation point before the inclusion of context information. Moreover, we used NFRs to guide the selection of a configuration and we did not extend the business process modeling language to include the contextual information. In doing so our approach can import BPMN models designed by any tool based in the Eclipse framework BPMN 2.0 metamodel.

6 Conclusions and Future Work

In this paper we proposed a novel and flexible approach for the configuration of business process models. It relies on contextual information and NFRs. A process was outlined. It includes the elicitation of variability information, which is central to the configuration process itself. Besides guiding the configuration with clear criteria, this approach also provides the rationale for the selected configuration.

We have proposed an approach that keeps the variability representation and context-information separate from the business process models. In doing so, we traded off intuitiveness for the sake of flexibility. For instance, we did not need to extend the BPNM to deal with variability. Moreover, we relied on patterns (i.e., workflow patterns) and analysis algorithms (i.e., SAT solvers). Hence, we envisage that our approach could be used with different business process modeling languages, i.e. few modifications are expected.

Different from some other approaches that support the NFR evaluation (e.g., [8]), we do not require the user intervention to solve conflicts during the configuration phase. If necessary our approach could also support the interaction with user to update their preferences, i.e., to change priorities of NFR. However, it is not mandatory to produce a new version of the model.

We consider that the most critical part of our approach is to relate the degree of impact of each variant to the NFRs. This could be eased through the creation of catalogs which could help to define, for each kind of activity in a business process, the impact of that activity on specific NFRs. Our approach requires a business analyst during the early stages of the configuration process. On one hand, it may be a limitation since an expert knowledge could be expensive and time-consuming. On the other hand, requiring a business analyst reduces the chance of producing incoherent process models.

Some may claim that the approach might be time consuming, as each element in the business process may experience several variations. Certainly, the elicitation effort is also related to the number of non-functional requirements under consideration. However, this seems to be an inherent problem of any approach that deals with variability, since the amount of variations that may arise in real situations is potentially large. We believe that further improvements, currently under way, such as the automation of some of its steps and the adoption of mechanisms to handle complex models, could minimize these shortcomings.

As future work we intent to improve the tool support and integrate our solution in a BPM execution engine. Moreover, several assessments can be performed to validate other aspects of our approach such as test the usability and acceptance in real situations.

Acknowledgments. We would like to thank funding agencies that supported this research: CNPq and CAPES.

References

1. Chang, J.F.: Business process management systems: strategy and implementation. Auerbach Publications, Baton Rouge (2006)
2. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. International Journal of Business Process Integration and Management 3, 47 (2008)
3. La Rosa, M., van Der Aalst, W., Dumas, M., ter Hofstede, A.: Questionnaire-based variability modeling for system configuration. Softw. and Sys. Modeling. 8, 251–274 (2009)
4. Montero, I., Peña, J., Ruiz-Cortés, A.: Business Family Engineering: Does it make sense? In: Proceedings of the 1st Taller sobre Procesos de Negocio e Ingeniería del Software (2007)
5. Santos, E., Pimentel, J., Castro, J., Sánchez, J., Pastor, O.: Configuring the Variability of Business Process Models Using Non-Functional Requirements. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 274–286. Springer, Heidelberg (2010)
6. Santos, E., Pimentel, J., Dermeval, D., Castro, J., Pastor, O.: Using NFR and Context to Deal with Adaptability in Business Process Models. In: Proceedings of the 2nd International Workshop on Requirements@Run time 2011, Trento, Italy (2011)
7. Schnieders, A., Puhlmann, F.: Variability Mechanisms in E-Business Process Families. In: Proceedings of the 9th Int. Conference on Business Information Systems, BIS 2006 (2006)
8. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers (2000)
9. Pohl, K., Bockle, G., Linden, F.: Software Product Line Engineering. Springer, Heidelberg (2005)
10. Wohed, P., van Der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: Pattern-based Analysis of BPMN, Brisbane, Australia (2005)
11. Xavier, L., Alencar, F., Castro, J., Pimentel, J.: Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e NFR. In: Proceedings of the 13th Workshop on Requirement Engineering, WER 2010, Cuenca, Equador, pp. 29–40 (2010)

12. Pavlovski, C.J., Zou, J.: Non-functional requirements in business process modeling. In: Proceedings of the Fifth on Asia-Pacific Conference on Conceptual Modelling (APCCM 2008). pp. 103-112. Australian Computer Society, Inc. (2008)
13. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requirements Engineering 15, 439–458 (2010)
14. De La Vara, J.L., Ali, R., Dalpiaz, F., Sanchez, J., Giorgini, P.: COMPRO: A Methodological Approach for Business Process Contextualisation. In: Proceedings of the 18th International Conference on Cooperative Information Systems (2010)
15. Liaskos, S., Lapouchnian, A., Yu, Y., Yu, E., Mylopoulos, J.: On Goal-based Variability Acquisition and Analysis. In: Proceedings of the 14th IEEE International Requirements Engineering Conference, pp. 92–96. IEEE Press (2006)
16. Saaty, T.L.: Relative Measurement and Its Generalization in Decision Making Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors The Analytic Hierarchy/Network Process. Rev. R. Acad. Cien. 102, 251–318 (2008)
17. Leggat, T., Catherine Yven, Y.: A Matter of Time: Air Traffic Delay in Europe. In: EUROCONTROL (2009)
18. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 246–261. Springer, Heidelberg (2007)
19. Business Process Models Variability Configuration with Contexts and Non-Functional Requirements, `http://portal.cin.ufpe.br/ler/Projects/BVCCoN.aspx`

A Combined Process for Elicitation and Analysis of Safety and Security Requirements

Christian Raspotnig[1,3], Peter Karpati[2], and Vikash Katta[2,3]

[1] University of Bergen, NO-5020 Bergen, Norway
[2] Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
[3] Institute for Energy Technology, NO-1751 Halden, Norway
christian.raspotnig@hrp.no, kpeters@idi.ntnu.no,
vikash.katta@hrp.no

Abstract. The aim of safety and security assessments are very similar since they both consider harm during system development. However, they apply different means for it and are performed in separated processes. As security and safety areas are merging in new systems that are critical, and more openly interconnected, there is a need to relate the different processes during the development. A combined assessment process could save resources compared to separated safety and security assessments, as well as support the understanding of mutual constraints and the resolution of conflicts between the two areas. We suggest a combined method covering the harm identification and analysis part of the assessment process using UML-based models. The process is applied on a case from the Air Traffic Management domain. Experts' opinions about the results have also been collected for feedback.

Keywords: safety, security, combination of assessments, requirements engineering, UML.

1 Introduction

Modeling methods have become an indispensable part of information system development, and modeling languages such as Unified Modelling Language (UML) [1] are widely used in many domains today [2]. One of the reasons for UML's popularity is that its visualization capabilities, which ease communication between stakeholders with various backgrounds [2]. This advantage has also been recognized in the security field [3, 4, 5, 6], and new modeling techniques combining UML and security aspects were developed for software engineering. The safety field has well established traditional techniques, such as Hazard and Operability studies (HAZOP) and Failure Mode and Effect Analysis (FMEA) [7]. Even though these techniques and methods can be used along with system models, they do not offer a way to integrate safety aspects into the regular models to support visualization during the safety assessments that involve communication between different stakeholders. What they do offer during the safety assessments, on the other hand, is a systematic process for communicating and collecting information, which some security methods lack. Common to the security

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 347–361, 2012.

and safety fields is that stakeholders with different background have to build a mutual understanding of important security and safety aspects during the system development. If communication among stakeholders fails, it can lead to mistakes in the system under construction.

We therefore suggest the Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS) method, which defines a unified process for safety and security assessments. The process includes the use of techniques from both the safety and security fields, in order to allow visual modelling as well as a structured process for harm assessment. For security assessment we use the modelling techniques Misuse Case (MUC) [2] with textual templates and diagrams, and Misuse Sequence Diagram (MUSD) [5]. MUC is also used for safety assessment, but we use Failure Sequence Diagram (FSD) instead of MUSD for detailed failure analysis [6]. In both safety and security assessments the HAZOP is used for enhancing certain creativity and allow for a structured process, as well as for specifying the requirements as an outcome of CHASSIS.

To try out and evaluate CHASSIS we are taking part in an Air Traffic Management (ATM) company network. The network gathers more than ten companies for jointly improving their safety and security processes. We follow their safety and security assessments of a future remote tower and apply CHASSIS in a shadow case. Our primary aim is to investigate how CHASSIS can be used for a unified process of both safety and security assessments, and to compare it with the regular methods used in the ATM domain today.

The rest of the paper is structured as follows: Section 2 presents related work, while Section 3 introduces the CHASSIS method. Section 4 illustrates the method on a real-world example, whereupon section 5 discusses the main findings. Finally, section 6 concludes the paper and gives suggestions for further work.

2 Related Work

As a part of an ATM industrial network project, analysis of some of the widely used safety standards[1] in ATM industry, techniques were suggested that could be used to perform the members' safety assessments at system and system-software levels [8]. These techniques were either recommended or referred by some of the analysed standards. Some of the suggested techniques at the system level are Functional Hazard Analysis (FHA), Fault Tree Analysis (FTA), Dependence Diagram (DD), Markov Analysis (MA), and Common Cause Analysis (CCA). Some of the suggested techniques at the system-software level are Failure Modes and Effects Analysis (FMEA), Failure Modes and Effects Summary (FMES), FTA, DD, MA, and CCA.

The techniques in our method (described in section 3) cover the system level and to some extent the system-software level. Compared to the techniques mentioned above, our techniques used in CHASSIS build on extending the UML with notions from safety and security. This allows visualizing hazards and failures for safety as well as threats and vulnerabilities for security along with corresponding mitigations. While

[1] The term *standard* refers to regulation, standard, guideline.

most of the safety techniques mentioned above rely on usage of worksheets to structure and record the process of identifying hazards and eliciting safety requirements, we have developed a method of combining visualization and structure, by use of diagrams and templates respectively.

There are several security methods that resemble ours, with CORAS as the closest [9]. Earlier work in the CORAS project also combined UML with more traditional safety techniques, such as HAZOP and FMEA [29]. However, the method focuses on security aspects and does not aim at unifying safety and security assessments. There are also other methods that allow visualizing threats, such as Secure Tropos [10] and Abuse Frames [11]. Both of these methods extend methods from the software development process, but do not address safety aspects. The KAOS method does have similarities with CHASSIS, as it can address security and safety aspects with respect to obstacles in goal modelling [12]. It also comes with a security extension for analysing security aspects more into detail [13]. However, it does not address a unified method that includes the advantages of more traditional safety techniques, such as use of guidewords or worksheets. Firesmith proposed an information model for Defensibility Engineering in [14] by aligning the concepts of safety, security and survivability. The unified concepts were our starting point when considering the combination of safety and security assessment and we plan to use his other results as well (e.g. [15]).

HAZOP has also been investigated for security in the literature. [16] establishes HAZOP to be used for security issues in safety context. Its approach is to combine systematically the guidewords identified for programmable electronic systems [17] with the CIA (confidentiality, integrity, availability) attributes leading to an extended list of guidewords with security specific guidewords for identifying security threats. [18] investigates further the ideas in [16]. It finds that the derived guidewords in [16] are not enough flexible to bring out the analysts' creativity and therefore defines its own approach. They consider basic HAZOP guidewords in a more generic context than safety-critical context and combine them with different elements from the UML use case model. For example, different set of basic guidewords is defined for the actor's actions and for the actor's capabilities as well as for associations between actors and use cases or other use case description elements. We follow ideas in [18] so far that we do not define security-specific guidewords; however, we apply the basic HAZOP guidewords with CIA attributes similarly to [16] thus strengthening focus on security at the security part of our assessment. Currently, the basic guidewords are associated only with the use cases. Extending this scope of the guideword associations as suggested in [18] is under investigation by considering the trade-off between more details and overly complex model because of that.

Compared to ATM safety, ATM security is a relatively new topic. Some standards are emerging, especially from the SESAR project, such as the preliminary Security Risk Assessment Method (SecRAM) [19] and the Security Risk Management Toolkit (SRMT) [20]. While the former standard resembles ISO 27005 [21], the latter provide a comprehensive framework for assessing risk with respect to security aspects. SRMT outlines a comprehensive risk management toolkit for the European Air Navigation Service Providers (ANSP) and describes structured elicitation techniques for impact analysis and determining likelihood of attack, just to mention a few. However, compared to CHASSIS these techniques are applied at a higher organizational level.

Several methods mentioned above are close to ours and could be considered for inclusion in CHASSIS. One of the strengths of our method is that it relies on a combination of techniques that clearly visualize harm with a fitting notation, and that it builds on the popular UML modelling language.

3 Combined Harm Assessment of Safety and Security for Information Systems – The CHASSIS Method

3.1 Related Modelling Techniques

Misuse cases (MUC) [3] are an extension of use cases where it is possible to model not only the normal functionality wanted in the system, but also negative functionality that is not wanted, with the purpose of addressing harm-related concerns. In addition to regular use cases and regular actors, there are negative actors performing misuse cases that cause harm to the system. A misuse case threatens a use case when "the use case is exploited or hindered by a misuse case" [3]. A use case mitigates a misuse case when "the use case is a countermeasure against a misuse case, i.e., the use case reduces the misuse case's chance of succeeding" [3]. MUCs were defined for security purposes but can be applied also for safety assessments [22, 23]. MUCs allow early focus on security in the development process and facilitate discussion among stakeholders including regular developers with no special security training.

MUC diagrams (D-MUC) use inverted notation for misuse cases (oval with black background and white text) and misusers (head colored black). Thereby, they can be shown together with regular UCs in the same diagram without confusion. Two templates are suggested in [3] for the specification of textual MUCs (T-MUC): a lightweight and an extensive description. The lightweight description incorporates a field called threats in the regular UC description. The extensive description defines a separate template to support detailed definition and analysis of MUCs (see [3] for more details). To distinguish between safety hazards and security threats in the same MUC diagram, it was suggested to use gray nodes for the first notation and black nodes for the second notation in [22]. We follow this recommendation in our method. Alexander applied MUCs for trade-off analysis in [24] where he introduced a notation for showing conflicts between possible mitigations that we use as well. There were also other additional notations introduced for MUC, which we do not discuss here.

Sequence Diagram (SD) is one of the UML behaviour diagrams. While UC describes the system functions, SD describes the interaction of system components or objects represented by sequences of messages. Misuse Sequence Diagram (MUSD) extends the SD with an own notation for visualizing an attack, by visualizing how an attacker can exploit system vulnerabilities by a sequence of messages [5]. In section 4.3, Fig. 4 shows an example of a MUSD diagram. Red/dashed circles represent the vulnerabilities, while the attack messages are shown as red/dashed arrows. Furthermore, the green/dotted circle shows mitigation towards vulnerability. The technique has been used in a controlled experiment, where it showed the strength of visualizing temporal sequence of actions for security aspects [5]. Failure Sequence Diagram (FSD) adapts MUSD to failure analysis with respect to safety aspects. While the notation remains very much the same, the FSD shows sequences of failures and how they can propagate for a given system [6]. Fig. 3 (in section 4.3) shows a FSD, where a failure is represented by a red/dashed circle, propagation by a red/dashed arrow and mitigation by a green/dotted circle and green/dotted arrow. FSD was applied in an ATM case study, where it was used by to support FMEA in an industrial setting [25]. The technique proved well suited to both facilitate discussions, and explore and correlate the understanding of the system among the participants.

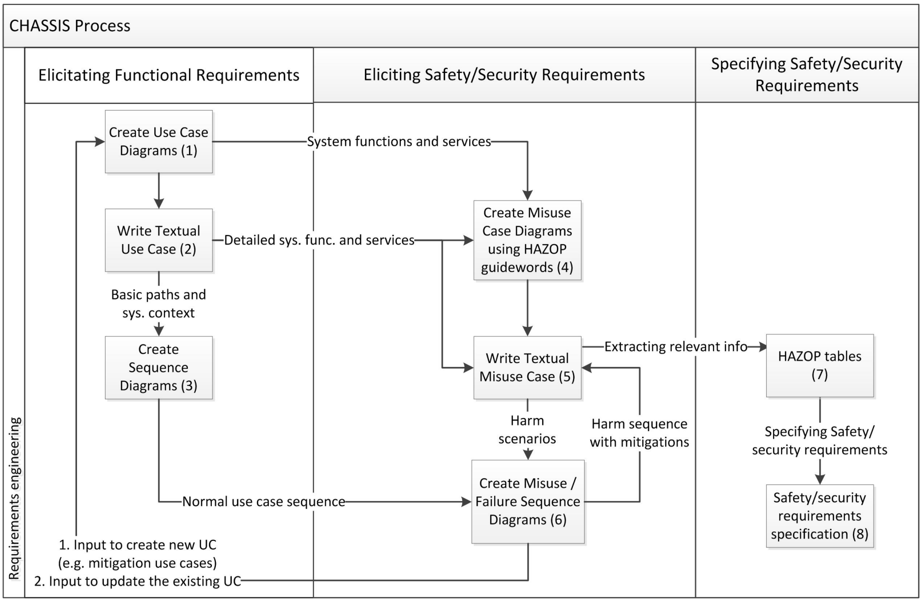


Fig. 1. CHASSIS' unified process with main activities and techniques

3.2 The CHASSIS Method

The CHASSIS method defines a unified process for both safety and security assessments. Common artefacts created through the assessments enable the overview of the relations between these two areas. The process is illustrated in Fig. 1 with numbered references for each step. We will use these reference numbers in the following description of the method to refer to the visualized steps.

The assessment process starts with the definition of system functions and services (Fig. 1 steps 1-3), which can be based on some operational and environmental descriptions of the system, and/or on discussions with stakeholders. UML use cases and sequence diagrams are used at this stage, which might have been already specified (e.g. in case of re-engineering a system).

The second stage focuses on the elicitation and analysis of safety or security requirements. Here, MUC diagrams are created based on UC diagrams and templates, extending them with possible misusers and harms towards the use cases (Fig. 1 step 4). The misuse cases are elicited in a systematic way by using guide phrases composed from a set of HAZOP guidewords applied for the use cases. In the case of security assessment, the possible meanings of the guide phrases are associated with CIA (confidentiality, integrity and availability) considerations. The MUC diagrams are detailed in textual MUC templates (Fig. 1 step 5). The stakeholders using FSDs or MUSDs refine the harm scenarios defined in the templates (Fig. 1 step 6). Those sequence diagrams are discussed further, vulnerable parts of the system and services are identified, as well as mitigation ideas considered and noted in the diagrams. The relevant new information from FSDs or MUSDs are fed back to the textual MUCs (Fig. 1 step 5), and the identified mitigations are defined as new functional use cases, leading to a new iteration starting with the first stage (Fig. 1 step 1). The new mitigations from security and safety areas can be in conflict with each other. MUC diagrams proved to be useful to support such trade-off discussions [24].

When a T-MUC is finished, HAZOP tables are prepared (Fig. 1 step 7) with its help in the third stage and the corresponding safety or security requirements defined (Fig. 1 step 8). The process ends when all the use cases and misuse cases were investigated concerning harms and mitigations respectively. Although Fig. 1 does not show, the most important source of information is the domain and expert knowledge of the stakeholders participating in the assessment (beside related documents).

CHASSIS focuses only on harm identification, analysis and mitigation, and does not consider risk management activities (e.g., compared to the process suggested in [21]). Our method provides input for assessing the risk of the elicited harms and the benefits of the collected countermeasures, which can be further analysed with quantitative techniques.

4 Desktop Example: ATM (Remote) Tower

4.1 Background

The Single European Sky ATM Research (SESAR) project [26] is about to impose great changes to European aviation. One such change is the development of ATM Remote Towers (RT) that aim at providing Air Traffic Services (ATS) from remote locations, as opposed to the traditional concept of providing the services from a physical tower at the airport. However, the remote concept has several safety and security issues to be addressed. The concept is of particular interest to the ATM company network, as it represents interesting challenges to technology development. The network consists of more than ten companies, ranging from SME's to a world leading supplier, as well as research institutes and a medium sized ANSP. The network represents stakeholders needed to develop and operate a RT solution. Furthermore, the network comprises safety and security competence for assessing such a solution.

There already exist ATM RT prototypes (e.g. [27]) on the market and further development of European solutions has been started by the SESAR project. Although the concept and equipment, e.g., camera surveillance of the manoeuvring area and the vicinity of the airport, have been developed for the prototypes, the thoroughness of safety and security assessments is not known. Although there exists many safety assessments of Air Traffic Control (ATC) tower systems, these assessments do not take into account that the ATS will be performed from a remote location. RT presents safety challenges both with respect to new operation and systems to be used, as the provision of ATS will rely on new technical solutions. Relying on new technological solutions can give benefits, but also make the system more vulnerable both with respect to security and safety. Thus, the RT case is an ideal subject for us to investigate CHASSIS in a relevant and practical setting.

The ATM company network has developed safety and security assessments of the RT concept, which are available for the authors, who also attended one safety assessment and one security assessment meeting. The authors applied CHASSIS for the same RT concept in a shadow case. The primary aim is to investigate whether the method can be used for a unified process of both safety and security assessments, and to compare it with the regular methods used by the ATM company network. An evaluation of the method by the companies of the network will be organized.

4.2 Applying CHASSIS

The RT example is of particular interest to the authors, since the first and third authors have been participating in the SESAR project with respect to safety. The first author also has more than 15 years of experience from ATM, mainly from tower operations and safety assessments. While the second author has broad experiences from security engineering, the third author has experience in both safety and security. In the example assessment, the first author acted as a domain expert while the other two authors as requirements engineers.

The method starts with the elicitation of functions and services of the RT. Two use cases were defined for our example: *monitoring* by the Air Traffic Control Officer (ATCO) and *providing clearance* by ATCO and Flight Crew (FC). Each use cases can be further detailed (e.g. the subject of monitoring can be weather conditions, aircrafts, etc., while the clearance can be given for different actions like landing, take-off etc.), but we leave them out for simplification. First the D-UC is defined (first/starting diagram in Fig. 2) and then it is detailed in the T-UC (Table 1). This is followed by the creation of regular SDs based on the basic path in the T-UC, but we do not provide them here for space reasons.

The next step elicits safety hazards and security threats in separate D-MUCs using HAZOP guidewords (see Fig. 2). In order to keep the focus on illustrating the CHASSIS method, we present only two misuse cases one for each area in this paper. HAZOP guideword “LATE” is applied for “Provide clearance” for safety, leading to the hazard “Flight crew has wrong clearance”. HAZOP guideword “OTHER” is applied for “Provide clearance” for the integrity aspect of security leading to the threat “Fabrication of false clearance”.

The next step includes the creation of T-MUCs (see Table 2 for safety and Table 3 for security) and the appropriate SDs, FSD for the safety aspect (see Fig. 3) and MUSD for the security aspect (see Fig. 4). The main focus is on the basic path in the T-MUCs since this specifies the harmful events and activities in more detail. This is further detailed and made more specific in the FSD and the MUSD where failures and vulnerabilities, hazards and threats as well as possible mitigations are pointed out. The new details (e.g. mitigations) are fed back to the T-MUCs and D-MUCs (see the grey parts in Table 2 and 3 and the last diagram in Fig. 2). The mitigation use cases become new regular use cases and thus the original D-UC and D-MUC are updated, and ready for a second iteration.

D-MUCs of the two areas are combined into one diagram and conflicts between the mitigations pointed out (see the last diagram in Fig. 2). The mitigation for the safety hazards suggests broadcasting clearances to all aircrafts and thus the flight crews can recognize conflicting clearances. The countermeasure for the security threat suggests making a clearance available only for the targeted aircraft. These two use cases are in conflict, which has to be resolved before the second iteration can start. Our solution is presented in Fig. 5 that can be the base for a second iteration considering possible harms towards the new use cases.

In the final step, HAZOP tables are derived (see Table 4 and 5) from the different techniques, supporting the specification of safety and security requirements. These tables summarize and record the *functions* that were analysed with a specific *parameter* and a *guideword* by the D-MUC. Furthermore, the tables summarize the MUSD and FSD by describing the *consequence* of a failure or an attack can have, and how it evolves in the system until it does some *harm*. This corresponds to the failure and vulnerability notation (including notes). Mitigations in the FSD and MUSD are recorded in the *recommendation* columns in the tables. Compared to the T-MUCs, the HAZOP tables summarize the information from different techniques in a structured and well-known manner, supporting the specification of safety and security requirements. As shown in previous work, it is possible to use FMEA tables together with FSDs [6]. This should however be done at a more detail design level when the system is decomposed into hardware and software components.

When we performed the analysis, the safety aspect was investigated first followed by the security aspect in an iteration step. This is a natural way of thinking to investigate the inadvertent hazards first and then consider how they and other harms can be realized by an intentional entity.

Table 1. Part of the textual template for the "Provide clearance" use case

Name	**Provide clearance**
Iteration	1
Summary	AIR TRAFFIC CONTROL OFFICER (ATCO) clears FLIGHT CREW (FC)
Basic path	bp1. ATCO gives clearance to the FC bp2. If radio communication, FC read-back clearance to ATCO bp3. If radio communication, ATCO hear-back (and checks) clearance from FC
Alternative paths	ap1. In bp2. if DataLink, FC acknowledges clearance from ATCO ap2. In bp3. if DataLink, ATCO receives acknowledgment of clearance from FC.
Exception paths	ep1. In bp3., ATCO realizes that FC read back of clearance is wrong. Return to bp1. to provide clearance again.
Extension points	...
Triggers	1) On request from FC 2) ATCO recognizes AIRCRAFT (ACFT) in a certain position 3) ATCO recognizes the need based on the air traffic flow
Assumptions	1) Communication only through radio communication or DataLink 2) Systems work as expected
Preconditions	...
Postconditions	...
Related business rules	...
Author	CR (domain expert), VK (requirement engineer), PK (req. eng.)
Date	23.02.2012

Table 2. Part of the textual template for the "Flight crew has wrong clearance" safety misuse case (the red/underlined text in the basic path denotes hazard-related issues, while the highlighted grey text denotes information from the second iteration after creating the FSD)

Name	**Provide clearance – Flight crew has wrong clearance**
…	*... (check paper [3] for more details about the left out fields)*
Basic path	bp1. ATCO clears FC1 for take-off from the runway bp2. Communication channel delays the take-off clearance bp3. ATCO gets no answer and thus repeats take-off clearance bp4. ATCO still does not get an answer and requests a radiocheck towards FC-1 bp5. ATCO believes FC1 has not got the take-off clearance and holds its position on taxiway bp6. FC-2 is on short final (approaches runway) and ATCO decides and gives landing clearance for FC-2 bp7. FC-2 reads back the clearance, ATCO hears back the clearance bp8. Delayed take-off clearance for FC-1 arrives bp9. FC-1 reads back the take-off clearance and starts to taxi out on the runway bp10. FC-2 lands while FC-1 is on the runway
Mitigation points	mp1. In bp2., a mechanism for monitoring the accurate functioning of the communication channel should be implemented. mp2. In bp4., ... mp3. In bp5., ... mp4. In bp9., FC-1 hears that FC-2 was cleared to land on the runway and asks for confirmation of take-off clearance mp5. In bp9., FC-2 hears FC-1's delayed take-off clearance and is ready for initiating go-around ...
Assumptions	1) There is only one runway 2) Runway not visible for ATCO (e.g. foggy conditions, camera problems etc.)
Preconditions	1) Two aircrafts, one ready for take-off, one in-bound for landing 2) For mp4. and mp5., broadcast style of communication is needed
Misuser profile	Faulty communication system
Stakeholders, risks	Safety of passengers of the aircrafts is in danger Companies' (air navigation service provider and airliners) reputation is at stake, might go out of business if accident happen
Author	CR (domain expert), VK (requirement engineer), PK (req. eng.)

Table 3. Part of the textual template for the "Fabrication of false clearance" security misuse case (the red/underlined text in the basic path denotes threats-related issues)

Name	**Provide clearance – Fabrication of false clearance**
…	*... (check paper [3] for more details about the left out fields)*
Basic path	bp1. ATCO clears FC2 for landing on the runway bp2. FC-2 reads back the clearance, ATCO hears back the clearance bp3. Attacker eavesdrops the communication and thus knows when to expect FC-2 on the runway bp4. Attacker impersonates ATCO and injects a take-off clearance for FC1 into the communication channel bp5. FC-1 reads back the take-off clearance and starts to taxi out on the runway while the attacker removes the read-back of clearance from communication channel so the ATCO cannot receive it bp6. FC-2 lands while FC-1 is on the runway
Mitigation points	mp1. In bp3., the communication between the ATCO and FC-2 should not be accessible for the attacker ...
Assumptions	1) Broadcast style of communication is assumed ...
Preconditions	... (similarly to the safety template)
Stakeholders, risks	... (similarly to the safety template)

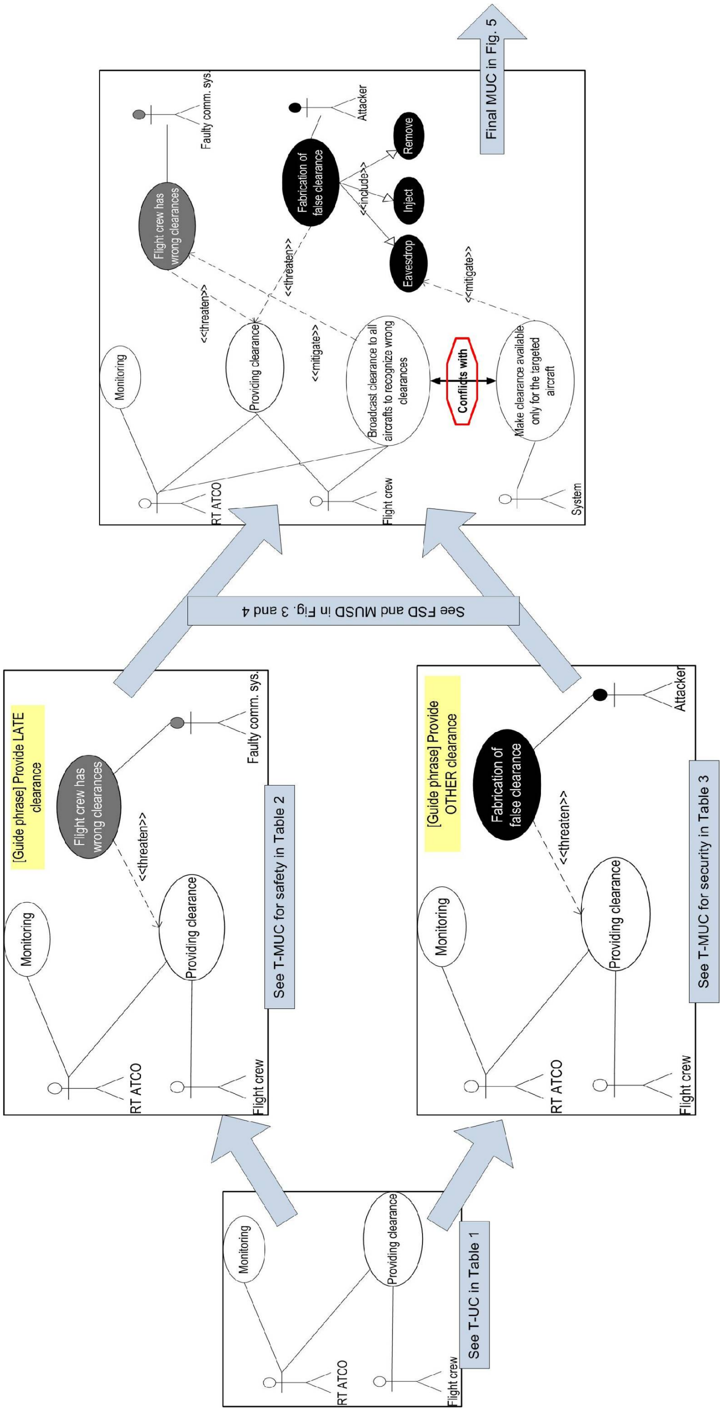


Fig. 2. Use and misuse case development for the Remote Tower example

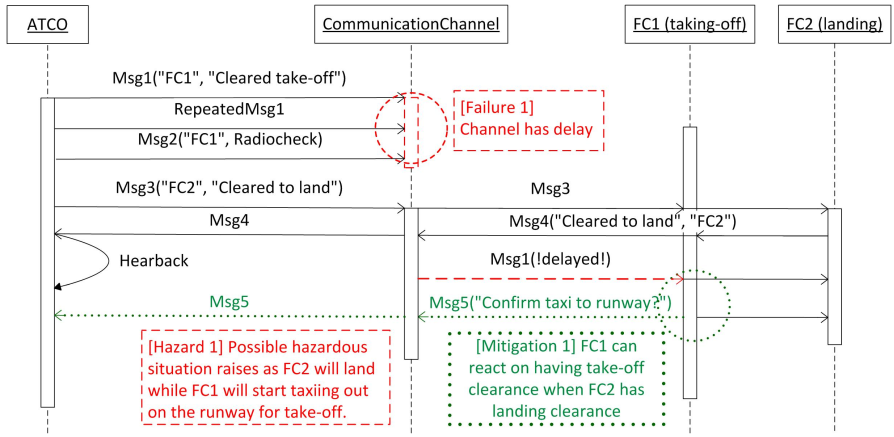


Fig. 3. An example Failure Sequence Diagram for the "Flight crew has wrong clearance" safety misuse case with a suggested mitigation

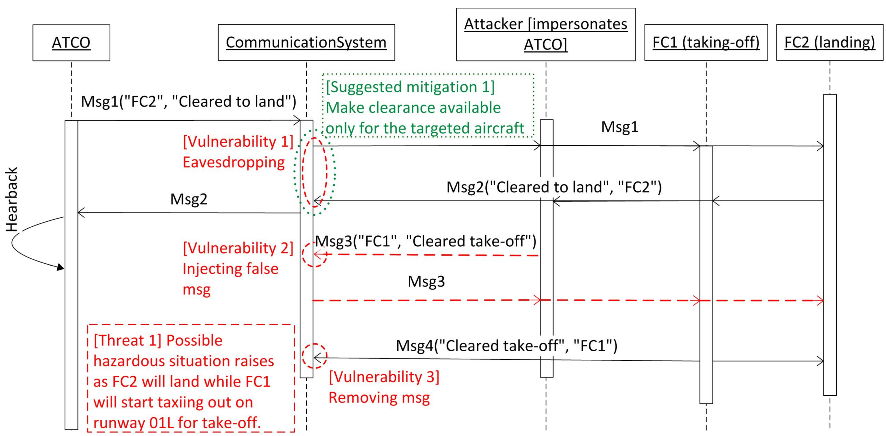


Fig. 4. An example Misuse Sequence Diagram for the "Fabrication of false clearance" security misuse case with a suggested mitigation

Table 4. HAZOP table for specifying safety requirements[2]

Function	Parameter	GW[3]	Consequence	Cause	Harm[2]	Recomm.[3]
Provide clearance	Take-off clearance message	Late	ATCO is not aware of take-off clearance arriving late; gives landing clearance to FC2 instead; FC1 receives late landing clearance and taxi out	Failure comm. system	Two aircrafts are using the runway at same time	Procedure that FCs monitor air traffic on radio and react in case of wrong clearance

[2] Some columns are removed due to space limitations, and *Hazard* is changed to *Harm* in order to address both hazard and threat, which differs from the original HAZOP table.
[3] GW is abbreviation for guideword and Recomm. for Recommendation.

Table 5. HAZOP table for specifying security requirements

Function	Parameter	GW	Consequence	Cause	Harm	Recomm.
Provide clear-ance	Take-off clearance message	Other	ATCO and FCs are not aware of take-off clear-ance injected; both landing and take-off clearances given; FC1 taxi out and FC2 lands on runway	Attacker injects false clearance	Two aircrafts are using the runway at same time	Make clear-ance available for targeted aircrafts only

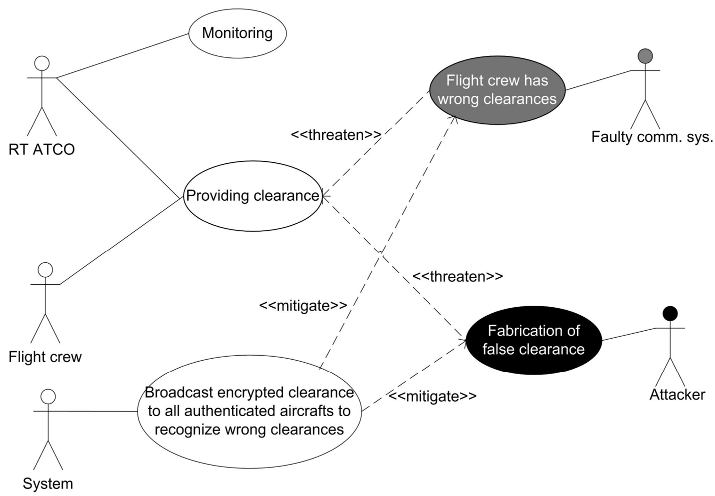


Fig. 5. The final Misuse Case diagram where the conflict between the suggested safety and security mitigations (see Fig. 2) is resolved

5 Discussion

5.1 Rationale behind CHASSIS

Unifying the safety and security assessment has many advantages. For example, since the techniques and many artefacts are common, it saves time and learning efforts of the stakeholders compared to performing two separate assessments. The background knowledge has to be built up only once and common understanding has to be achieved only once as well. It also positively affects later phases as the common arte-facts produced at the end of the requirement analysis phase can be re-used for both areas in design, implementation and testing. When considering safety and security simultaneously, it supports the clarification of their relation, such as which issues can be handled by a common countermeasure or which countermeasures are in conflict. However, for the same reason, the complexity level of the assessment is elevated compared to the complexity of the independent assessments. In addition, problems not foreseen here might appear in practical applications of such a combined method.

CHASSIS supports the above ideas only partially so far as it unites the two as-sessments processes and applies the same (or almost identical) techniques, which enables to represent aspects from both areas together in D-MUCs and produces common artefacts. However, it does not tighten the bond between safety and security more than that, although there is a potential for this, e.g., based on the conceptual model in [13]. This should be a direction for future work.

Another improvement in CHASSIS compared to traditional safety methods is the use of diagrams to facilitate discussions, understanding and creativity. This has advantage for smaller cases, but can lead to overly complex diagrams for larger systems, which cannot be decomposed in a meaningful way. This aspect has to be investigated further as well. To enhance the creative process when identifying hazards and threats to system functions, CHASSIS uses HAZOP guidewords [7]. These guidewords are generic and well suited to use for both safety and security. By using all eighteen guidewords the analyst can enhance stakeholders' creativity and cover many aspects. However, combining all guidewords with all system functions can be time consuming. There must be a trade-off between the size and criticality of the system, together with time available for conducting safety and security assessments. Other techniques that use less guidewords were also considered (e.g. FHA [28]), but we found them less effective for the AT example. This could be however a path for further work.

5.2 Initial Feedback and Preliminary Findings

Our method has been presented to the ATM company network in two different meetings on two consecutive days in order to obtain an initial feedback from safety and security experts. In the first meeting, we presented our method to a safety expert from one of the world-leading suppliers of ATM systems. In the second meeting, the method was presented to security, safety and ATM domain experts from various companies at the end of a security workshop, which was a part of the ATM company network's security assessment activity of the RT case.

The comments by the safety expert (who undertook the safety assessment) from the first meeting mentioned that the method seems to be structured and interesting, especially with respect to combining the safety and security. However, she required good description of each technique to be provided in order to use it by the industry. Furthermore, she pointed out that the strength of the method is to find and visualize sequences and combinations of failures. An overview of the method and its application for security was presented in the second meeting. The experts liked how different threats, vulnerabilities and mitigations were visualized in the MUSD. In their security workshop this was not done, and our method could complement their assessment by representing certain threat scenarios. Furthermore, the experts were interested in how we combined the security with safety. We explained how we used the same techniques in a process where we first analysed the safety aspect followed by the security aspect in an iteration, which they liked. The subsequent discussions also gave important input to us, e.g., the limitations of our technique with respect to quantitative techniques and what should be emphasised more for understanding the method.

The initial feedback from the meetings with the ATM company network shows that the method has the potential for being used by the industry. However, better descriptions of the techniques and the processes are needed. This paper is a first step of providing such a description of the processes. Some of the techniques are well established while some of them have to be detailed further. The MUC and FSD techniques have been applied by the industry before, but the combined use of them has not been described before. So far, we have not taken feedback on the techniques from UML users, which would give us a better understanding of the ease of use with prior knowledge to UML. Furthermore, it is clear that our method is qualitative and does not address certain quantitative parts of the risk assessment, which would be a welcomed addition by the industry.

6 Conclusions and Further Work

A unified process for elicitation and analysis of safety and security requirements was outlined and illustrated in this paper. The CHASSIS method combines safety and security modelling techniques with the aim of transferring their best characteristics and aligning them in a beneficial way. For instance, safety techniques feature systematic assessment processes and means for recording the result, while UML-based techniques apply visual representations and template-based documentation.

Our method was thoroughly illustrated on a real-world example called Remote Tower from the Air Traffic Management domain. There is an ongoing case study using CHASSIS focusing on this example. We plan to evaluate the method from many different aspects like usability, scalability, appropriateness, and combinability with risk management methods and compare it with the methods currently used by the industry. Further work will also include extending our method to support the system design phase. This will be done by including previous work of combining FMEA with FSD [6], and extend this for security with MUSD [25].

References

1. Object Management Group: Unified modeling language (OMG UML), superstructure (2011), `http://www.omg.org/spec/UML/2.4/Superstructure`
2. Watson, A.: Visual modelling: past, present and future (2011), `http://www.uml.org/VisualModeling.pdf`
3. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirement Engineering 10, 34–44 (2005)
4. Sindre, G.: Mal-Activity Diagrams for Capturing Attacks on Business Processes. In: Sawyer, P., Heymans, P. (eds.) REFSQ 2007. LNCS, vol. 4542, pp. 355–366. Springer, Heidelberg (2007)
5. Katta, V., Karpati, P., Opdahl, A.L., Raspotnig, C., Sindre, G.: Comparing Two Techniques for Intrusion Visualization. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 1–15. Springer, Heidelberg (2010)
6. Raspotnig, C., Opdahl, A.: Supporting Failure Mode and Effect Analysis: A Case Study with Failure Sequence Diagrams. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 117–131. Springer, Heidelberg (2012)
7. Ericson, C.: Hazard analysis techniques for system safety. Wiley-Interscience (2005)
8. ATM Bedriftsnettverk: Delievery DA-1.1. Technical report (2011)
9. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis - The CORAS approach. Springer (2011)

10. Giorgini, P., Mouratidis, H.: Secure tropos: A security-oriented extension of the tropos methodology. Journal of Autonomous Agents and Multi-Agent Systems (2005)
11. Lin, L., Nuseibeh, B.A., Ince, D.C., Jackson, M., Moffett, J.D.: Analysing security threats and vulnerabilities using abuse frames. Technical Report 2003/10, The Open University, Walton Hall, United Kingdom (October 2003)
12. Lamsweerde, A.V.: Requirements Engineering - From System Goals to UML Models to Software Specifications. Wiley (2009)
13. Lamsweerde, A.V.: Elaborating security requirements by construction of intentional anti-models. In: Proceedings of the 26th International Conference on Software Engineering, ICSE 2004, pp. 148–157. IEEE Computer Society, Washington, DC (2004)
14. Firesmith, D.G.: Common Concepts Underlying Safety, Security, and Survivability Engineering. Technical Note CMU/SEI-2003-TN-033, Software Engineering Institute (2003)
15. Firesmith, D.G.: A taxonomy of security-related requirements. In: Proceedings of the Fourth International Workshop on Requirements Engineering for High- Availability Systems. RHAS'9205. IEEE Computer Society, Washington, DC (2005)
16. Winther, R., Johnsen, O.A., Gran, B.A.: Security Assessments of Safety Critical Systems Using HAZOPs. In: Voges, U. (ed.) SAFECOMP 2001. LNCS, vol. 2187, pp. 14–24. Springer, Heidelberg (2001)
17. Ministry of Defence: HAZOP studies on systems containing programmable electronics. UK Ministry of Defence Interim Def Stan 00-58 (1994), http://www.dstan.mod.uk/dstandata/ix-00.htm
18. Srivatanakul, T., Clark, J.A., Polack, F.: Effective Security Requirements Analysis: HAZOP and Use Cases. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 416–427. Springer, Heidelberg (2004)
19. Eurocontrol: SESAR ATM preliminary security risk assessment method (2011)
20. Eurocontrol: ATM security risk management toolkit. Technical Report EUROCONTROL-GUID-144, EUROCONTROL ATM Security Domain (2010)
21. ISO/IEC: Information Technology - Security Techniques - Information Security Risk Management ISO 27005
22. Sindre, G.: A look at misuse cases for safety concerns. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences. IFIP, vol. 244, pp. 252–266. Springer, Boston (2007)
23. Stålhane, T., Sindre, G.: A Comparison of Two Approaches to Safety Analysis Based on Use Cases. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 423–437. Springer, Heidelberg (2007)
24. Alexander, I.F.: Initial industrial experience of misuse cases in trade-off analysis. In: Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering, RE 2002, pp. 61–70. IEEE Computer Society, Washington, DC (2002)
25. Raspotnig, C., Opdahl, A.: Improving security and safety modelling with failure sequence diagrams. International Journal of Secure Software Engineering (IJSSE), 20–36 (2012)
26. SESAR Joint Undertaking: About SESAR JU (2012), http://www.sesarju.eu/about
27. Saab, L.F.V.: Advanced remote tower (2012), http://adv.remote-tower.net/
28. Eurocontrol: Air Navigation Safety Assessment Methodology. (electronic) edn. 2.1 (2006)
29. CORAS: The CORAS Method, http://coras.sourceforge.net/

Optimizing Monitoring Requirements in Self-adaptive Systems

Raian Ali[1], Alberto Griggio[2], Anders Franzén[3], Fabiano Dalpiaz[4], and Paolo Giorgini[4]

[1] Bournemouth University, UK
[2] Fondazione Bruno Kessler, Italy
[3] Jasper Design Automation, Sweden
[4] DISI, University of Trento, Italy

Abstract. Monitoring the system environment is a key functionality of a self-adaptive system. *Monitoring requirements* denote the information a self-adaptive system has to capture at runtime to decide upon whether an adaptation action has to be taken. The identification of monitoring requirements is a complex task which can easily lead to redundancy and uselessness in the set of information to monitor and this, consequently, means unjustified instalment of monitoring infrastructure and extra processing time. In this paper, we study the optimization of monitoring requirements. We discuss the case of *contextual goal model*, which is a requirements model that weaves between variability of goals (functional and non-functional requirements) and variability of context (monitoring requirements) and is meant to be used for modelling mobile and self-adaptive systems requirements. We provide automated analysis —based on a SAT-solver— to process a contextual goal model and find a reduced set of contextual information to monitor guaranteeing that this reduction does not sacrifice the system ability of taking correct adaptation decisions when fulfilling its requirements.

Keywords: Requirements, Adaptation, Mobility, Monitoring Optimization.

1 Introduction

Monitoring requirements is a notion which refers to the information that a system has to capture at runtime to assess the effectiveness of its current configuration and, if needed, to determine how to adapt to a more suitable configuration. The monitoring activity requires the deployment of an adequate infrastructure, such as sensors and databases, and collects up-to-date values of certain attributes of the system internal state (e.g., available resources, errors and faults, and security breaches) and the surrounding environment (e.g., user's location, user's computing device, user's movement and activity, temperature). Depending on the collected information, certain adaptation actions may be triggered in order to switch the system into different course of execution.

Monitoring is crucial in several areas and for several reasons. In addition to feeding logging and reporting functionalities, which is essential for the off-line processing established by system analysts, monitoring is a fundamental activity of self-adaptive systems [15]. These systems are autonomously capable of observing their environment and internal state and adapting their behaviour in order to keep their functional requirements fulfilled and optimize the satisfaction of their non-functional requirements.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 362–377, 2012.

Monitoring is a costly activity, both in terms of resources and processing time. The deployment of a comprehensive monitoring infrastructure may be as expensive as the implementation of the core functionalities of the system. This is especially true when considering online monitoring, such as in self-adaptive systems, where the monitoring infrastructure has to continuously collect and process information at runtime. Mobile information systems [14] are a clear example of an adaptive system which is subject to this problem, as some monitoring functionalities have to run on hand-held devices with low processing power and low-capacity battery. In these systems, optimizing monitoring requirements is essential both to reduce costs and to maximize performance.

In our research, we focus on monitoring information that is relevant at the requirements level, namely that affects requirements activation, applicability, and satisfaction. We adopt the notion of *context* to represent any information related to the system environment which affect system's requirements. Specifically, we build on top of *contextual goal models* which we proposed in [1,2], a requirements engineering framework that allows for representing, analysing, and reasoning about the relationship between context and requirements (goals). Our practical experience indicated that the identified and represented contexts can become very complex and this causes redundancy and/or inconsistency problems in both of functional and monitoring requirements.

Example. Let us consider the following three contextual attributes which affect the behaviour of a museum-guide mobile information system: $a1$= “the visitor is in the reception area”, $a2$= “there is enough light in the user's location”, and $a3$= “the user location is not noisy”. Imagine that the context $C1$= $a1 \wedge a2$ is a precondition for a system requirement of showing a demo using certain visualizing settings. In a museum where the light level in the reception area is always high, i.e. $a1 \rightarrow a2$, $C1$ can be reduced to $a1$ and, thus, the system should not monitor the light level and there will be no need to deploy the sensors needed for that. In another case, a behaviour like adopting voice recognition is allowed only in the reception area and feasible when the place is quite, i.e., when $C2$= $a1 \wedge a2$ holds. In a busy museum where the reception is always noisy then $C2$ could be reduced to false and there will be no need to install its monitoring infrastructure unless it helps for monitoring other contexts for other functionalities.

In this paper, we develop a modeling and analysis framework to optimize monitoring requirements in adaptive systems. Our ultimate goal is to minimize the amount and costs of information to monitor in a way that does not compromise the system ability in taking correct adaptation decisions. We study the case of contextual goal models and propose to consider the dependencies between contexts defined on it as an input for our developed SAT-based solver. We develop algorithms to optimize monitoring requirements and explain our approach on a scenario of a mobile information system for assisting visitors of a museum.

The paper is structured as follows. Section 2 explains our baseline: the contextual goal model proposed in [1,2]. Section 3 states and analyzes the context dependency importance for optimizing monitoring requirements. Section 4 presents automated reasoning to treat redundancy, triviality, and inconsistency in a monitoring requirements specification. Section 5 discusses related and future work.

2 Background: Contextual Goal Model

Context, the environment surrounding the system [12], plays an important role in adaptive system engineering, which has to be considered since the early requirements analysis stage. We consider Goal-Oriented Requirements Engineering (GORE), in which stakeholders goals are identified and analyzed, and alternative system behaviours—sets of tasks the system shall implement—are identified to satisfy such goals. In [2,1], Ali et al. advocate the importance of integrating contextual factors in goal analysis, so as to allow for the derivation of contextualized goal satisfaction alternatives. Moreover, the authors explained how context itself needs to be modeled and analyzed. Goal analysis provides constructs to hierarchically analyze goals and discover alternative sets of tasks the system can execute to satisfy such goals, while context analysis provides constructs to hierarchically analyze context and discover alternative sets of facts the system has to monitor to verify a context and then act accordingly.

Fig. 1 represents a contextual goal model for a mobile information system that assists the visitors of a museum and interacts with them and assistance staff, mainly through their personal digital assistants (PDAs). The model represents alternative ways to satisfy the top-level goal of the system: giving information to visitors about the pieces of art in the museum. Contextual annotations ($C1..C8$) express the relationship between context and goals, and are related to the following variation points in the goal model:

1. *Or-decomposition*: a contextual annotation on a branch specifies that an alternative sub-goal (task) can be adopted only if that context holds. For instance, to provide information about a piece of art, a visitor can be directed to a dedicated terminal only if such terminal is free and close to the visitor and he/she is able to use and interact with it $(C2)$. The visitor's PDA can be alternatively used to show information when the piece of the art information are not so complicated, and the visitor has the ability and the knowledge to use PDAs $(C3)$. Getting information through an assistance staff requires that the visitor is not able to use his PDA and not familiar with terminals, or that the visitor is classified as an important visitor $(C4)$.
2. *Root goals* . The activation of a root goal depends on a contextual trigger. To activate goal *"visitor gets informed about a piece of art"*, the system needs to verify if the visitor is interested in the piece, does not already know about it, and if there is still time to explain about it $(C1)$.
3. *Means-end*: goals can be ultimately satisfied by means of specific executable processes (*tasks*). The adoptability of each task in means-end decompositions might depend on the context. The visitor can be notified about the availability of information terminals through a voice message when he/she puts the headphones on, and is not talking to somebody or using his/her PDA for a call $(C5)$, while notifying him/her by SMS can be adopted in the opposite context $(\neg C5)$.
4. *Actor dependency*: An actor can attain a goal or get a task executed by delegating it to another actor only in a specific context. The dependency between the two system actors for providing the information to a visitor through an assistance staff requires a staff that is close to and talks a language common to the visitor, and knows enough about the considered piece of art comparing to the visitor knowledge $(C6)$.

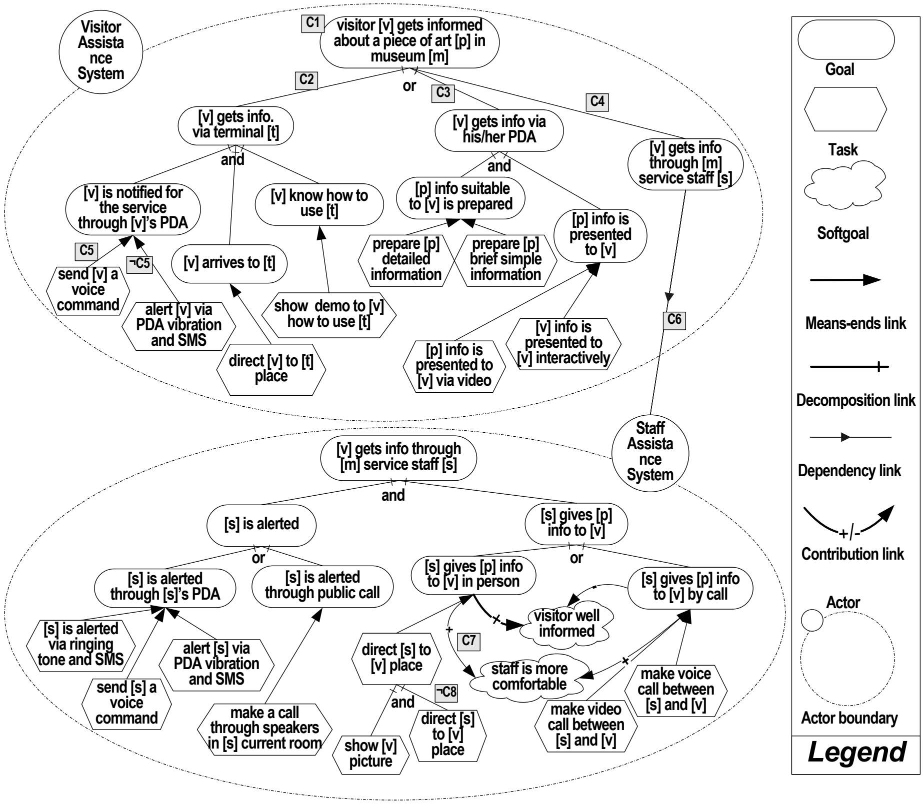


Fig. 1. Tropos goal model for the museum assistance system

5. *And-decomposition*: a sub-goal/sub-task is needed to achieve their parent goal only in a certain context. Guiding the assistance staff to the visitor place is not needed if the visitor stays around and can be seen directly by the assistance staff ($C8$).
6. *Contribution to softgoals*: softgoals are qualitative objectives having no clear-cut satisfaction criteria, and can be contributed either positively or negatively by goals and tasks. The contribution value can vary from one context to another. Giving the information in person is comfortable if the visitor is in the same room as the assistant ($C7$), while it is not comfortable when they are in different rooms.

Contextual precondition on the goal model might need to be analyzed, in order to identify, represent, and agree on how the system can monitor and verify if a context holds. The context analysis proposed in [2,1] provides modeling constructs to hierarchically analyze context. An example of the analysis of the context $C1$ of Fig. 1 is shown in Fig. 2.

Accordingly, context is specified as a formula of world predicates. Based on their verifiability by an actor, world predicates can be either *facts* or *statements*. A world predicate is a fact (a statement) for an actor, if that actor can (cannot) verify it. Verifiability is clearly linked to *monitoring requirements*: a fact requires the deployment

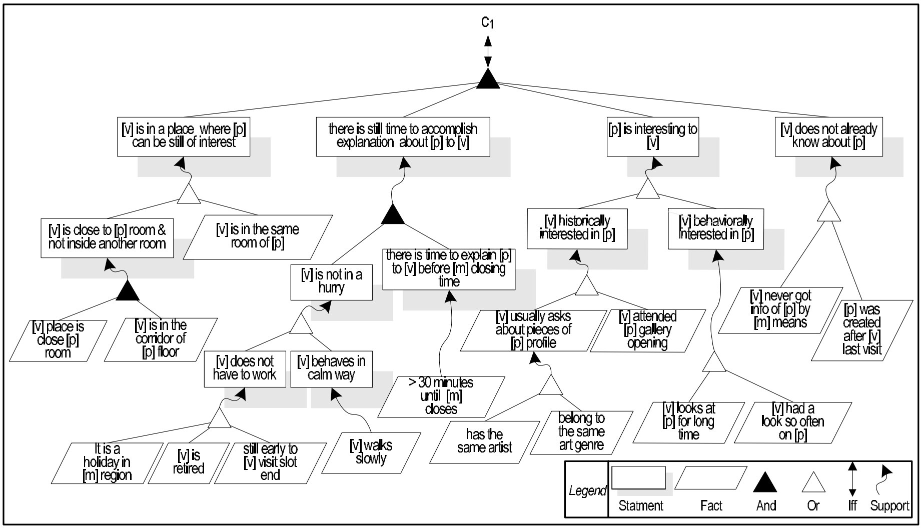


Fig. 2. Context analysis for $C1$: identifying observable facts to judge whether a context holds

of an adequate IT infrastructure (sensors, databases, etc.) to determine its value. Statements, on the other side, cannot be monitored as such: analysts shall further analyze them till reaching observable facts that the system can monitor. Context analysis allows the analyst to specify that a fact (or a formula of facts) is a means to infer evidence that a statement holds. To represent such evidence, context analysis provides the notion of *support* relation.

We explain context analysis constructs with the help of examples from Fig. 2. *"the piece of art [p] artist [a] has lived in the visitor's [v] city of birth"* is a fact, as the system can verify through checking the profile of the artist and the city of birth of the registered visitor. *"the visitor [v] is interested in the piece of art [p]"* is a statement, since the system can not determine its truth value without further analysis. The previous statement is supported if either *"[v] is behaviorally interested in [p]"*, or *"[v] is historically interested in [p]"*. The system can get evidence of the first sub-statement via the support relation from the formula of facts *"[v] looks at [p] for long time"* and *"[v] comes to [p] area and has a look at [p] so often"*.

3 Context Dependency

In this section, we discuss context dependency, show its importance, and argue about its generality and complexity. The context hierarchial analysis we explained in the last section, is easily transformable into a propositional formula consists of the leaf facts as atomic predicates (variables). The dependency, namely the implications, between these facts can make the context formula redundant, trivial, or even inconsistent. Here we give the definition of *redundant* context and its two extremes: the *trivial* and the *inconsistent*.

Definition 1 (Redundant Context) *given the implications between its facts, a context is redundant iff some facts has no effect on its validity.*

Definition 2 (Trivial Context) *given the implications between its facts, a context is trivial iff it is always reduced to true.*

Definition 3 (Inconsistent Context) *given the implications between its facts, a context is inconsistent iff it is always reduced to false.*

Context redundancy makes the context representation more complex without justification and leads to a useless monitoring of facts that have no effect on its validity. Context redundancy motivates us to *optimize monitoring requirements*. Let us take the two facts $f1$ = *"the visitor is inside a museum room"* and $f2$ = *"there is enough light at the location of the visitor"*. Consider a context $C = f1 \wedge f1$; if the system is going to operate in a museum that its rooms are well illuminated then $f1 \rightarrow f2$ and C is reduced to $f1$ which means that there is no need to install sensors to capture the level of light in the museum. Alternatively, if $C = f1 \vee f2$ then C is reduced to $f2$ and there will be no need for installing a positioning system to decide if the visitor is inside a museum room. Some base reductions rules are shown in the following table:

Assured Implication	$A \vee B \leftrightarrow$	$A \wedge B \leftrightarrow$
$A \rightarrow B$	B	A
$A \rightarrow \neg B$	$A\ xor\ B$	$false$
$A \leftrightarrow \neg B$	$true$	$false$

While the redundancy of context implies a redundancy in monitoring requirements; context inconsistency adds to that inapplicability in the software functionalities. Besides the uselessness of monitoring their facts, inconsistent contexts deny the adoptability of the software functionalities preconditioned by them. E.g., if a functionality is preconditioned by the context $C = f1 \wedge f2$, and if the museum rooms are not well illuminated for some decoration reasons or to conserve the quality of a piece of art, i.e. $f1 \rightarrow \neg f2$, then such functionality is never adoptable since C is inconsistent.

After showing its influence, now we argue about the generality of the context dependency problem and that it is not tied to or caused by our context analysis and goal model. We expect any self-adaptive system to monitor several pieces of information regarding its context that could be also combined through logical relations to conform logical expressions. Assuring some implications between these information pieces might reveal a problem of redundancy, triviality, or inconsistency. Let us take the following generic pseudo-code that can be part of a decision making process of a self-adaptive system:

if $(A \vee B) \wedge C$ **then**
    if $D \wedge E$ **then**
        adopt alternative set of actions (action_set 1)
    else
        if $F \vee G$ **then**
            adopt another set of actions (action_set 2)
        end if
    end if
end if

The three contexts boolean abstractions we have here are $(A \vee B) \wedge C$, $D \wedge E$, and $F \vee G$, that involve monitoring the set of facts $S = \{A, B, C, D, E, F, G\}$. The model has two alternative set of actions *action_set 1* and *action_set 2* each fitting to a certain context. When we assure the implication $C \rightarrow A$ then $(A \vee B) \wedge C$ can be reduced to just C and therefore there will be no need to monitor A and B. If there is no implication between D and E then $D \wedge E$ alone is not redundant, but this does not mean that *action_set 1* is adoptable or D and E are not redundant; suppose we have the implication $C \rightarrow \neg E$, then the accumulated context at line 2, which is $C \wedge D \wedge E$, is inconsistent and reduced to *false*, and the *action_set 1* is not adoptable. In case we assure that $C \rightarrow \neg G$ then the accumulated context at line 5, $C \wedge (F \vee G)$, can be reduced to $C \wedge F$ which means that G is redundant and has no effect on the validity of the accumulated context.

The implications between facts can be *absolute* or *dependent* on the characteristics of the system environment. Absolute implications are applied wherever the system has to operate, e.g., $f3 \rightarrow \neg(f4 \wedge f5)$ where $f3 =$*"piece of art [p] is always exclusive to museum [m]"*, $f4 =$*"visitor is visiting the museum [m] for the first time"* and $f5$=*"visitor has seen [p] some date before today"*. Other implications depend on the nature of the environment the system is going to operate in, like the aforementioned implication between the light level and being inside a museum room. Moreover, the environment itself assures that some contexts are always true or false, so we have to consider a special kind of implications between the system environment and context analysis facts, i.e. $Env \rightarrow facts\_formula$. E.g., if the museum opens only in holidays, so the fact *"the day is holiday in museum region"* is always true.

Facts verification has costs; costs are those related to the facts verification process itself and to getting the data needed to make the verification possible, such as installing physical equipments like sensors, inserting data by human operators, having enough storage, processing time and so on. When we have more than one possibility to reduce contexts, we should choose the one that minimizes the costs.

After the above explanation of the context dependency and its effects, we now explain two main specific problems that we need to face in order to optimize monitoring requirements:

– *Optimizing monitoring requirements*: checking and fixing the redundancy, triviality, and inconsistency of contexts lead to minimizing the costs of the system as it avoids us sensing, storing, processing data to verify facts that have no effect on any decision, and developing software functionalities that are preconditioned by inconsistent contexts until such inconsistencies are fixed.

  Let us consider the contextual goal model of the top-left of Fig. 3. Whenever the analyst defines a direct context at each variation point ($C1$, $C2$, $C3$, $C4$, $C5$, $C6$), the automated reasoning has to check if this direct context alone and if the accumulated context ($C1$, $C1 \wedge C2$, $C1 \wedge C3$, $C1 \wedge C4$, $C1 \wedge C5$, $C1 \wedge C3 \wedge C6$ respectively) are consistent and non trivial, and notify the analyst to fix any error before repeating the check and proceeding with the next contexts. However, this check has also to be done progressively for the accumulated context on the alternatives in the goal model to know if they can be adopted together, e.g., if $C1 \wedge C2 \wedge C5$ is inconsistent then the root goal satisfaction alternative $A2$= $\{G5, G3, G8\}$ is never adopted.

After defining contexts at all of the variation points and passing the consistency and triviality check, we can start to optimize the monitoring requirements. Optimization takes the set of all contexts associated to the different goal satisfaction alternatives $\{A1, A2, A3, A4\}$and softgoals $\{SG1\}$, i.e. $\{A1C, A2C, A3C, A4C\}$, and $\{SG1C\}$ and gives equivalent reduced contexts $\{A1C', A2C', A3C', A4C'\}$, and $\{SG1C'\}$ that can be verified on a sub-set of facts with minimum monitoring costs. The analyst has to study the set of facts of the resulted formulas and elicit the data that the monitoring system has to obtain.

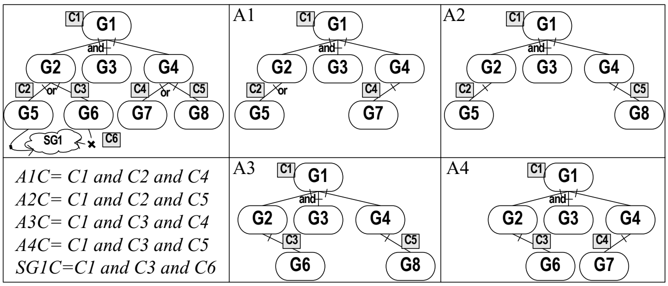


Fig. 3. The accumulated context for the root goal satisfaction alternatives and softgoals

The automated reduction has to minimize the total cost of monitoring all the reduced contexts facts, as doing it separately for each context of $\{A1C, A2C, A3C, A4C\}$, and $\{SG1C\}$ does not guarantee, in the general case, having the minimal total monitoring costs. The problem of optimizing a set of contexts together to reduce the overall cost is highly expensive as we explain later. In the next section, we provide an algorithm, based on SAT techniques and greedy algorithms, that takes a context formula together with the implications between facts (assumptions), checks its consistency and redundancy, and produces an equivalent formula with less costs.

– *Efficient specification of implications*: when the number of facts is high, it will be hard for the analyst to specify even the binary implications between facts. Moreover, the specified implications themselves might be wrong and inconsistent. Designing a supporting tool that helps the analyst to correctly specify, with a minimum number of interactions, the implications between facts is another main problem. We expect such tool to make a kind of facts analysis and asks the analyst to specify the relation where the probability of implication is high. In this paper, we do not provide solution for this problem and we aim to address it in future work.

4 SAT-Based Redundancy Elimination

In this section we describe our algorithm for determining whether a context is inconsistent or trivial, and for identifying redundant facts in a context. The algorithm is based on propositional satisfiability (SAT), and in particular on SAT-based techniques for the enumeration of all the models of a propositional formula. Before describing the algorithm, we recall some necessary definitions and notions from propositional logic.

Definition 4 (Model, Satisfiability, Equivalence) *Let φ be a propositional formula, and $V(\varphi)$ be the set of its atomic predicates. Let μ be a function $\mu : V(\varphi) \rightarrow \{0,1\}$, and let ν be a function from propositional formulas to the set $\{0,1\}$ defined as:* [1]

$$\nu(P) = \mu(P), P \in V(\varphi) \qquad \nu(\neg\phi) = 1 - \nu(\varphi) \qquad \nu(\phi \wedge \psi) = min(\nu(\phi), \nu(\psi))$$

μ is a model *for φ if $\nu(\varphi) = 1$. φ is* satisfiable *if it has at least one model,* unsatisfiable *otherwise.*

Two formulas ϕ and ψ are equivalent *if and only if they have the same models. A formula ϕ* entails *another formula ψ, denoted as $\phi \models \psi$, if all the models of ϕ are also models of ψ, but not vice versa.*

In what follows, we might denote a model μ as a *set of literals* μ_S, such that for each variable P, if $\mu(P) = 1$ then $P \in \mu_S$, and if $\mu(P) = 0$ then $\neg P \in \mu_S$. Analogously, we might denote μ as a *formula* μ_F which is the conjunction of the literals in μ_S.[2]

Example 1. Let φ be the formula $(P \vee Q) \wedge (R \vee \neg S) \wedge (S \vee P)$. Then $\mu := \{P, Q, \neg R, \neg S\}$ is a model for φ.

Definition 5 (Equivalence under assumptions) *Let ξ and φ be two formulas. Then a formula φ' s.t. $\xi \models \varphi \leftrightarrow \varphi'$ is said to be equivalent to φ under the assumption of ξ.*

Example 2. Let P and Q be predicates. Given the definition of equivalence under assumptions, $P \wedge Q$ is equivalent to P under the assumption $P \rightarrow Q$ since $P \rightarrow Q \models (P \wedge Q) \leftrightarrow P$. There are other formulas which will be equivalent, e.g. $P \wedge Q$ is trivially equivalent to itself.

By substituting every fact (a predicate) in a context with a fresh propositional variable (*fact variable*) we obtain the *boolean abstraction* of a context. In the same way, we can obtain the boolean abstraction of the assumptions which are known to be true in a context. Given the boolean abstraction for a context φ and the corresponding assumptions ξ we can express the problem of reducing redundancy of contexts as the problem of finding an equivalent context φ' which is equivalent to φ under the assumptions ξ.

In order to obtain such a φ', we exploit SAT solvers, and in particular techniques for generating all the models of a boolean formula. The pseudo-code of our algorithm is reported in Fig. 4. The algorithm enumerates the models of the boolean abstraction φ of the context, and for each such model μ it checks whether μ is compatible with the assumptions ξ (which express the known implications between facts). If μ is compatible with the assumptions, the algorithm tries to reduce μ by removing literals from it as long as it is still a model for φ *under the given assumptions*, that is, as long as $\mu \wedge \xi \models \varphi$, or in other words as long as $\mu \wedge \xi \wedge \neg\varphi$ is unsatisfiable. Then, the reduced context is given by taking the disjunction of all the reduced models that are compatible with the assumptions.

Theorem 1. *Let ξ and φ be two formulas, and let φ' be the result of applying the algorithm of Fig. 4 to φ and ξ. Then φ' is equivalent to φ under the assumptions ξ.*

[1] We define ν only for the connectives $\neg$, $\wedge$ since they are enough to express all the others.
[2] Moreover, we shall drop the subscripts $_S$ and $_F$ when they are clear from the context.

Input: context φ, assumptions ξ
Output: reduced context φ'
1: $\varphi' \leftarrow \bot$
2: **for all** models μ of φ **do**
3:   **if** Is_Satisfiable($\mu \wedge \xi$) **then**
4:     **for all** literals $l \in \mu$ **do**
5:       $\mu' \leftarrow \mu \setminus \{l\}$
6:       **if not** Is_Satisfiable($\mu' \wedge \xi \wedge \neg\varphi$) **then**
7:         $\mu \leftarrow \mu'$
8:       **end if**
9:     **end for**
10:     $\varphi' \leftarrow \varphi' \vee \mu$
11:   **end if**
12: **end for**
13: **return** φ'

Fig. 4. Pseudo-code of the context reduction algorithm

Proof. We have to show that:

1. every model of φ that is compatible with ξ is also a model of φ'; and
2. for each model μ of φ', all its extensions to all the variables in $V(\varphi) \setminus V(\varphi')$ that are compatible with ξ are models of φ.

– Let μ be a model of φ compatible with ξ (that is, $\mu \wedge \xi$ is satisfiable). Then, by construction (lines 4-10 of the algorithm) φ' contains a subset of μ as a disjunct. Therefore, μ is a model for φ'.
– Let μ be a model of φ' compatible with ξ. Since φ' is a disjunction of conjunctions of literals, μ must be a superset of the set of literals σ in one of such conjunctions. We can assume w.l.o.g. that σ is the smallest such set, because clearly if μ satisfies $\psi \wedge l$, then μ satisfies also ψ. Moreover, the variables occurring in μ are a subset of the variables of φ. Consider any extension μ' of μ to all the variables of φ, such that μ' is compatible with ξ, and suppose that μ' is not a model for φ. Then μ' can be turned into a model for φ by flipping [3] some of the literals in $\mu' \setminus \sigma$, since by construction the literals in σ occur in a model of φ compatible with ξ (lines 3-10 of the algorithm). Let η be a minimal set of literals to flip to obtain a model μ'' of φ from μ'. By construction, $\mu'' \wedge \xi \wedge \neg\varphi$ is unsatisfiable, and for all the literals l in η, $(\mu'' \setminus \{l\}) \wedge \xi \wedge \neg\varphi$ is satisfiable. [4] But then, none of the literals in η would have been removed from μ'' by the algorithm (lines 4-9) when processing μ'' (which must have been processed since it is a model of φ), and so η must be a subset of σ, which is a contradiction. Therefore, μ' is a model for φ.

Example 3. Let the context be $\varphi = (P \wedge Q) \vee R$, and we wish to reduce this formula under the assumption $\xi = (P \rightarrow \neg Q) \wedge (P \rightarrow R)$

[3] Flipping a literal here means replacing l with $\neg l$ or vice versa.

[4] Because $((\mu'' \setminus \{l\}) \cup \{\neg l\}) \wedge \xi \not\models \varphi$, so $((\mu'' \setminus \{l\}) \cup \{\neg l\}) \wedge \xi \wedge \neg\varphi$ is satisfiable, and so also $(\mu'' \setminus \{l\}) \wedge \xi \wedge \neg\varphi$ is satisfiable.

To obtain a reduced context we can enumerate all models of φ and reduce given the assumption in this way:

1. We set up the algorithm by setting $\varphi' \leftarrow \bot$
2. φ is satisfiable, and the first model returned is e.g. $\mu = \{\neg P, \neg Q, R\}$
   (a) $\neg P \wedge \neg Q \wedge R \wedge ((P \rightarrow \neg Q) \wedge (P \rightarrow R))$ is satisfiable (line 3), so the model is compatible with the assumptions.
   (b) Since $R \wedge (P \rightarrow \neg Q) \wedge (P \rightarrow R) \wedge \neg((P \wedge Q) \vee R)$ is unsatisfiable, both $\neg P$ and $\neg Q$ are redundant in this model, so they are removed from μ' in lines 4-9 of the algorithm.
   (c) Update $\varphi' \leftarrow R$
3. the second model of φ returned is e.g. $\mu = \{P, Q, \neg R\}$
   (a) As $P \wedge Q \wedge \neg R \wedge ((P \rightarrow \neg Q) \wedge (P \rightarrow R))$ is unsatisfiable (line 3), the model is not compatible with the assumptions, so we skip lines 4-10.
4. the other two models returned are $\mu = \{P, \neg Q, R\}$ and $\mu = \{\neg P, Q, R\}$. As above, they can be reduced to $\{R\}$ only, since $R \wedge (P \rightarrow \neg Q) \wedge (P \rightarrow R) \wedge \neg((P \wedge Q) \vee R)$ is unsatisfiable (line 6).

The resulting reduced context becomes $\varphi' = R$, and we have found that P and Q are redundant for this context.

We remark that the above algorithm can be used also to detect inconsistent or trivial contexts: in the former case, none of the models will be compatible with ξ, so φ' will be always equal to $\bot$; in the latter case, $\xi \wedge \neg\varphi$ will be always unsatisfiable, so in the loop of lines 8-10 all the literals would be removed from μ, which will therefore be reduced to $\top$. However, for efficiency reasons it might be preferable to check for inconsistency and triviality before entering the main loop of lines 2-12, by checking the unsatisfiability of the formulas $\xi \wedge \varphi$ and $\xi \wedge \neg\varphi$ respectively.

Efficiency of the Algorithm. The algorithm enumerates all models, and in the worst case there are an exponential number of them. For each model, we solve a number of SAT problems. So in the worst case, we need to solve an exponential number of NP-complete problems.

Despite this, the cost of the calls to a SAT procedure can be greatly reduced by using an incremental SAT solver such as MiniSat [9]. The call on line 3 will use the same formula ξ in every iteration of the outer loop, only varying the model μ. In this case, one single solver instance containing ξ can be reused from one iteration to the next. In the same way, the call on line 6 uses the same formula $\xi \wedge \neg\phi$ in each call, only varying the model μ. A single SAT solver instance can be reused for all these calls. The advantage of using an incremental SAT solver for each of these three cases is that everything learnt from the formulas in one iteration of the outer loop can be reused for all following iterations and will not have to be rediscovered. Lastly, enumerating all models can be done with an efficient algorithm for the all-SAT problem.

Further optimizations are possible. E.g. the number of iterations in the loop enumerating all models on line 2 can be reduced by *blocking clauses* gained from the reduction. We can conjunct the negation of the reduced model μ computed on lines 4–9 to the formula ϕ after each iteration. In example 3 above, this improvement would remove the two last iterations.

4.1 Greedy Strategies for Cost Reduction

In the problem of reducing contexts, we wish to remove redundant facts from the context. This corresponds to producing a formula φ' with less variables than φ. In fact, we want to reduce the *cost* of monitoring facts in a context. If we associate a cost (a real number) to each fact variable in the boolean abstraction of a context, our aim is that of finding a φ' such that the sum of the costs of the variables occurring in φ' is smaller than the sum of the costs of the variables occurring in φ.

As presented, our context reduction algorithm does not take costs into account. One simple possibility to make it aware of costs is to apply some *greedy* strategies when determining the order in which variables are eliminated from the current model (line 4 of Fig. 4). For example, one strategy could be to sort the variables in the model μ according to their cost, to try to eliminate more expensive variables first. A more sophisticated strategy could also consider whether a variable already occurs in the current φ' constructed so far, to try to keep the set of variables $V(\varphi')$ as small as possible.

Example 4. Consider the following context formula φ and assumptions ξ:

$$\varphi = ((\neg P \vee \neg R) \wedge (\neg Q \vee \neg S)) \vee (\neg P \wedge S)$$
$$\xi = (P \leftrightarrow Q) \wedge (R \leftrightarrow S) \wedge (S \rightarrow Q)$$

Suppose that the cost of P is 3, that of Q is 1, that of R is 5 and that of S is 4. Moreover, suppose that the first model found by the algorithm of Fig. 4 that is compatible with ξ is $\mu_1 = \{P, Q, \neg R, \neg S\}$.

If the algorithm does not consider costs, μ_1 might get reduced in lines 4-9 to $\{\neg R\}$. Therefore, after the first iteration of the loop of lines 2-12, $\varphi' = \neg R$. The only other model of φ that is compatible with ξ is $\mu_2 = \{\neg P, \neg Q, \neg R, \neg S\}$. In this case, μ_2 might get reduced to $\{\neg P\}$, and thus the resulting reduced context φ' would be $\neg P \vee \neg R$, whose cost is 8.

However, if costs are considered, in the process of reducing μ_1 and μ_2 the algorithm would try to eliminate first the more expensive variables, resulting in the reduced models $\{\neg S\}$ and $\{\neg Q\}$ respectively. Therefore, in this case the reduced context φ' would be $\neg S \vee \neg Q$, whose cost is 5.

Finally, if the algorithm takes into account also the presence of variables in the current φ', in the process of reducing μ_2 the order in which literals are processed in the loop of lines 4-9 would be $\neg R, \neg P, \neg Q, \neg S$, as S is already in φ' (because of μ_1). With this order, also μ_2 would be reduced to $\{\neg S\}$, and so in this case the final φ' would be $\neg S$, whose cost is only 4.

Efficiency of the Algorithm. The algorithm has the same complexity as the algorithm without costs, since we are only modifying the order in which we try to eliminate variables. We can therefore expect similar performance.

4.2 Finding an Optimal Solution and Reducing Multiple Dependent Contexts

The algorithm of Fig. 4 (and its greedy variant) computes *one* reduction for the input context formula, but it does not find (in general) the reduction with minimum cost.

Input: context φ, assumptions ξ
Output: reduced context φ'
1: $\varphi' \leftarrow \varphi$
2: **for all** subsets S of variables $V(\varphi)$ **do**
3: **for all** formulas $\psi(S)$ over S **do**
4: **if** $\xi \models \psi(S) \leftrightarrow \varphi$ **then**
5: **if** cost of $\psi(S)$ is lower than cost of φ' **then**
6: $\varphi' \leftarrow \psi(S)$
7: **break**
8: **end if**
9: **end if**
10: **end for**
11: **end for**
12: **return** φ'

Fig. 5. Naive algorithm for finding the context with minimum cost

Clearly, finding such optimal context wrt. costs would be very desirable. However, solving this problem is far from trivial. A naive algorithm/solution for it is shown in Fig.5.

This algorithm works by enumerating up to *all* the formulas $\psi(S)$ that are equivalent to the context formula φ under the assumptions ξ, and picking the one with minimum cost. Such exhaustive enumeration is prohibitively costly: the outer loop of lines 2-10 is executed $2^{|V(\varphi)|}$ times, and, since the number of different boolean formulas over k variables is 2^{2^k}, the inner loop of lines 3-9 is executed up to $2^{2^{|S|}}$ times. Moreover, checking whether $\psi(S)$ and φ are equivalent under the assumptions ξ (line 4) is an NP-complete problem. Therefore, the naive algorithm would require to solve up to $\sum_{S \in 2^{V(\varphi)}} 2^{2^{|S|}}$ NP-complete problems.

In practice, the algorithm can be improved by performing a branch-and-bound search [13] (on the sum of costs of the variables) instead of enumerating all the subsets of variables, thus avoiding to enumerate (and check) formulas over variables whose cost is known to be higher than the best solution found so far. However, in the worst case the complexity would not improve.

5 Discussion and Future Work

In this paper, we have proposed a framework to optimize monitoring requirements, so as to minimize the monitoring infrastructure a system has to deploy. We have presented the context dependency problem, which may lead to trivial, redundant, and inconsistency in monitoring requirements, and proposed algorithms to detect such problems as well as to suggest ways to fix redundancies. In our approach, monitoring requirements are identified as a result of contextual goal analysis. First, relevant contexts are identified by means of contextual goal models [2]. Second, the contexts in such models are analysed, via context analysis technique, in order to identify monitorable facts, which constitute the monitoring requirements. Such requirements can suffer from different problems; they can be (i) redundant, if the monitoring infrastructure is going to observe more data than necessary; (ii) inconsistent, if a context to be monitored is always false; (iii) trivial, if a context to be monitored is always true. In order to detect these problems and to fix them, by producing an equivalent monitoring requirements specification with reduced cost, we propose techniques based on state-of-the-art SAT-solvers. Research on monitoring requirements (and their optimization) is still at an early stage. To the best of our knowledge, ours is the first approach that argues for the importance of optimizing monitoring requirements in adaptive systems by detecting redundancies, inconsistencies, and trivialities. Thus, we will also consider literature on requirements monitoring and contextual requirements, which provides useful insights for our research.

Salifu et al. [16] have clarified the importance of considering monitoring requirements. In particular, they relate monitoring requirements to what an application shall monitor in order to check whether a requirement has failed or is being met. They consider the existence of alternative means to perform monitoring, and the choice among such alternatives. Our approach, instead, focuses on minimizing the amount of information to be monitored: this is a particularly important concern when considering mobile and adaptive systems with a large space of variations (alternatives).

Requirements monitoring is about insertion of a code into a running system to gather information (mainly about computational performance), so as to determine whether the running system is meeting its design objectives and reconcile the system behaviour to requirements if a deviation occurs [11]. In [10], the GORE (Goal-Oriented Requirements Engineering) framework KAOS [8] is integrated with an event-monitoring system (FLEA [6]) to provide an architecture that enables the runtime automated reconciliation between system goals and system behaviour with respect to a priori anticipated or evolving changes of the system environment. Their work does implicitly specify monitoring requirements; thus, our approach could be applied to their modelling framework in order to minimize the monitoring infrastructure to be deployed.

Wang et al. [17] propose an approach to requirements monitoring based on goal models, wherein tasks and goals are associated with pre- and post-condition. A failure occurs if (i) a post-condition is met while the respective pre-condition does not hold, or (ii) an event representing a precondition occurs and at the next time-step the postcondition does not hold. They argue for and show the importance of requirements monitoring trade-off, especially when their approach is applied to multi-layer monitoring, e.g. in service-oriented architectures. They conduct experiments on how different granularity of monitoring affects the accuracy of the diagnosis. Our approach focuses on minimizing the amount of contextual information to monitor, while leading to optimal diagnosis.

Baresi et al. [5] propose Dynamo, an approach that provides dynamic monitoring in web services. In their framework, monitoring requirements are specified by an analyst an monitorable rules. Dynamic monitoring is provided by the monitoring manager component: depending on the current context, the component decides whether a rule is to be monitored or not. Like the previously mentioned approaches, however, they do not focus on identifying a minimal monitoring infrastructure and contextual information which do not sacrifice the system ability of taking correct adaptation decisions.

In future work, we aim to develop a supporting tool for our framework that assists the analysts for building correctly our proposed models and simulating the system behaviour. We plan to integrate our techniques in frameworks for requirements-driven self-adaptive software; a good candidate is the work by Dalpiaz et al. [7], which already captures the relation between context and requirements. The role of users in monitoring a system's environment and quality is being recognized through the concepts of Social Sensing [4] and Social Adaptation [3]. We also plan to allow users to report meta-data about the data they are asked to monitor so we can further optimize monitoring requirements. Moreover, we will apply our framework to case studies to understand how our optimization techniques save costs in practice and reduce their complexity.

Acknowledgement. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grants n o 257930 (Aniketos), 256980 (Nessos), and 258109 (FastFix), and Science Foundation Ireland under grant 10/CE/I1855 and also from Provincia Autonoma di Trento and the European Community's FP7/2007-2013 under grant agreement Marie Curie FP7 - PCOFUND-GA-2008-226070 "progetto Trentino", project ADAPTATION.

References

1. 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.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 326–338. Springer, Heidelberg (2009)
2. Ali, R., Dalpiaz, F., Giorgini, P.: A Goal-based Framework for Contextual Requirements Modeling and Analysis. Requirements Engineering 15, 439–458 (2010)
3. Ali, R., Solis, C., Omoronyia, I., Salehie, M., Nuseibeh, B.: Social Adaptation: When Software Gives Users a Voice. In: Proceedings of ENASE 2012 (2012)
4. Ali, R., Solis, C., Salehie, M., Omoronyia, I., Nuseibeh, B., Maalej, W.: Social Sensing: When Users Become Monitors. In: Proceedings of ESEC/FSE 2011, pp. 476–479 (2011)
5. Baresi, L., Guinea, S.: Towards Dynamic Monitoring of WS-BPEL Processes. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 269–282. Springer, Heidelberg (2005)
6. Cohen, D., Feather, M.S., Narayanaswamy, K., Fickas, S.S.: Automatic Monitoring of Software Requirements. In: Proceedings of ICSE 1997, pp. 602–603 (1997)
7. Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Adaptive Socio-Technical Systems: a Requirements-driven Approach. In: Requirements Engineering (to appear, 2012)
8. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1-2), 3–50 (1993)
9. Eén, N., Sörensson, N.: An Extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
10. Feather, M.S., Fickas, S.S., van Lamsweerde, A., Ponsard, C.: Reconciling System Requirements and Runtime Behavior. In: Proceedings of the 9th International Workshop on Software Specification and Design (IWSSD 1998). ACM, New York (1998)
11. Fickas, S.S., Feather, M.S.: Requirements Monitoring in Dynamic Environments. In: Proceedings of RE 1995, p. 140. IEEE Computer Society (1995)
12. Finkelstein, A., Savigni, A.: A Framework for Requirements Engineering for Context-Aware Services. In: Proceedings of STRAW 2001 (2001)
13. Hillier, F.S., Lieberman, G.J.: Introduction to Operations Research, vol. 6. McGraw-Hill, New York (1990)

14. Krogstie, J., Lyytinen, K., Opdahl, A.L., Pernici, B., Siau, K., Smolander, K.: Research Areas and Challenges for Mobile Information Systems. International Journal of Mobile Communications 2(3), 220–234 (2004)
15. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems 4, 14:1–14:42 (2009)
16. Salifu, M., Yu, Y., Nuseibeh, B.: Specifying Monitoring and Switching Problems in Context. In: Proceedings of RE 2007, pp. 211–220 (2007)
17. Wang, Y., McIlraith, S., Yu, Y., Mylopoulos, J.: Monitoring and Diagnosing Software Requirements. Automated Software Engineering 16(1), 3–35 (2009)

Identifying Safety Hazards: An Experimental Comparison of System Diagrams and Textual Use Cases

Tor Stålhane and Guttorm Sindre

Norwegian University of Science and Technology
{stalhane,guttors}@idi.ntnu.no

Abstract. As ICT is increasingly used in critical systems, safety is a growing concern. Safety hazards should be discovered and handled at an early stage of IS development, since it is much more expensive to redesign a system post hoc due to threats that were initially overlooked. It is therefore interesting to integrate safety analysis with textual and diagrammatic specifications used in mainstream system development. This paper reports on an experiment comparing how well system diagrams and textual uses cases support the identification of hazards in a simple railway control system. The two most important conclusions are that textual uses cases are as good as or better than system diagrams for hazard identification in all cases except for peripheral equipment and that including system diagrams in the documentation is not enough − they must be brought into focus for the analysis.

Keywords: safety, use case, requirements engineering, systems development, hazard identification.

1 Introduction

While the number of IT systems that can have safety implications increase, safety analysis is already a bottleneck in software development [1], and an international embedded systems developer we have collaborated with report as much as 30% of software development man-hours spent on certification. Most software development companies choose one of two ways out of this. Either they suspend all safety analysis until the final acceptance phase – e.g., customer acceptance – or they skip safety analysis altogether. Neither alternative is particularly attractive. The former may cause the system to be rejected by a certifying authority, or cause a lot of extra costs for necessary redesign [2], while the latter is dangerous.

Hence, it is feasible to introduce HazId (hazard identification) techniques that the developers and customers can apply early in the development process [3-4], in order to identify possible hazards and thus include design changes and safety barriers. This approach will be most efficient if it is applied in the concept and requirements engineering phases of the project [5-6]. If waiting until later stages, expensive rework may be necessary to handle the safety issues [7].

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 378–392, 2012.

Based on our experience, the basic requirements for a HazId technique to be used with stakeholders who are not safety experts are as follows:

- Easy to learn and not require any previous experience in safety analysis. It must be possible to quickly teach it to customers and to novice developers.
- Able to work with only high level descriptions of the system to be built – e.g., a system diagram or a set of use cases or user stories.
- Able to identify important hazards

In previous work [8-10] we have already made some investigations into candidate techniques. In [8] we made an experimental comparison of two approaches, one using a combination of use case diagrams and Failure Mode and Effect Analysis (FMEA) [11] tables, and another using a combination of use and misuse case diagrams [12]. In [9] we then compared the use of textual use (and misuse) cases with diagrams. The objective in each experiment was to see which technique found the most hazards, and the results indicated that misuse cases did better than FMEA, and textual misuse cases did better than diagrams. In [10] we compared textual (mis)use cases with sequence diagrams, indicating that use cases were better for identifying hazards resulting from the operation of the system (e.g., human operator or user errors), while sequence diagrams were better for identifying system-internal hazards. Sequence diagrams, however, is typically a design-level technique, not for the concept or requirements stage. Hence, an even more interesting candidate for comparison with textual use cases might be higher level diagrams showing an overview of the system, rather than internal message passing.

The objective of this paper, therefore, is to present an experimental comparison between textual use cases and system diagrams which show a high level view of the system components. The rest of the paper is structured as follows: Section 2 presents the research questions and hypotheses, section 3 discusses related work, and section 4 explains the experiment design, whereupon section 5 presents the results. Section 6 discusses threats to validity, and finally section 7 concludes the paper.

2 Research Questions

The choice of research questions is important, since they strongly influence on how an experiment is organized, which data can be collected and how they should be analysed. The research questions for this study are as follows:

- RQ1: Are one of the two methods (hazard identification with system diagrams, vs. hazard identification with textual use cases) perceived as easier to use?
- RQ2: Does one of the two methods enable people to identify more hazards?
- RQ3: Does the methods have different effectiveness for different types of hazards, or is one method uniformly better than the other?

RQ1 is answered applying a t-test to the results from a questionnaire with a number of questions about the participants' preference for the techniques used; all scored on a five point Likert scale. In order to answer RQ2 we will compare the number of hazards identified by more than 30% of the experiment participants for each method. The reason for choosing a value of 30% is that this will usually enable a group of four persons to identify more than 80% of the hazards present – see [13]. RQ3 is answered by comparing the portion of experiment participants who identify each hazard.

3 Related Work

The first thing to take note of when doing a literature survey on hazard identification and UML is the paucity of experimental results. Bernardi et al. [14] have published a survey over methods for modelling and analyzing dependability in systems described using one or more UML diagrams. The survey covers 43 papers and identifies a total of 33 approaches. The focus for all these papers is to identify what we must add to UML in order to perform dependability modelling and analysis. Of the 14 papers that we have checked, however, none of them present any experimental evidence and only one of them present a case study in the proper sense of this word [15]. The rest of the 14 papers contain only discussions or proofs of concept arguments.

Searches in IEEE Xplore and Google Scholar confirmed Bernardi's conclusion that there is little empirical work in this area. Even though there is a lot of work published on UML and safety analysis, there are few real experiments. In addition we find that a lot of what is labelled as experiments is really case studies or proof of concept discussions related to simple examples.

An important part of all work in the area of safety analysis focuses on the earliest possible stage of the development process where we can identify the most important hazards and what is the best basis for such an analysis. In addition to on-going research in our own research group at the NTNU, we have also been working in this area in the EU/ARTEMIS project CESAR. While the NTNU work has focused on concepts and UML, the CESAR work has focused on HazId based on requirements. In three separate papers we compared

- Use case diagrams for part of an electronic patient journal system, analysed using misuse cases [12] against Failure Mode Effect Analysis (FMEA) [11] and found that basing the HazId on misuse cases outperformed using general FMEA [8]. The only exception was hazards related to network communication, where FMEA was better than MUC
- Textual use case for part of an electronic patient journal system against use case diagrams for the same system analysed using misuse cases showed that misuse case analysis based on textual use cases outperformed use case diagrams [9]. However, we also observed that the results are sensitive to the degree of hints supplied by the textual use cases.
- Textual use cases for a simplified train control system outperformed UML system sequence diagrams for hazards stemming from operators and systems functionality, while the UML system sequence diagrams outperformed textual use cases for the inner working of the system [10].

A thorough review of the available research on diagrams vs. tables is presented in [16]. Unfortunately, the result is not too helpful, since it concludes that the relative merits of tables and diagrams vary over the experiments and also over what type of persons participating in the experiments. However, one conclusion seems clear – engineers in general worked better with diagrams although they indicated in the questionnaire that they preferred tables.

Guiochet and Vilchis [17] give a good summary of how UML diagrams can be used as a basis for safety analysis. Lu and Halang [18] have performed a case study of component based HazOp and Fault Tree Analysis [19] and found that both techniques could easily be applied to UML system descriptions.

The results from a controlled experiment are presented by Lauritsen and Stålhane [20]. The experiment was performed with participants from three Norwegian IT-companies with four experienced developers from each company. The purpose of the experiment was to compare safety analysis based on FMEA and the ad hoc method that the companies currently used. The conclusion was clear – the developers preferred FMEA to their current ad hoc method.

Another important case study was performed by Martin-Guillerez et al. [21]. The case study is important for our work since it includes a study of HazOp applied to textual use cases. Their main results are that (1) as the safety analyses share the UML diagrams with the developers, flexibility is improved by consistency checks between modelling and HazOp tables but that (2) hazards related to use of machinery are not identified, the analysis is a repetitive task and it is thus difficult to keep the analyst motivated.

Jarzebowicz and Gorski [22] have also run a set of HazOp UML experiments to study the effect of adding scenario based reading of the UML diagrams and the influence of the time allotted to the job. The experiments showed that adding scenarios did not add value to the process in the sense that more hazards were discovered. Two groups of experiment participants were given less time to identify hazards – 2.5 hours vs. 3.0 hours. These groups found as many hazards as the other groups but achieved a higher efficiency score due to the shorter time assigned.

Törner et al. [23] have performed several interesting experiments on methods for hazard identification. His experiments are of special interest since they have been done in an industrial setting with professional software developers. The most important experiment compared Functional failure Analysis (FFA) and a method used by the European Space Agency (ESA) based on a predefined set of generic hazards. The main result of the experiment was that the participants found FFA easier to use than the ESA method. FFA was also found to be more efficient.

4 Experiment Design

4.1 General Considerations

The experiment participants – 48 third year IT students – were sampled through convenience sampling and split into two groups of 24 students each through a random selection process and placed in separate parts of a large auditorium. The number of seats was more than three times the number of participants, and no participant was allowed to sit right next to another participant, to make sure that no participant could easily see what any other participant wrote, thus ensuring independence of responses and preventing diffusion of treatments. Both groups were handed a document containing a pre-experiment questionnaire (5 minutes were spent for answering this), a short intro to hazard identification (15 minutes reading time), a case to be solved (45 minutes) and a post-experiment questionnaire (10 minutes). Group 1 received an intro to hazard identification and a description of a case based on analysis of textual use cases, while group 2 got the same material adapted to system diagrams. The 45 minute task to be solved was of course the same for both treatment groups.

All participants analysed the same simple train control system. The textual use case group (group 1) received two use cases – one for the control operator doing train scheduling and one for the engineer's status reporting. The system diagram group (group 2) just got a set of system diagrams and were instructed to indicate a system component on the diagram and write down possible hazards related to this component on the free part of the page. The man difference was not to indicate hazards on system diagram or on textual use cases but the focus of the documents that the experiment participants used – either a system diagram first and then the use cases for identifying hazards or just the system diagrams.

The pre-experiment questionnaire focused on the participants' previous experience related to UML and analysis of safety and reliability. The post-experiment questionnaire was inspired by the Technology Acceptance Model (TAM) [24], having questions in the categories Perceived Ease of Use (PEOU), Perceived Usefulness (PU), and Intention to Use (ITU). Both questionnaires used five-point Likert scales. The pre-experiment questionnaire showed no difference between the two groups that was significant at the 5% level.

4.2 The Use Case Experiment

The textual use case for the control room operator's train scheduling is shown in table 1. The experiment participants also got a similar use case for the engineer aboard the train. The textual use case has three parts – the main part which shows the main activities for the use case, an alternate part which handles alternative actions and an exception part, used to handle conflicts that arise during train scheduling. In addition to the two columns showing user and system actions, the use case also contains a threats column where the experiment participants should write down hazards identified during the experiment. Being empty here, this column is left quite narrow for space reasons, but was a lot wider in the experiment sheets given to participants, to ensure that they had enough space for inserting threats that they came up with.

In addition to the use case tables, the textual use case participants also got a small version of the system diagram on top of each use case table plus a complete system diagram which they could use for reference during the hazard identification process – see figure 1. Thus, group 1 had more information than group 2 – both textual use cases and system diagram – and we expected group 1 to outperform group 2 on all measures and indicators.

Table 1. Use case for train scheduling

Use case name	(Re-)Schedule train	
Use case actor	Control central operator	
User action	**System action**	**Threats**
1. Request to enter schedule info 3. Enter the schedule (train-ID, start and stop place and time, as well as timing for intermediate points) 5. Confirm schedule	2. Show the scheduling screen 4. Check that the schedule does not conflict with other existing schedules; display entered schedule for confirmation	
Alternative paths		
1.1 The request is to edit an existing schedule 3.1 The operator changes some info in the schedule, then the use case proceeds as normal	2.1 The system shows the schedule	
Exceptions		
5. Operator must decide whether to change the schedule or alternatively to reschedule also the other train / event	4. Schedule conflicts with another scheduled train or maintenance task	

4.3 The System Diagram Experiment

The system diagram shows all the main system components such as train, engineer and control room operator and how these components are connected.

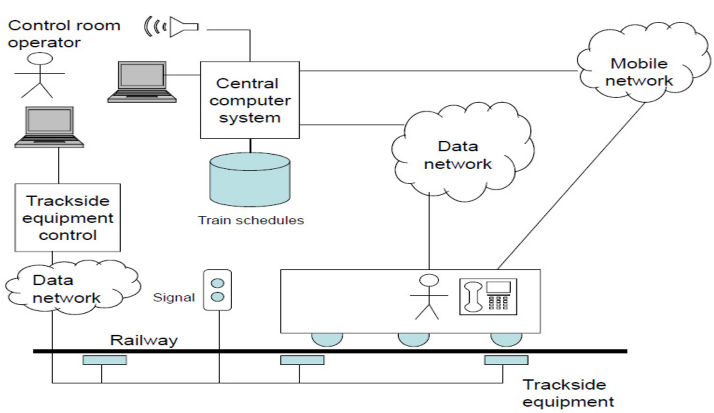


Fig. 1. System diagram

The project participants should mark the part of the diagram they were considering and write down all identified hazards in the free space below the diagram. Each group 2 participant had four identical pages available for writing down identified hazards.

5 Results and Analyses

5.1 Experiment Results Coding

All experiment results were coded independently by two researchers. The coding was done based on a predefined set of possible hazards for this system. The hazards were split into four categories – hazards induced by problems in the control centre, by engineer trouble, by computer system faults and by telecom and trackside equipment faults. In order to be able to compare results across experiments we have used the same set of hazards for all experiments pertaining to the train control case. The results from the coding were entered into a Minitab working sheet for statistical analysis. The working sheet has one row per experiment participant and one column per hazard as defined in the coding sheet.

5.2 Effect Size and Sample Sizes

Effect size and sample-size are important factors when we analyse experiment results. These two factors are linked through the risk acceptance level – our choice of significance level (α-value). In most cases, however, we do not know the size of an effect. In order to help us out, several authors have made tables, mapping a qualitative effect value to a quantitative one. The table used in this paper is taken from W.G Hopkins' work [25]. As for all such tables, they should be used with some care. They are not the ultimate answer to our problem but they are useful as guidelines.

In all analysis of experimental results, we need to consider the relationship between the sample-size (N), our choice of significance level (α) and the size of the variable (p) or difference (Δ) that we observe. In addition to standard statistical notation, we will use the following:

- N: sample size. If we are drawing two samples, as is common in an experiment, N is the size of each sample.
- ES: effect size, defined as Δ/SD –SD being the standard deviation
- p: the probability of finding a certain hazard – equations 2 and 3
- d: the uncertainty of p – equation 3.
- Δ: the observed difference between p_1 and p_2 – equation 2

$$N = 4\frac{(u_{\alpha/2} - u_{\beta})^2}{ES^2} \tag{1}$$

$$N = (u_{\alpha/2} - u_{\beta})^2 \frac{p_1(1-p_1) + p_2(1-p_2)}{\Delta^2} + \frac{2}{\Delta} + 2 \tag{2}$$

$$N = u_{\alpha/2}^2 \frac{p(1-p)}{d^2} \tag{3}$$

The sample size N is in our case is fixed – 48 for eq. 1 and 24 for eq. 2 and 3. Using the equations to decide observable effect size and uncertainties for each set of observations gives ES = 0.81 and d = 0.18 for p = 0.3. For eq. 2 we have computed N for all cases and only accepted observations – p_1 and p_2 – where we find N < 25.

5.3 RQ1 – Method User Friendliness

The post-experiment questionnaire had 13 questions. When using the t-test, only one of the questions shows a difference that is significant at the 5% level, namely "Q5: If I need to identify safety threats in a future project course I would use method T", where method T indicates the method the participant used in the experiment.

The participants that used textual use cases scored an average of 3.6 – agree somewhat – while the participants that used the system diagram scored an average of 4.0 – agree. This gives an effect size of 0.4 which, according to W. Hopkins [25], is a small to moderate effect size. The relationship between effect size and sample size used is given by equation 1.

5.4 RQ2 – Hazard Identification

The participants that used textual use cases had 10 hazards with identification probability larger than 0.30. The participants using the system diagram had 14 hazards with identification probability larger than 0.30 – see discussion on RQ2 in section 2. Based on this observation it is reasonable to conclude that HazId based on diagrams finds more hazards than HazId based on textual use cases. This conclusion is, however, changed when we consider the uncertainties of the identification probabilities.

The relationship between sample size and uncertainty is given by equation 3 where d is the uncertainty in p. With a sample size of 24 and $\alpha = 0.05$, we get d = 0.18. With the computed d value we get a 95% significance if we accept p-values $\geq 0.30 + 0.18 = 0.48$. These are marked by an asterisk in table 3. Using this information we find that HazId based on textual uses case identify eight hazards while the HazId based on system diagrams only identify five hazards.

5.5 RQ3 – Method Efficiency

When we plot the differences between the portions of experiment participants who identified which hazard, we get the diagram shown in figure 2 below. The y-axis shows the *difference* between the percentage of participants using textual use cases (group 1) and the percentage of participants using the system diagram (group 2) who found each hazard indicated on the x-axis.

The relation between sample size and observed portions is given by equation 2. When we use this equation to compute the necessary sample size we find that only the nine items marked with an asterisk in table 3 have enough observations.

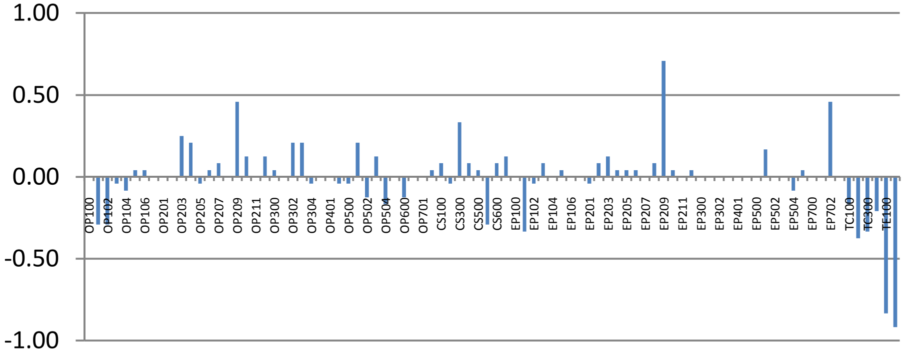


Fig. 2. Text (group 1) – Diagram (group 2) for all considered hazards

All results from the data analysis – RQ2 and RQ3 – are summed up in table 3. We see that HazId based on textual use cases is better than diagram based HazId in seven cases while diagram based HazId is better than textual use cases in nine cases. When we look at the details, as they are documented in table 3, the picture gets a bit more complicated. The most obvious observations are:

- In six out of 16 cases where we have significant differences between system diagrams and use cases, neither of the methods gives an identification probability of 0.3 or more. We should also note that 10 of the 16 differences are small or moderate. Only six of them are large or very large.
- For the control room operator, we find three cases where diagrams give the best results and four where the textual use cases give the best results. Only one difference is large to moderate and this one is in favour of the textual use cases.
- For the train engineer case we find that the diagram gives the best result in one case while the textual use cases give the best result in two cases. The cases where use cases give the best results are large or large to moderate, while the diagram gives a small to moderate result.
- The system diagram is significantly better in all cases when it comes to technical communication problems (codes TC and TE) and four out of five differences are significant at the 5% level. If we use Hopkins' table for effect sizes, only two of the differences are very large, the rest is moderate or small to moderate.
- For the computer system, only one hazard gives a significant difference, and this difference is in favour of the textual use case.

To sum up: there is no uniformly best method. Textual use cases are better than system diagrams for identifying hazards where humans are involved – control operator and engineer and for computer systems – while the system diagram is better for hazards related to telecommunication and track-side equipment.

Table 2. Experiment result overview

Text	T-D sign.	T > 0.29	D > 0.29
OP 101 Receives message too late	-0.29*		
OP 102 Misunderstand message, info or request.	-0.29		0.42
OP 203 Fails to ack., cannot ack.	0.25		
OP 204 Wrong ack.	0.21		
OP 209 Enters wrong info	0.46	0.88*	0.42
OP 302 Wrong train scheduling		0.58*	0.38
OP 303 Fails to schedule	0.21		
OP 501 Wrong situation analysis		0.54*	0.33
OP 504 Lack of training	-0.17		
CS 300 Reacts wrongly to command	0.33	0.46	
CS 400 Shows wrong info		0.67*	0.58*
CS 500 Unavailable, e.g. network problems		0.54*	0.50*
CS 510 Down – e.g. due to crash or power loss			0.58*
CS 700 Other software errors		0.54*	0.42
EP 101 Receives message too late	-0.33*		0.33
EP 102 Misunderstand message, info or request.		0.33	0.38
EP 209 Enters wrong info to operator	0.71*	0.75*	
EP 702 Problems with telephone or radio	0.46*	0.75*	
TC 200 Bad signal coverage or poor radio signal	-0.38*		0.46
TC 300 Busy line	-0.33*		0.33
TC 400 Other technical communication problems	-0.21*		
TE 100 Wrong signal set	-0.83*		0.88*
TE 200 Signal equipment fails	-0.92*		0.92*

When we weed out all differences and portions that have too little support – too small sample size, we get the results shown in table 4. This table strengthens our previous conclusion: textual use cases give HazId results that are as good as or better the system diagrams for all cases except for telecom and track-side equipment.

The differences between textual use cases and system diagrams cannot stem from the presence or absence of system diagrams as the diagram is present in both sets of documentation. For group 1, the system diagram was presented at a separate page before the textual use cases. In addition, the experiment participants were continuously reminded of the diagram, since a smaller version, without any text, was placed on top of each textual use case page. Group 2 had only the system diagram, since the hazards were identified by making notes on and below the diagrams.

As far as system diagrams are concerned, we conclude that presence of information is not enough. It must be presented to the experiment participants – and real analysts – in a way that makes it simple to consider during the whole HazId process. In addition, the textual use case might have drawn the attention away from the system diagram.

Some hazards inside each relevant hazard area have been identified – six out of seven for the control room operator, five out of seven for the computer system and three out of four for the engineer or a total of 78% of all relevant hazards.

Table 3. Experiment result overview when sample size is considered

Text	T-D sign.	T > 0.47	D > 0.47
OP 101 Receives message too late	-0.29*		
OP 209 Enters wrong info	0.46	0.88*	0.42
OP 302 Wrong train scheduling		0.58*	0.38
OP 501 Wrong situation analysis		0.54*	0.33
CS 400 Shows wrong info		0.67*	0.58*
CS 500 Unavailable, e.g. network problems		0.54*	0.50*
CS 510 Down – e.g. due to crash or power loss			0.58*
CS 700 Other software errors		0.54*	0.42
EP 101 Receives message too late	-0.33*		0.33
EP 209 Enters wrong info to operator	0.71*	0.75*	
EP 702 Problems with telephone or radio	0.46*	0.75*	
TC 200 Bad signal coverage or poor radio signal	-0.38*		0.46
TC 300 Busy line	-0.33*		0.33
TC 400 Other technical communication problems	-0.21*		
TE 100 Wrong signal set	-0.83*		0.88*
TE 200 Signal equipment fails	-0.92*		0.92*

6 Threats to Validity

We will base our discussion of threats to validity on the threat categories used by Wohlin et al. [26] but we will only go into details for the threat categories that we consider to be important for our experiment.

Conclusion Validity – threats against being able to draw the correct conclusions about the relationship between treatment and observations:

- Low statistical power. In addition to applying standard statistical tests, we have also included sample size and effect size considerations. This ensures us that we in the conclusions only include results that have a sufficient set of observations to support it. In addition, it should be noted that all the statistical methods used have shown large robustness against the assumption of approximate data normality.
- Some purists may dislike the use of the t-test and other parametric statistics on ordinal data. However, Tukey's seminal paper on this question [27] presents an in-depth argument on why parametric analyses make sense also for ordinal data. The only precondition necessary is that the data distribution is uni-modal and the data collected fulfil this condition.
- Reliability of measure: the measures are generated through a two-step process – coding and counting. The coding is done based in a coding scheme that has been used in several similar experiments and both persons who coded the participants' responses have used this scheme several times earlier. Thus, misunderstandings or misinterpretations are unlikely.

- Reliability of treatment implementation: all participants got an introduction to HazId, what a hazard is and how they should indicate them on the sheets that were distributed before the experiment. We found no cases where the participants had misunderstood the instructions.

Internal Validity – threats due to something else than the treatment influencing the observations:

- Instrumentation – threats due to bad forms, diagrams score sheets etc.: the questionnaires had a straight-forward layout, common to the questionnaires used in most cases where the Likert scale is used. This type of questionnaire is commonly used and all experiment participants should be able to understand it.
- Selection – threats due to the way we selected experiment participants, especially participant heterogeneity: using the pre-experiment questionnaire, we have checked that the two groups in the experiment have the same average scores for important characteristics, such as experience with safety analysis and the use and understanding of UML diagrams. The use of students as experiment participants will only be a problems when we discuss generalization of our results.

Construct Validity – will the experiment reflect the real world phenomena that we want to observe:

- Instructions to the experiment's participants. All the participants got the same introduction to the system to be analysed and to safety analysis. Whether this introduction was good or bad is immaterial since all participants got the same one and differences in the experiment results can thus not stem from the instructions.
- Mono-method. All experiment results depend on one single variable – the number of identified hazards. On the other hand, the assessment of the differences in treatment is also only centred on one variable – the number of hazards. Thus, in our opinion, the mono method bias is not important for our experiment.

External Validity – can we generalize the results of the experiment and if the answer is "Yes", into which areas can we generalize it:

- Coupling between system complexity and HazId method. It is possible that different methods will be best suited for different levels of system complexity. System complexity is, however, not considered in our experiment.
- Interaction of selection and treatment. This an important threat – not to the validity of the experiment but to our ability to generalize the results to an industrial setting. The participants had an average of 2 months of industrial experience but there was no significant difference between the two groups – $p = 0.84$. We should, however, be a little careful about generalizing the results to the IT-industry. On the other hand, related experiments on defect detection done by P. Runeson [28] concluded that there was no significant difference between graduate students and professionals. Since we are only interested in the differences between the two methods, the claim that the differences will be the same for IT-professionals is a weaker claim than that the results – e.g. portion of hazards identified – are the same.

- Interaction of setting and treatment. The experiment was run in an auditorium, while most people working in industry will work in their office. Working in a large auditorium is, however, not all that different form working in an office landscape. A more important problem is such work in industry will often be team work, while the experiment participants pursued the task individually and it is possible that one technique could be better for individual work, while another technique could be better for team work, thus limiting the possibility of generalizing the results to an industrial setting. We have no way to check or control this effect and its influence on the results of the experiments can thus not be decided.

To sum up – the main threat to validity of this experiment is that it was performed with students and not with IT-professionals. On the other hand – we are looking for differences between methods, not their absolute merits. Thus, we will claim that the results are valid, also for IT-professionals. In addition we should remember Wright's conclusion in [29]: "...the difference in performance between experts and lay persons were small in magnitude and the nature of the biases were the same in both groups"

7 Conclusions and Further Work

In previous experiments, we have concluded that although no method is uniformly best, textual uses cases are the best basis for HazId, given that the use case includes the system parts and actions under consideration. Although the effect size is only small to moderate it is still a clear preference for the system diagram as a basis for analysis among the participants. The fact that a method is effective for a type of activity is, however, no guarantee that it is the preferred methods among the users. This effect has also been observed by Coll et al. [30].

If we sum up the experiment results currently available it is reasonable to conclude that textual use cases in general are better as a basis for HazId than use case diagrams, system sequence diagrams and FMEA. The reason for this is mainly that only actions, messages and components that are mentioned in the information supplied to the experiment participants are considered in the HazId. Thus, there are two conclusions:

- Persons who are not professional safety analysts can identify most of the important hazards, given that the item under risk is brought to their attention. The ability to include hazards pertaining to items not explicitly mentioned stems from domain knowledge and HazId experience.
- If users and developers shall perform safety assessment we must include information pertaining to all parts of the system, i.e. GUI, logic and data communication.

In the current experiment it was observed that it is not enough to provide information related to the system's working. It must also be continuously kept in the analysts' focus. This observation strengthens our general observation and experience that human factors and abilities is an important part of the system safety assessment process.

References

1. Croxford, M., Sutton, J.: Breaking through the V and V Bottleneck. In: Toussaint, M.-J. (ed.) Ada-Europe 1995. LNCS, vol. 1031, pp. 344–354. Springer, Heidelberg (1996)
2. Jürjens, J.: Developing Safety-Critical Systems with UML. In: Stevens, P., Whittle, J., Booch, G. (eds.) UML 2003. LNCS, vol. 2863, pp. 360–372. Springer, Heidelberg (2003)
3. Bush, D.: Modelling Support for Early Identification of Safety Requirements: A Preliminary Investigation. In: Firesmith, D. (ed.) 4th Int'l Workshop on Requirements for High Assurance Systems (RHAS 2005), Paris, France (2005)
4. Abdel-Hamid, T.K.: The Economics of Software Quality Assurance: A Simulation Based Case Study. MIS Quarterly (1988)
5. Lutz, R.R.: Analyzing software requirements errors in safety-critical, embedded systems. In: Proceedings of IEEE International Symposium on Requirements Engineering (1993)
6. Allenby, K., Kelly, T.: Deriving Safety Requirements Using Scenarios. In: Fifth IEEE International Symposium on Requirements Engineering (RE 2001). IEEE, Toronto (2001)
7. Berry, D.M.: The Safety Requirements Engineering Dilemma. In: Proceedings of the 9th International Workshop on Software Specification and Design, p. 147. IEEE Computer Society (1998)
8. Stålhane, T., Sindre, G.: A Comparison of Two Approaches to Safety Analysis Based on Use Cases. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 423–437. Springer, Heidelberg (2007)
9. Stålhane, T., Sindre, G.: Safety Hazard Identification by Misuse Cases: Experimental Comparison of Text and Diagrams. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M., et al. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 721–735. Springer, Heidelberg (2008)
10. Stålhane, T., Sindre, G., du Bousquet, L.: Comparing Safety Analysis Based on Sequence Diagrams and Textual Use Cases. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 165–179. Springer, Heidelberg (2010)
11. Stamatis, D.H.: Failure Mode and Effect Analysis: FMEA from theory to execution. American Society for Quality (ASQ), Milwaukee (1995)
12. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering 10(1), 34–44 (2005)
13. Turner, C.W., Lewis, J.R., Nielsen, J.: Determining Usability Test Sample Size. In: Karwowski, W. (ed.) International Encyclopedia of Ergonomics and Human Factors, pp. 3084–3088. CRC Press, Boca Raton (2006)
14. Bernardi, S., Merseguer, J., Petriu, D.C.: Dependability modeling and analysis of software systems specifiec with UML. ACN Computng Surveys (2011)
15. Iwu, S.E.A.: Integrating safety and formal analyses using UML and PFS. Reliability Engineering and Systems Safety 92(2), 49–65 (2007)
16. Allmendinger, L.: Diagrams and Design Tools in Context. ACM SIGDOC Asterisk Journal of Computer Documentation 18(4), 25–41 (1994)
17. Guiochet, J., Vilchis, A.: Safety Analysis of a Medical Robot for Tele-echography. In: Proc. 2nd IARP IEEE/RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments, Toulouse, France, pp. 217–227 (2002)
18. Lu, S., Halang, W.A., Zalewski, J.: Component-based HazOp and Fault Tree Analysis in Developing Embedded Real-Time Systems with UML. In: Proceedings of the 4th WSEAS International Conference on Electronics, Control and Signal Processing, pp. 150–155 (2005)

19. Leveson, N.G.: Safeware: System Safety and Computers. Addison-Wesley, Boston (1995)
20. Lauritsen, T., Stålhane, T.: Safety Methods in Software Process Improvement. In: Richardson, I., Abrahamsson, P., Messnarz, R. (eds.) EuroSPI 2005. LNCS, vol. 3792, pp. 95–105. Springer, Heidelberg (2005)
21. Martin-Guillerez, D., et al.: A UML-based method for risk analysis of human-robot interactions. In: 2nd International Workshop on Software Engineering for Resilient Systems. Birkbeck College, London (2010)
22. Jarzebowicz, A., Gorski, J.: Empirical Evaluation of Reading Techniques for UML Models Inspection. International Transactions on System Science and Applications 1(2), 103–110 (2006)
23. Törner, F., Johannessen, P., Öhman, P.: Assessment of Hazard Identification Methods for the Automotive Domain. In: Górski, J. (ed.) SAFECOMP 2006. LNCS, vol. 4166, pp. 247–260. Springer, Heidelberg (2006)
24. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. Management Science 35(8), 982–1003 (1989)
25. Hopkins, W.G.: A New View of Statistics. Brisbane, University of Queensland (2001)
26. Wohlin, C., et al.: Experimentation in Software Engineering: An Introduction. Kluwer Academic, Norwell (2000)
27. Tukey, J.W.: Data analysis and behavioral science or learning to bear the quantitative's man burden by shunning badmandments. In: Tukey, J.W., Jones, L.W. (eds.) The Collected Works, pp. 187–389. Wadsworth, Monterey (1986)
28. Runeson, P.: Using Students as Experiment Subjects – An Analysis on Graduate and Freshmen Student Data. In: Linkman, S. (ed.) Proc. 7th International Conference on Empirical Assessment & Evaluation in Software Engineering (EASE 2003), pp. 95–102. Keele University, Staffordshire (2003)
29. Wright, G., Bolger, F., Rowe, G.: An Empirical Test of the Relative Validity of Expert and Lay Judgement of Risk. Risk Analysis 22(6), 1107–1122 (2002)
30. Coll, R.A., Coll, J.H., Thakur, G.: Graphs and tables: a four factor experiment. Communications of the ACM 37(4), 77–84 (1994)

Towards Tool Support for Design and Safety Analysis of High Consequence Arming Systems Using Matlab

Dan Slipper[1], Wilson Ifill[2], Gordon Hunter[2], Roger Green[2], Richard Johnson[2], and Alistair A. McEwan[1]

[1] Department of Engineering, University of Leicester
[2] AWE, Aldermaston

Abstract. High consequence arming systems are designed to prevent unwanted external (or potentially internal) energy flowing to a critical component without intention. The hazard analysis of such systems can be a slow and difficult manual process, potentially repeated in various life-cycle phases or on multiple design options. This paper details a simulation tool under development at AWE to provide a fast and repeatable analysis process. The simulation generates a set of possible paths along which different energy types could potentially propagate through the system. Behaviour identified by the tool can support the design of the system and selection of an architecture providing assurance of safety whilst still providing reliability. We present an outline of the model development process, results from its use with a case study and demonstrate the advantages over manual analysis. A number of limitations of the current implementation are discussed, we then propose future work aimed at alleviating some of these issues.

Keywords: safety analysis, matlab, simulation, propagation.

1 Introduction

High consequence arming systems such as nuclear weapons pose a potential hazard throughout their operation. *High consequence systems* can be defined as ‘those where failures can cause catastrophic results’ [1]. This can apply to many industries where the combination of different energy types within a system can be dangerous, for example the chemical process industry. However for our specific industry, (and the nuclear power industry) the catastrophic results would be radiological dispersal or nuclear yield. Within both normal and abnormal environments there are system *hazards* (potential conditions that can cause injury [2]). Where possible these hazards should be removed from the system during the design phase to maintain assured safety. When components essential for operation present a potential hazard, the impact of such a hazard should be reduced through the implementation of safety features. The effects of external *insults* (physically measurable phenomena with the potential to detrimentally

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 393–405, 2012.

affect the system [3]) must also be considered to ensure that the system remains adequately safe through all environments.

Regulations and design principles set out in JSP 538 [4] and JSP 372[5] specify the top level safety requirements which UK Nuclear Weapons must adhere to. Evidence that these stringent requirements are fulfilled must be provided before a system can be commissioned for service. Evidence from sub-system tests, design rationale and modelling is used to demonstrate that the implemented safety features are reliable and adequate.

System level safety is dependent on subsystem or component behaviour. Many of the components can potentially affect energy passing through them and produce a hazard. These hazards can be caused by conversion of given energy into a another type, or a change of magnitude by amplifying or reducing it. Whilst sound engineering and scientific judgment is required to determine the effectiveness of different designs, the variety of threats to components coupled with the complexity of the systems can make it difficult to effectively determine whether all areas of potential concern to the engineer have been investigated.

Software based modelling methods can ease the analysis of these areas by providing the user with a fast, thorough and repeatable process. With use of an object-oriented approach [6] a model can be easily developed and modules re-used through the use of abstract classes. This paper introduces a Matlab [7] simulation tool designed as the initial step towards a methodology to aid design and analysis through modelling. The simulation automates a manual analysis technique used at AWE [3], by identifying the possible paths from an insult through to the critical component of a system. The paths along which energy could potentially travel are then analysed, producing an expected 'safe' or 'unsafe' result for each path, depending on whether appropriate safety features exist. The contributions of the paper are details of a simple, re-usable simulation technique built upon manual methods for analysis and case study evidence demonstrating how this method improves upon the existing process. The technique does not aim to provide proof or evidence that a path is sufficiently safe, however it has been developed to aid the safety analyst in identification of points of concern within a system design.

The contents of the paper are structured as follows; Section 2 describes the requirements of the simulation tool. Section 3 details the model components. Section 4 presents the results of an example application analogous to the high consequence system of interest. Section 5 presents a discussion of the results, highlighting the limitations of the tool and potential future work. Section 6 proposes how this future work will be achieved and development of an overarching methodology. Section 7 provides some conclusions about the tool and methodology.

2 Requirements and Context

The tool is intended to aid the design and analysis of a system to ensure it meets the appropriate safety requirements. In order to do this the safety analysis team identified a number of requirements for a tool, which shall:

– Address abnormal environments with non-design mode connectivity.
– Cater for electrical and non-electrical hazards and transformations.
– Only address loss of assurance of safety and not performance.
– Use a deterministic approach that combines critical safety functions and inherent hazards and vulnerabilities of the system.
– Produce a list of the unsafe pathways through the system (some of which could be potentially missed using the manual method).
– Aid design of the system and selection of appropriate safety features.

The structure of the tool is described in Figure 1 using the Unified Modelling Language [8] (UML). It can be seen from this diagram that the system can be modelled as: a group of abstract components, a defined topology, and a defining scenario.

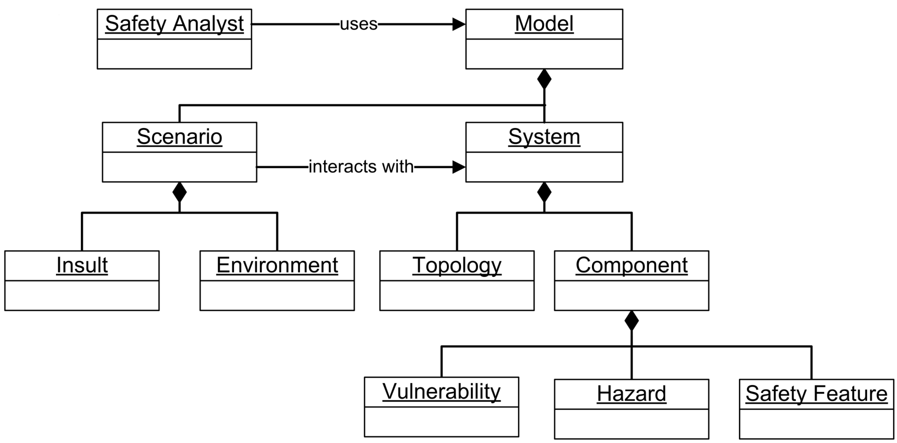


Fig. 1. UML model of the system context

The safety analyst uses a system model to identify potential paths through the system (in a defined scenario). Energy is introduced to the system from an insult (or threat) and then passed between components by affecting vulnerabilities which in turn produce a hazard. Each component will have multiple response functions, each of which represents a vulnerability and could potentially present a hazard to other system components. The components within the system have a topology (or layout) which could be defined by the scenario under test. The requirements of this tool are to analyse environments in which all components could potentially interact with each other. This interconnectivity is only limited by the use of safety features, which can stop energy flowing within known limits. The safety features use the concepts of Isolation, Incompatibility or Inoperability as described further in [9], [10], [11], [12] and [13].

3 Modelling and Execution Process

The context shown in Figure 1 captures both the Matlab elements used to create a system model and its environment. The main elements of interest are the component models, insults and the system topology. When the model is executed it uses a path generator and insult propagator to analyse paths which are potentially unsafe. Each of these stages is described in more detail in the following sections.

3.1 Component Models

Each component (or sub-system) in the model will either be a required to provide a necessary system function, or to provide safety. For either scenario energy may be generated by, contained within, or passed through a component. Therefore each component is modelled with an input and output port (P1 and P2 respectively), through which it can transfer or receive energy between any connecting components via an insult vector (see Section 3.2). A component can have multiple responses, depending on the type, magnitude and direction of energy present at either the input or output port (depending if it is forward flowing or backward flowing). Figure 2 depicts a component with two potential responses to an electrical insult.

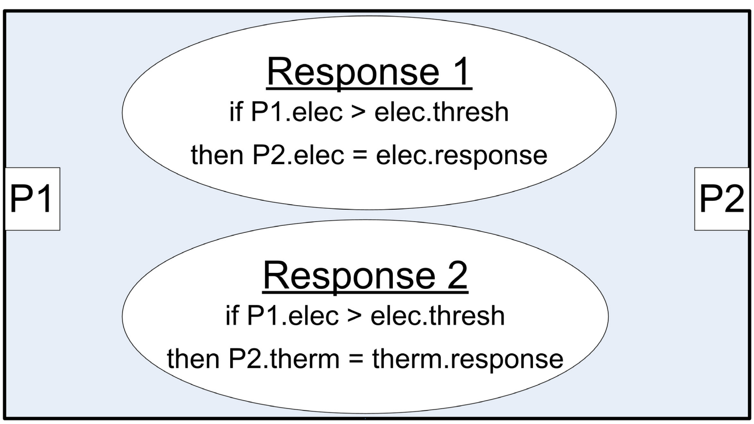


Fig. 2. An example Component with input ports and a number of internal response functions

Although Matlab provides the ability to develop transfer functions for component responses, safety analysis is only concerned with the threshold at which a component can output energy, or in the case of a safety feature, the point that safety can not be assured. Figure 3 demonstrates how the behaviour can be transformed into a function where the component assures safety up to a threshold of X. This is modelled by the broken line, i.e. any insult energy levels above X are seen as unsafe.

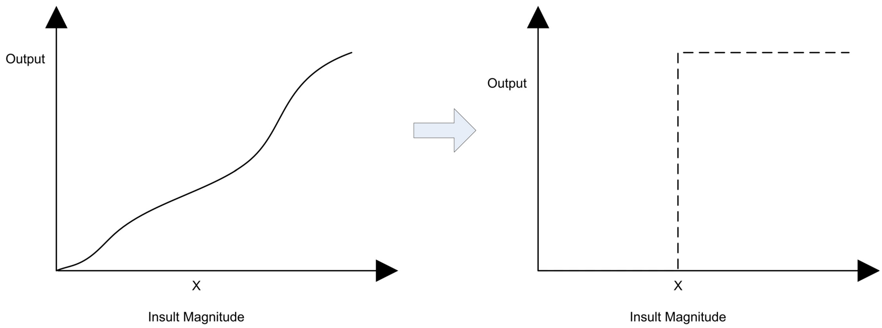


Fig. 3. Representation of a components safety assured response

The software for these models is developed upon a rule-based approach [14], where conditional statements represent the safety assured response in Figure 3. A benefit of using an object-oriented design is that components have no hard coded values. Each instance of the component is created with its thresholds and output values as constructor arguments. For example component C1:

```
C1 = Component_A(240, 240, 100)
```

would create an instance of Component A with an electrical threshold of 240V, electrical response of 240V and a thermal response of 100 Degrees. Assigning values from the main program code allows repeatability by changing the values in a single place. In early system design stages (e.g. generating and testing architecture options) the information about component response functions would be difficult to provide accurately. The values assumed would be the estimated worst case for assured safety and based on expert judgment. The output from the model can aid the selection of appropriate components if margins of unsafe behaviour were considered. By using this iterative development process the model specification can be refined as further detailed design information becomes available, or by using results of trials on individual components. The tool allows us to model the system level response that emerges from a network of interrelated components.

3.2 Insults

To run a simulation an insult source should be provided, representing either external energy introduced to the system or internal energy that could propagate (i.e. its safety is not assured). The model is representative of the scenario of interest to the safety analyst and could involve multiple types of energy being provided at a given point. In the Matlab model an insult is captured as a vector of different insult objects, each object being an instantiation of an insult type (e.g. Electrical object, Thermal object etc). This enables us to model the propagation of multiple energy types between components simultaneously. This vector is passed between components through p1 and p2 along the paths that are generated by the system (see Sections 3.4 and 3.5). The values in the insult vector are modified by the component response functions as the insult passes through each component, until the the critical component is reached. The component then generates a 'safe' or 'unsafe' response. In the current implementation only a single critical component is modelled.

3.3 System Topology

To understand how system components interact the relationships between them should be modelled. This is to identify the paths along which the insult vector can be passed between components. It is assumed that all components could potentially be connected to all others unless a safety feature removes this connectivity (such as an area isolated by a barrier). The system is modelled like a graph, with components represented as nodes and potential connections between them as vertices. These connections can be represented by an adjacency matrix [15].

Capturing all of the interactions in this way aids the modelling of the system in Matlab. Each row of the matrix would be translated into an appropriate line of code detailing where each component could connect to. For example:

```
I_p2 = {'A_p1'};
A_p1 = {'A_p2'};
A_p2 = {'B_p1'};
B_p1 = {'B_p2'};
B_p2 = {'C_p1', 'D_p1'};
C_p1 = {'C_p2'};
C_p2 = {'D_p1'};
```

In the listing above I is the insulting component, and D is the critical component. Letters represent component names, p1 and p2 are pathways in and out of the component, and the connected ports between components are listed within the brackets. A connection from X_p1 to X_p2 shows propagation of energy through a component.

3.4 Path Generation

To generate a list of all of the paths from the insult source through to the critical component, a breadth first search [16] of the network is performed. This algorithm identifies each connection from its current component then in turn repeats the process on each branching component. This generates a tree across the breadth of the graph rather than the depth. When a full chain is found that does not end with the critical component, it is removed from the list. The resulting paths are written to a file and provide the first part of the output. An example of the component search order is shown in Figure 4. Paths generated also show which ports the path passes through (e.g. component A would receive energy

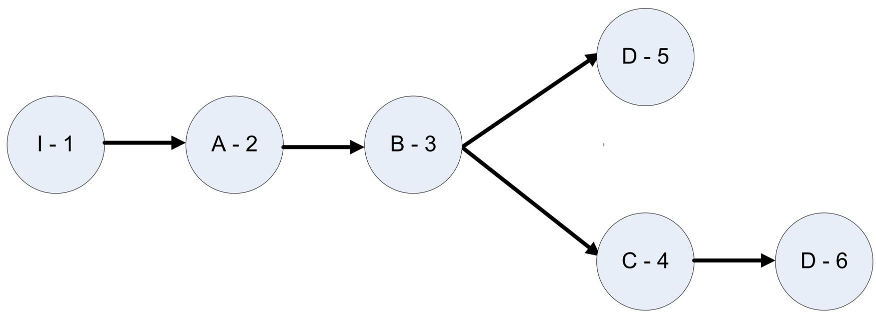


Fig. 4. Selection order from a breadth first search algorithm

in through A_p1 and transmit it out of A_p2). The paths from the adjacency matrix connections previously shown would be:

```
I_p2    A_p1    A_p2    B_p1    B_p2    D_p1
I_p2    A_p1    A_p2    B_p1    B_p2    C_p1    C_p2    D_p1
```

3.5 Insult Propagation

To propagate the insult the simulation iterates through each of the paths generated, taking an insult magnitude that is input to the system and then calculating the change to this vector based on the component responses. The components in the system are instantiated by calling their constructor with the appropriate values (setting thresholds and outputs based on the components responses). This could look like:

```
i1  = I(50);           %electrical insult of 50V
a1  = A(40, 50, 80);  %threshold of 40V, responses 50V, and 80 deg
b1  = B(.., .., ..); %Appropriate types or values are given ..
```

The final component determines if it is unsafe based on the magnitudes held within the insult vector (if applicable). The resulting output is a list of statements of whether the paths are safe or unsafe, linking to the order in which they were generated.

4 Example Application

To demonstrate the use of this simulation tool and highlight the strengths in its usage, a simple case study has been devised. This case study is analogous to that of a high consequence arming system, where there is a potential hazard that should be protected from external energy sources. The selected case study is the design of a car. Modern cars have a number of safety devices both to protect the passengers and to reduce the effects of a potential impact (many are summarised in [17]). The area of interest for the safety in this case study however, is that the fuel provides a constant hazard of fire or explosion. Although vehicles are now designed with safety of fuel tanks in mind and appropriate measures are considered, older vehicle designs have demonstrated these risks. This risk needs to be balanced along with other safety, performance and reliability requirements of the system.

An example is the Ford Pinto. This vehicle was designed with the fuel tank rear of the axle, according to [18] this was due to the limitation on boot space (a balance between performance and safety), this also limited the amount of crush space around the component. This design decision resulted in the fuel tank exploding upon impact (above a certain magnitude) to the rear of the vehicle. A number of components have been identified within the vehicle system, these could each have a number of responses to insulting energy and in some way affect the fuel tank. The connectivity of some components of interest are shown in Figure 5.

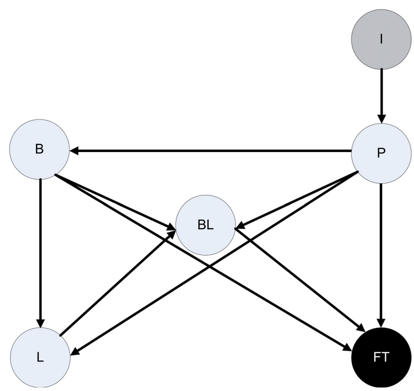


Fig. 5. Potential non-design mode connectivity of the system of interest

For this example only 5 components from the system will be discussed, these are the: Fuel Tank (FT), Lighter (L), Brake Pedal (P), Brake Light (BL) and Battery (B). There is also an Insult (I) shown, of type Shock. In reality these components are unlikely to be collocated in the same physical location within the car, however we must demonstrate that for all design mode and non design mode paths compatible energy cannot reach the fuel tank. The scenario described within the following section is a shock to the brake pedal mechanism, which in a drive by wire system would produce an electrical signal elsewhere.

4.1 Manual Analysis

The case study was analysed manually to identify the expected results and for verification of the model. Using the 5 components previously described, all potential paths through the system were generated by hand and compared against the output from the simulation to validate its results. Manual analysis of the system took just under an hour, with the author having prior knowledge of the case study. The analysis highlighted 16 paths through the system, 2 of which were potentially unsafe for the scenario of interest. Notes under the arrows demonstrate the type and magnitude of energy transfer of a given type (e.g. el = electrical, t = thermal, s = shock). Some paths are shown as an example:

$$I \underset{s=1}{\rightarrow} P \underset{el=12}{\rightarrow} L \underset{t=100,el=12}{\rightarrow} FT = UNSAFE$$

$$I \underset{s=1}{\rightarrow} P \underset{el=12}{\rightarrow} L \underset{t=100,el=12}{\rightarrow} B \underset{t=100}{\rightarrow} BL \underset{0}{\rightarrow} FT = SAFE$$

It was noted during manual path generation that possible paths could easily be missed, even when generating possibilities for a small system of 5 components. This becomes far more difficult with a larger set of components and also with multiple energy types.

4.2 Simulation Results and Analysis

The outcome of the simulation was that the 16 expected paths were generated and as with the manual analysis two of these were potentially unsafe. An example path shown previously was a shock insult which passes through the Brake Pedal generating electrical energy, propagating to the Lighter, which converts the electrical energy to thermal energy, which the Fuel Tank is vulnerable to. The simulation itself only takes a matter of seconds to execute, providing a fast method to get repeatable results. To test a range of scenarios we use the constructor functions to modify the insult magnitude, or component thresholds and responses. Changes to the model structure are also simple and can be iterated based on results of an initial analysis.

Execution of the model with a scenario where the Brake Pedal is shocked has demonstrated that unsafe paths exist within the design. These unsafe paths are scenarios where thermal energy can propagate from the Lighter to the Fuel tank. For this particular scenario it is possible to argue that the path identified has a very low likelihood of occurrence due to the distance between components. To ensure safety however, the incorporation of a barrier with an area for electrical cables and the fuel line to pass through would remove all potentially unsafe paths *for this scenario*. Figure 6 shows the updates to the system, where passing the cable/fuel lines through a barrier limits some of the energy types that can be transmitted to the fuel tank.

Other scenarios also need testing to assess the overall system safety, as extra safety devices may need to be incorporated into the system to assure safety.

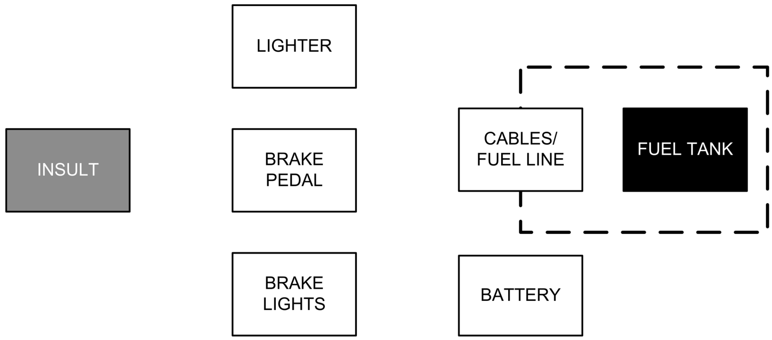


Fig. 6. Updated system with a new safety feature included

5 Discussion

The overall outcome from using the modelling technique is that the generation and analysis of paths with use of a software based tool is faster and less error prone than the current manual process. Removing the aspect of human error from the analysis process is an important motivation for this development. Use of the simulation tool allows repeatability of experiments with very minor changes to the code, which would require a full repetition of the analysis process if undertaken manually. Despite these advantages, the system is only in the early stages of becoming a useful tool to aid analysis and support design decisions. Ideally a number of system design options would be compared against each other to demonstrate which provides the highest assurance of safety. There are also many limitations on modelling the real world with the concepts captured through this technique, as will be discussed.

Component Responses. The first limitation of the system is the component response function. These are programmed as a safety assured response and it is thought that there is little added value at this stage to have a detailed system response for a full range of possible inputs. In the current implementation an insult vector is passed from component to component along a single path. In reality it is possible that an insults could be passed along several of these paths simultaneously.

Component Composition. Within this implementation components are individual modules which cannot be combined to form a larger section of the system without programming their combined behaviour manually. It would be desirable to analyse the system at different levels of abstraction, allowing low level component models to be composed together to represent a subsystem model.

Distributed Insults. Another limitation of the system in its current form is the way insults to the system are input into a single component. In some scenarios, the environments which the system may experience could involve multiple insults to different places. For example dropping the system could crush some components upon impact and provide a mechanical shock to others. The result of these parallel events would then propagate between the components.

Capturing Design Information. The way in which component specifications are captured is a possible expansion of the technique. When developing a model for a complex system, tools to capture the appropriate information are desirable (for example the adjacency matrix method). When capturing component responses, a structured representation of the system responses would be useful, for this the Unified Modelling Language (UML) is proposed, and discussed further in Section 6.

Design Decisions. Once components required for the systems functionality have been identified, there could be a range of potential arguments or safety devices that could be used to assure system safety. Some of these options may not be the most efficient or cost effective, and part of the design process would be to select appropriate features to achieve the required level of safety as economically as possible. Ideally the model would aid analysis of this.

6 Future Work

Future work is proposed to fulfill the limitations described in the previous section. Ideally a full methodology to aid design and analysis is desired and this should be supported by an appropriate simulation tool. This tool should aid the team selection of an appropriate safety architecture, and possibly extend to consider the reliability of a design. The use of simulation would provide the team with a robust, repeatable method of analysing the system and supporting the stringent safety requirements placed upon them. This methodology would consist of stages to capture the system information appropriately, with the use of multiple UML views. These models could then be (manually) translated into appropriate rules about the system in a language that supports simulation, concurrency and model checking. Tools of interest are Coloured Petri Nets [19] or Communicating Sequential Processes [20]. Both are being investigated for their potential use. A process for development of such a model will be considered in order to avoid model checking state space explosion problems. Possible ways to remove the problem could be to develop the model subsystem at a time or removing know impossibilities before model development (such as removing energy types that are not compatible). All of this information will be captured in an appropriate format to present the safety analyst with a set of arguments about the system which assure safety.

7 Related Work

Related work has been published from Sandia National Laboratories where the combination of Fault Tree Analysis, Event Trees and Finite Element Models have been described in [21]. Failure modelling techniques exist with a similar concept of component specifications and responses when analysing software systems. The Failure Propagation and Transformation Notation (FPTN), developed by Fenelon and McDermid [22], provides a notation for capturing component responses and analysing ways through which a component failure can propagate between components. This has been developed further into a calculus by Wallace [23], where different types of transfer of energy can occur (source, sinks, transformation or propagation). Our tool adds the path generation aspect to these existing methods, but utilises the existing concepts for propagation of insults. The methodology under development is expected to provide much more functionality for analysis of the system and to aid the whole design process.

8 Conclusions

The issues of speed, completeness and reliability of safety analysis with manual processes can be improved through the use of software tools. Case study evidence has demonstrated that it is possible to analyse all paths through a system and identify which are of concern in a short time scale using the Matlab tool. The tool was designed to fulfil a number of requirements as stated in Section 2, the model is seen to have fulfilled these requirements, however some of the limitations we have described pose possible extensions beyond this initial requirement set. Future work aims to provide a full methodology to support design and analysis (considering reliability alongside safety), whilst tackling some of the highlighted issues with the current tool.

References

1. Davis, J.: Integrated Safety, Reliability, and Diagnostics of High Assurance, High Consequence Systems, Electrical Engineering (May 2000)
2. Department of Defense. Standard Practice for System Safety MIL-STD-882D (1993)
3. Johnson, C.R.: Methodology for Designing and Analyzing High Consequence Arming Systems. In: Livingston, J.M., Barnes, R., Swallom, D., Pottraz, W. (eds.) Proceedings of the US Joint Weapons Systems Safety Conference 2009, Huntsville, Alabama, pp. 552–561 (2009) ISBN 9781617387142
4. Ministry of Defence. JSP538 - Regulation of the Nuclear Weapon Programme (2008)
5. Ministry of Defence. JSP372 - Approving Authority Management Arrangements for the Trident Re-entry System (2011)
6. Budd, T.: An Introduction to Object-Oriented Programming. Addison-Wesley (2002)
7. Mathworks. MATLAB - The Language of Technical Computing (2011), `www.mathworks.com/products/matlab/` (accessed December 15, 2011)
8. Object Modeling Group. OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2. Technical report (November 2007)
9. Spray, S.D.: Deriving and applying generally applicable safety principles. In: International System Safety Conference, Seattle, WA (September 1998)

10. Cooper, J.A., Covan, J.M.: Predictable Safety in the Control of High Consequence Systems. In: 3rd IEEE High-Assurance Systems Engineering Symposium, Washington, DC (November 1998)
11. Plummer, D.W., Greenwood, W.H.: The History of Nuclear Weapon Safety Devices. In: Conference: 34. AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Cleveland, OH (July 1998)
12. Elliott, G.: US Nuclear Weapon Safety and Control. In: MIT Program in Science, Technology, and Society (2005)
13. Ekman, M.E., Werner, P.W., Covan, J.M., D'Antonio, P.E., Perry, E.: A Thematic Approach to System Safety. In: Process Safety Progress 17:3, American Institute of Chemical Engineers (1998)
14. Zadeh, L.A.: Commonsense Knowledge Representation Based on Fuzzy Logic. Computer 16(10), 61–65 (1983)
15. Kreyszig, E.: Advanced Engineering Mathematics. John Wiley & Sons (2010)
16. Jungnickel, D.: Graphs, Networks, and Algorithms. Algorithms and computation in mathematics. Springer (2008)
17. Fournier, E., Bayne, T., Kot, J.: Review of the State-of-the-Art in Fuel Tank Systems - Phase II. Technical report (May 2003)
18. Schwartz, G.T.: The Myth of the Ford Pinto Case. Rutgers Law Review
19. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall PTR, Upper Saddle River (1981)
20. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)
21. Carlson, D.D., Jones, T.R.: Model-Based Safety Assessments. In: Conference: Lockheed Martin Systems Engineering and Software Symposium, New Orleans, LA (May 1998)
22. Fenelon, P., McDermid, J.A.: An Integrated Tool Set for Software Safety Analysis. J. Syst. Softw. 21, 279–290 (1993)
23. Wallace, M.: Modular Architectural Representation and Analysis of Fault Propagation and Transformation. In: Proc. FESCA 2005. ENTCS, vol. 141(3), pp. 53–71. Elsevier (2005)

Towards the Reconstruction and Evaluation of Conceptual Model Quality Discourses – Methodical Framework and Application in the Context of Model Understandability

Peter Fettke, Constantin Houy, Armella-Lucia Vella, and Peter Loos

Institute for Information Systems (IWi)
at the German Research Center for Artificial Intelligence (DFKI) and
Saarland University, Campus, Building $D3_2$
66123 Saarbrücken, Germany
{Peter.Fettke,Constantin.Houy,
Armella-Lucia.Vella,Peter.Loos}@iwi.dfki.de

Abstract. Within the information systems (IS) discipline conceptual models have gained tremendous importance in the past years. Different approaches for systematic model quality evaluation have emerged. However, these approaches are based on different understandings, definitions as well as operationalizations of the term "model quality". In this article we refrain from conceptualizing and operationalizing model quality a priori. In contrast, assuming that the determination of model quality and appropriate criteria are negotiated in a discourse between modelers and model users based on their different perspectives, we develop a methodical framework for the critical reconstruction and evaluation of conceptual model quality discourses in order to identify relevant model quality criteria and understandings. Our method is exemplarily applied for the reconstruction of the discourse on the quality criterion model understandability based on relevant laboratory experiments. This application shows that many research results on model understandability are hardly comparable due to their different basic assumptions and should preferably be interpreted based on a methodical reconstruction of underlying understandings.

Keywords: conceptual models, model quality, discourse orientation, discourse reconstruction, model understandability.

1 Introduction

As well in theory as in practice, conceptual modeling is considered a promising tool for designing enterprises [1]. Conceptual models are used as methodical instruments for information systems (IS) engineering (*primary design*) as well as in all following phases of the IS life cycle (*secondary design*) [2], and offer application potential in fields such as business process engineering, software engineering as well as for the choice, implementation and customization of standard software. At the same time it

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 406–421, 2012.

can be stated that conceptual models can only fulfill their important function if they are of an adequate quality. Therefore, questions concerning model quality are of high significance for IS engineering.

However, existing contributions indicate that model quality is in general interpreted, conceptualized and operationalized differently [3]. In previous discussions, the importance of model quality for IS research, software engineering, and further scientific disciplines has been identified and it has been acknowledged that model quality might be evaluated differently according to the views and perspectives of different modelers and model users. All of these communities have at least one or usually several different understandings of model quality. In fact, the term *quality* is often used as an umbrella term which subsumes a plethora of different understandings also depending on the intended context of a model's usage. Against this background, the term model quality is not only ambiguous, but also its specific characteristics and facets remain vague. Thus, using general and *a priori* defined model quality criteria for model quality evaluation without considering the specific situation and the different relevant views and perspectives seems to be highly problematic.

This article aims at overcoming these limitations by means of an approach for the reconstruction and evaluation of conceptual model quality discourses in theory and practice. Thereby, model quality criteria are not *a priori* conceptualized and operationalized. In contrast to that, it is assumed that the understanding and determination of model quality and appropriate quality criteria are negotiated in a discourse between modelers, model users and evaluators. In such a model quality discourse, different views and perspectives regarding the understanding as well as the methods for measurement and improvement of model quality become obvious. Investigating and documenting these views and perspectives is important to gain a comprehensive understanding in order to be able to choose adequate quality criteria for the evaluation of conceptual models. Therefore, in our article, a methodical framework for a critical reconstruction and evaluation of conceptual model quality discourses will be developed and exemplarily applied.

In our contribution a *design-oriented research approach* [4] is taken as a basis. Firstly, the current state of the art regarding conceptual model quality is investigated and a terminological analysis of relevant concepts is performed. Furthermore, a methodical framework for the reconstruction and evaluation for conceptual model quality discourses is introduced. In order to illustrate the relevance and feasibility of our approach, the framework is then exemplarily applied and the results are critically discussed.

The article at hand has the following *structure*: after this introduction, the state of the art concerning conceptual model quality is analyzed and discussed in section two. The following section introduces the methodical framework for the reconstruction and evaluation of conceptual model quality discourses which consists of basic concepts and a procedure model [5]. In section four the introduced framework is exemplarily applied for reconstructing the quality discourse on conceptual model understandability as one of the most important model quality dimensions. The reconstructed discourse is methodically based on relevant laboratory experiments. Results are discussed and implications for conceptual modeling research as well as for modeling in practice are presented. Section five summarizes the findings of our contribution and provides an outlook on future research.

2 Model Quality – State of the Art

In literature, the term *model quality* has not been defined consistently. Several types of approaches for understanding and defining model quality can be differentiated: a transcendent approach, a product-oriented approach, an application-oriented approach, a creation-oriented as well as a value-oriented approach [6]. Depending on each approach, different quality aspects and dimensions are relevant for defining and understanding model quality. Application-oriented approaches, e.g., center usability aspects for model users, whereas the creation-oriented approaches focus on characteristics of conceptual model construction processes. Furthermore, a broad spectrum of different approaches concerning the conceptualization of model quality have emerged in literature [3]:

1. *View-specific approaches*: A whole host of works discuss single or one-dimensional requirements regarding conceptual models. These works in most cases refer to specific conceptual modeling methods, which justifies describing and classifying their approaches as *view-specific*. Furthermore, dedicated quality criteria for data models on the one hand [7] and process models on the other hand [8] can be distinguished. In this context, there is a significantly higher number of contributions concerning data models.

2. *View-combining approaches*: Several reference frameworks integrate different model quality aspects which are particularly independent of certain modeling methods and approaches. There exists a series of view-combining approaches [9]. Prominent examples are the *Guidelines of Business Process Modeling* (GOM) [10] or the *SEQUAL framework* [8]. Special variants of the GOM are discussed, e.g., by Janiesch et al. [11].

3. *Constructive suggestions for conceptual modeling*: Selected works offer constructive suggestions on how model quality can be increased. Often, in these works descriptions for the determination of model quality are only addressed indirectly. Linguistic approaches can be found in Ortner et al. [12].

This variety of different approaches and their understanding of quality show that the term model quality is quite differently defined, conceptualized and operationalized. Thus, it remains vague which dimensions are comprised by the term model quality. To what extent existing conceptualizations as well as operationalizations of model quality correspond to common validity requirements, remains unacknowledged, so far. First contributions which conduct explicit validations of quality measurement instruments are Moody [13], Moody et al. [14] and Sedera et al. [15]. In summary, suggestions for the determination of model quality exist. However, in many cases these instruments do not sufficiently correspond to the requirements concerning an exact, detailed and comprehensive conceptualization and operationalization of every possible facet of model quality. Hence, there is a significant need to capture the multitude of relevant quality criteria, to systemize them and to establish a capable terminology that suffices methodical standards. Our methodical framework, which will be introduced in the following, is supposed to contribute to these requirements.

3 Methodical Framework for the Reconstruction and Evaluation of Conceptual Model Quality Discourses

3.1 Basic Concepts

As shown in the related work part, there are different demands concerning the conceptualization and operationalization of model quality. It is understood that some model quality criteria can be determined less problematic than others (*objective criteria*); e.g. syntactic criteria, which are defined in order to measure the correctness of machine execution semantics. A consensus on this may probably be found more quickly than for pragmatic quality criteria such as *understandability*, which depend on *subjective* human perceptions. Besides different criteria systems, which are used to evaluate quality, there exist fundamental differences regarding epistemological assumptions, which have a crucial influence on the quality understanding of conceptual models [16]: If, e.g., the belief is accepted that a model represents or should represent, *the* reality without distortion, then criteria of representational similarity can be used as quality standard. If, in contrast, the idea of an objectively identifiable reality is rejected, then quality standards referring to representational similarity seem to be pointless. Furthermore, there are diverging opinions concerning the question whether empirical experience should be accepted as a potential source of knowledge.

It is, moreover, an important question if dependencies between different quality criteria should be considered to be true by definition (*analytical*) or based on matters of fact (*synthetical*). If, e.g., the quality goal *model understandability on the part of business personnel* is accepted as an important model purpose and thus considered a basic quality criterion, then an increased understandability of conceptual models on the part of business personnel leads to an increased model quality by definition. While the acceptance of the above mentioned quality criterion seems promising in the context of general organizational design, this relation is debatable in other contexts: why should the said criterion be a priori accepted if conceptual models are not used by the business personnel, but by different user groups such as software engineers?

Accordingly, the term model quality is embedded in a plethora of different relationships. The examination of these relationships is an interesting challenge for research. Thus, approaches for discourse-oriented evaluation offer interesting opportunities to face these challenges.

The term *discourse* generally describes a spoken or written discussion subsuming different perspectives on and beliefs about a subject matter. Often, they also show characteristics of an argument. Besides this usage, the term discourse has specific meanings within philosophy, linguistics and other scientific disciplines [17-19]. Within the IS community and in the conceptual modeling context the term discourse is used with specific denotations. For instance, Halpin [20, p. 26] and Sindre et al. [21, p. 252] use the term "universe of discourse" for the denotation of a modeled part of reality which is up for discussion. Although the term discourse is not seldom used within modeling literature, no explicit discussions on the meaning and use of the term exist. In fact, the term is often used as a non-explicated fundamental term. There exist particular works within the IS community that take on the ideas of discourse analysis [22-26]. However, they hardly refer to the evaluation of conceptual model quality discourses.

A critical reconstruction and evaluation of model quality discourses, which can be understood as combinations of speech and modeling activities according to the language/action perspective [27, 28], does not only offer significant advantages. Such an approach is, moreover, necessary to satisfy the concept of quality which is always connected to the perspectives of different roles (e. g., modelers or model users) and situations. Therefore, the paper at hand refrains from conceptualizing model quality *a priori* and formulates the following central assumption:

The term "model quality" can be interpreted, conceptualized and operationalized very differently. Its interpretation, conceptualization and operationalization should be understood depending on a modeling discourse. A modeling discourse can be described as a combination of different speech and modeling activities in the context of the construction and application of conceptual models. Model quality indicates to what extent the model fulfills criteria whose definition, type, extent, identification, value specification, weighting and aggregation are negotiated in a discourse between modelers, model users and evaluators. A quality discourse is the particular part of the modeling discourse that broaches the issue of conceptual model quality.

As mentioned above, we have identified several description parameters for model quality criteria which should be considered for the negotiation of criteria in a quality discourse. These are defined more in detail in the following: First of all, it is important that criteria for model quality should be explained and defined in detail (*definition*). Concerning the measurement of quality, different *types* of criteria can be distinguished e. g., metrical or non-metrical quality criteria. The variable *extent* specifies the amount of relevant model quality characteristics in a discourse. *Identification* summarizes techniques applicable for the determination of a quality criterion. Furthermore, possible measurement values of model quality in the context of empirical investigations should also be established (*value specification*). In order to be able to compare different quality criteria their *weighting* plays an important role. In addition, it seems necessary to determine how several criteria can be merged (*aggregation*).

3.2 Procedure Model for the Reconstruction and Evaluation of Conceptual Model Quality Discourses

Based on our introduced central assumption concerning model quality, the concept of discourse can be further concretized in the context of model quality evaluation. Conceptual models, as a tool for enterprise design in practice or as theoretical objects of research, are not simply "given". In contrast, the rapprochement to a conceptual model is connected with presuppositions, which can be interpreted differently depending on the context of analysis or the language and terminology used. A methodical reconstruction of the terminology used in a discourse should not cover the whole use of language but a relevant part of it. For closed communication acts, a cohesive explanation of the usage of relevant terms should be demanded. As an example, scientific theories or discourses which aim at founding or justifying statements or norms can be seen as such closed communication acts. The explanation of the usage of relevant terms is important to prevent communication from failure and misunderstandings based on the use of language. The following procedure model proposes recommendations for action how to reconstruct conceptual model quality discourses in order to identify and understand relevant quality criteria expressed in a discourse which, furthermore, allow for an adequate evaluation of conceptual models. The procedure model can be used to analyze and assess quality discourses concerning conceptual models in the scientific context as well as in practice. It should be noted that depending on the selected quality discourse topic, the procedure steps may vary in detail concerning the elaboration and processing of elaborated results. However, the procedure model itself provides a structured approach for the reconstruction and evaluation of model quality discourses. Figure 1 gives an overview of the individual steps which are explained in more detail in the following paragraphs.

Procedure model for the reconstruction and evaluation of conceptual model quality discourses
Step 1: Identification and selection of the model quality discourse
Step 2: (Re)construction of the model quality discourse a. Model quality definition b. Model quality conceptualization c. Model quality operationalization/measurement
Step 3: Validation of (re)construction
Step 4: Analysis and evaluation of the model quality discourse
Step 5: Overall assessment

Fig. 1. Steps for the reconstruction and evaluation of conceptual model quality discourses

Concerning the (1) *identification and selection of the model quality discourse*, characteristic criteria and key concepts of a quality discourse topic need to be known based on which a discourse can be identified and delineated. To identify a discourse, different methods can be applied depending on the evaluation context. In the course of a scientific application of our procedure model, all relevant written contributions, e.g. articles, reviewers' comments (if available), response articles etc. could be analyzed exhaustively (systematic review) based on which the underlying model quality understanding becomes apparent. Furthermore, relevant discourse participants can be identified. Using the procedure model in practice, for instance, interviews with model users can help to gain a better understanding of practical modeling discourses. A combination of multiple methods is also possible.

In the next step (2) a *(re)construction of the model quality discourse* is conducted. For a (re)construction of the quality discourse different aspects are relevant. First of all, it should be examined which different quality understandings can be identified in the discourse. In the course of this investigation, addressed quality dimensions should be explored and their conceptualization and operationalization should be captured. In order to support further interpretation it seems reasonable to document additional aspects of a modeling discourse, e.g., discourse participants, contributions, perspectives, background knowledge, design recommendations etc. The mentioned aspects should be structured and concretized. A concrete reference framework classifying the investigated quality understandings should be the goal of this step.

The quality of the discourse (re)construction has to be evaluated based on adequate criteria. These criteria can be identified by compiling relevant evaluation standards which are commonly accepted by a scientific community during the review of each investigated source. This is done in the phase (3) *validation of (re)construction*. Following the idea of established approaches of data acquisition, the same discourse perspective should be reconstructed by several researchers and practitioners so that inconsistencies and ambiguities which can occur during this process can be eliminated.

Different methods can be applied for the (4) *analysis and evaluation of the model quality discourse*. Against the background of using discourse-oriented concepts and methods for the determination of the conceptual modeling discourse quality, it is only consequent to understand the discussion about the quality of the modeling discourse as a discourse itself (*modeling meta discourse*). The aim of this meta discourse is to identify, conceptualize and operationalize criteria for the understanding of the quality of the modeling discourse. At this point it is especially interesting to examine if the discourse is held in theory or practice. Scientific discourses should *per se* comply with established scientific rules. Therefore, the two minimum criteria of *validity* and *consistency* are proposed to evaluate a discourse. The application of other criteria is also possible. In the case of a violation of scientific criteria, it is arguable if the course of a scientific discourse should still be followed. Finally, the (re)constructed discourses are analyzed. Here, it is examined to what extent the identified discourse quality criteria concerning the (re)construction phase are fulfilled. Subsequently conclusions regarding the evaluation of the quality of the reconstructed quality discourses can be drawn.

The reconstruction and evaluation of model quality discourses according to our procedure model close with the (5) *overall assessment*. Indications for the final evaluation of the model quality discourse are given in order to document the understanding of the investigated subject matter which has been identified in the context of the discourse analysis. In the following, the feasibility and relevance of our introduced procedure model is verified and presented by means of the procedure model's exemplary application. Thereby, the quality discourse regarding the understandability of conceptual models – which is one of the most important model quality dimensions – is reconstructed.

4 Application of the Methodical Framework in the Context of Conceptual Model Understandability

4.1 Identification and Selection of the Model Quality Discourse

During the application of our methodical framework the quality discourse on model understandability is reconstructed under special consideration of contributions using experimental research approaches which is a limited but important part of the discourse on model understandability. However, due to the high demands which are made on the detailed operationalization of variables in experiments, the underlying understanding of conceptual model understandability can be precisely specified. For the identification and delineation of the examined discourse, a systematic review [29] of experimental literature on model understandability was performed using three leading international literature databases (Science Citation Index, Scopus and EBSCO Business Source Premier). The articles were retrieved using the following search terms: "understand*", "comprehen*" combined with "conceptual model", "process model", "data model" and "experiment*". A total of 27 laboratory experiments were identified which serve as the basis for the discourse reconstruction and evaluation.

4.2 Reconstruction of the Model Quality Discourse

First, every article's research design was investigated. In this context, all used variables, their operationalization as well as the measuring instruments concerning the dependent variable ("understandability") and other interesting aspects were analyzed and documented. Table 1 shows an excerpt taken from the overview of documented research designs, variables and measuring instruments used.

Table 1. Investigated variables and the measurement of understandability (excerpt)

Reference	Research design	N	Independent variables	Understandability as dependent variable	Measuring instruments
1. Agarwal et al. 1999 [30]	Laboratory experiment + replication, two groups, randomly assigned participants	36 + 35	*Conceptual modeling approach:* 1. Usage of object-oriented models (structure) 2. Usage of process-oriented models (behaviour)	1. *Accuracy of model comprehension*	*1. Comprehension test:* comprehension score rating participants' answers (7-point-Likert scale) on eight comprehension questions.
2. Bodart et al. 2001 [31]	Three laboratory experiments, mixed designs, randomly assigned participants	52 + 52 + 96	*Representational complexity of a conceptual model:* 1. Mandatory properties representation 2. Optional properties representation	*1.surface-level understanding* *2.deeper-level understanding* (response accuracy and problem-solving)	*1. Seven measures for recall accuracy* (total number of correctly recalled construct instances) *2. Response accuracy* (10 comprehension questions) and three measures for problem-solving performance concerning 9 questions
3. Burton-Jones and Weber 1999 [32]	Laboratory experiment, 2x2 mixed design, randomly assigned participants	67	*Ontological Clarity of ERM* *1.* relationships can have attributes 2. relationships can not have attributes *Domain knowledge of a user*	*1. problem-solving performance* *2. perceived ease of understanding*	*1. problem-solving measurement*: number of acceptable answers to six problem-solving questions 2. *PEU*: Six items derived from common ease of use-instruments
4. Burton-Jones and Meso 2006 [33]	Laboratory experiment, 1*3 between-groups design, randomly assigned participants + replication	57 + 66	*Model decomposition quality* (minimality, determinism, losslessness, weak coupling, strong cohesion)	1. *Actual understanding* (comprehension and problem-solving) 2. *perceived understanding*	*1. problem-solving test:* number of acceptable answers to problem-solving questions and cloze test (participant's ability to complete a narrative of the domain, number of filled blanks) *2. Four items to measure perceived ease-of-understanding*
5. Burton-Jones and Meso 2008 [34]	Laboratory experiment, 2*2 between-groups design, randomly assigned participants	168	*Model decomposition quality* *Multiple forms of information* (information on model content provided by diagrams or narrative)	1. *Perceived ease of understanding* 2. *Surface understanding* (comprehension) 3. *Deep understanding* (problem-solving)	*1. Four items to measure perceived ease-of-understanding* (5-point Likert scale) *2.comprehension test* (number of acceptable answers concerning comprehension questions) *3. problem-solving test* (number of acceptable answers concerning problem-solving questions)
6. …	…	…	…	…	…

The investigation shows that the term model understandability is conceptualized and measured very differently within the 27 laboratory experiments. While Agarwal et al. [30] consider only one dimension of understandability, which they measure by means of comprehension questions, Bodart et al. [31] define different "depths of understanding". *Surface-level understanding* corresponds to correctly recalling model parts while a *deeper-level understanding* is related to correctly answering questions concerning the model content, which is relevant for problem solving. Furthermore, it seems very interesting that many different measuring instruments for model understandability have been used. While in the contribution of Agarwal et al. [30] the answers concerning the comprehension test were documented and their correctness was subjectively judged by the conductor of the experiment using a 7-point Likert scale, in Bodart et al. [31] the number of correctly recalled model parts respectively successfully solved problems was counted. Hence, it can be stated that not only is model understandability defined very differently, but also its measurement significantly varies in experimental research.

The different conceptualizations of model understandability which were found during the reconstruction of the quality discourse have been classified into categories differentiating objectively measured vs. subjectively judged as well as effectiveness- vs. efficiency-related dimensions which form our model understandability reference framework (Table 2). A plus (+) documents a certain understanding in a contribution.

Table 2. Dimensions of understandability (sources in chronological order)

Investigated dimensions of understandability			1. Juhn and Naumann 1985, [35]	2. Palvia et al. 1992, [36]	3. Kim and March 1995, [37]	4. Agarwal et al. 1999, [30]	5. Burton-Jones and Weber 1999, [32]	6. Nordbotten and Crosby 1999, [38]	7. Bodart et al. 2001, [31]	8. Moody 2002, [39]	9. Moody 2004, [40]	10. Serrano et al. 2004, [41]	11. Gemino and Wand 2005, [42]	12. Sarshar and Loos 2005, [43]	13. Poels et al. 2005, [44]	14. Burton-Jones and Meso 2006, [33]	15. Khatri et al. 2006, [45]	16. Mendling et al. 2007, [46]	17. Recker and Dreiling 2007, [47]	18. Serrano et al. 2007, [48]	19. Burton-Jones and Meso 2008, [34]	20. De Lucia et al. 2008, [49]	21. Genero et al. 2008, [50]	22. Mendling and Strembeck 2008, [51]	23. Reijers and Mendling 2008, [52]	24. Vanderfeesten et al. 2008, [53]	25. Recker and Dreiling 2011, [54]	26. Reijers and Mendling 2011, [55]	27. Schalles et al. 2011, [56]
Objective dimensions of understandability	effectiveness	1. Recalling model content							+																				+
		2. Correctly answering questions about model content	+	+	+	+		+	+	+	+		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		3. Problem-solving based on the model content	+				+		+				+			+	+		+		+						+		
		4. Verification of model content			+					+	+																		
	efficiency	5. Time needed to understand a model		+						+	+	+	+						+	+			+				+		+
Subjective dimension of understandability	effectiveness	6. Perceived ease of understanding a model			+		+						+	+		+		+			+		+						

4.3 Validation of Reconstruction

The discourse on model understandability can be very precisely reconstructed. This is mainly related to the circumstance that all participants in this rather limited part of the quality discourse concerning model understandability have to meet the general quality requirements of laboratory experiments regarding the conceptualization and operationalization of investigated constructs. The fact that many of the investigated laboratory experiments have been published in highly ranked publication outlets with high quality standards, such as ISR, JAIS or MISQ, resulted in a reconstruction which could be done without larger difficulties. The high quality of underlying contributions and the full accessibility of construct conceptualizations and operationalization provided good preconditions for a valuable discourse reconstruction. If any uncertainties or different opinions among the authors of this article occurred during the reconstruction, e. g., while classifying the articles according to our framework, these points were discussed and resolved.

4.4 Analysis and Evaluation of the Model Quality Discourse

The reconstruction of the quality discourse shows that the quality dimension model understandability has been very differently defined and interpreted, conceptualized and measured, which allows for the first central finding of our analysis which will also be explained in the following:

1. *Research results on conceptual model understandability are ambiguous and thus hardly comparable without a clear reconstruction of underlying understandability conceptualizations.*

Understandability as an important dimension of model quality and the act of understanding a model are described by several different characteristics and dimensions in the investigated articles, e. g., by correctly recalling model content, problem solving based on models or quickly answering questions about the model content. As shown in Table 2, model understandability has subjective and objective as well as effectiveness- and efficiency-related dimensions. In this context, *correctly answering questions about the model content* seems to be an indisputable dimension of model understandability which is used in 25 of the 27 investigated experiments. Likewise, *problem solving based on the model content, perceived ease of understanding a model* or *time needed to understand a model* are considered important dimensions of understandability while the other two dimensions seem to play a rather tangential role. However, in every case research results concerning model understandability are presented, which makes these results highly ambiguous if the underlying conceptualization of understandability is not made explicit. If, e.g., two experiments compare the understandability of *UML class diagrams* and *entity relationship models* and one of them investigates *perceived ease of understanding* while the other analyzes *time needed to understand a model* and both come to the conclusion that UML class diagrams are easier to understand, these final statements are actually hardly comparable. In order to further clarify the dimensions of understandability and to support the communication about model understandability between researchers, the discourse reconstruction results (Table 2) can serve as a reference framework for the future definition of variables in experimental research on model understandability. There is another interesting observation and a second finding of our analysis:

2. *The understanding and conceptualization of model understandability in IS research has not been further differentiated during the last years (1985-2011).*

It could be intuitively expected that in almost thirty years of research on conceptual model understandability a more differentiated understanding of the topic has evolved. However, this is obviously not the case. Investigated contributions in our research often cite related work but conceptualizations and operationalization used in former work are seldom included in following research. These observations have important implications which will be discussed in the overall assessment of the discourse.

4.5 Overall Assessment

During the exemplary application of the introduced procedure model, the author team takes the perspective of one individual participant in the modeling meta discourse. Therefore, in this article the overall assessment of the discourse is limited to the discourse reconstruction performed here. Our investigation of experimental contributions revealed interesting differences regarding the underlying understanding and measurement of model understandability. The article's central working assumption concerning the conceptualization of quality fully applies to the quality dimension *model understandability*. The results of our discourse reconstruction corroborate the assumption that for a deeper understanding of model quality criteria it is necessary to consider all possible perspectives. While the discourse reconstruction in Table 2 gives an insightful overview of conceptualizations of model understandability and, thus, of the variety of different model quality understandings, it seems very interesting that within 26 years of research obviously no detailed differentiation of the conceptualization of model understandability has been established. From the perspective of cumulative research, this is unfavourable as existing research results on model understandability are not always taken into account and in some cases experimental research in this area seems to begin again "from scratch". Even more striking is the fact that many research results concerning model understandability can hardly be compared as they have different basic assumptions of understandability.

5 Conclusion and Outlook

In this article, we have investigated and demonstrated the relevance of discourse orientation for the reconstruction and evaluation of conceptual model quality discourses. Based on the conclusion that model quality criteria can be interpreted highly differently depending on the discourse participants and that a multi-perspective view is necessary to fully understand model quality in all its facets, a methodical framework has been introduced and exemplarily applied in a case study focusing on laboratory-experimental contributions on the topic of conceptual model understandability.

Thereby, it showed that model understandability as an important model quality criterion is in fact conceptualized and operationalized very differently. In this context, using our methodical framework can considerably support the identification of different quality understandings. Furthermore, the reconstruction results can support the communication between different stakeholders about model quality dimensions and, thus, the realization and development of conceptual models considering the different quality requirements of different stakeholders. However, further research into the usage of our framework in quality discourses is necessary, especially under consideration of a broader variety of discourse contributions such as reviewers' comments on articles, response articles printed in journals etc.

Discourse-oriented investigations and the review of the understandings concerning introduced terms are particularly necessary since, otherwise, scientific results in all likelihood are interpreted ambiguously. This circumstance not only impedes the development of a more precise research but also the expedient accumulation of research results of the IS community, such as reliable empirical relationships for the development of dedicated IS theories. Therefore, a consistent use of terminology is of essential significance. Against this background, it is important to further sharpen the terminology of IS research in general and the terminology of conceptual modeling in particular. In this regard, our discourse-oriented approach can also make a significant contribution.

In practice, the introduced methodical framework is of relevance as well, because a lack of clarity in the language used and ambiguous model quality criteria may have crucial influence on organizational success. The consideration of different perspectives for model quality evaluation and a corresponding systematic advancement of models may significantly improve modeling success in practice. However, a further examination of the procedure model's suitability for practice is needed. As a matter of fact, it should be stated that a non-consistent use of language in organizations is not unusual and organizations may anyway be successful. However, it can be expected that using our approach may improve the creation and enhancement of conceptual models in practice by delivering an overview of relevant quality dimensions. In this context also the transferability of results from the reconstruction of scientific discourses into practice shall be further investigated. It can be assumed that in practice especially the analysis of intra-corporate quality discourses is of importance, while the evaluation of the discourse quality, which seems especially important in a scientific context, presumably is of minor relevance.

In conclusion, an examination and reconstruction of model quality discourses supports the identification of relevant model quality criteria in general and, thus, the development of valuable conceptual models. Against this background it is important that future IS research does not only advance evaluation methods for models but also gains a deeper insight into all relevant dimensions of conceptual modeling by means of discourse-oriented approaches.

Acknowledgement. The research described in this paper was supported by a grant from the German Research Foundation (DFG), project name: "Pluralistische Beurteilung der Qualität von Unternehmensmodellen – Qualitätsdiskurse und Diskursqualität innerhalb der Wirtschaftsinformatik (PluralistiQue)", support code LO 752/4-1. The authors would also like to thank the six anonymous reviewers for their valuable comments which helped to improve this paper.

References

1. Fettke, P.: How Conceptual Modeling Is Used. Communications of the AIS 25, 571–592 (2009)
2. Germonprez, M., Hovorka, D., Gal, U.: Secondary Design: A Case of Behavioral Design Science Research. Journal of the AIS 12, 662–683 (2011)
3. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Eng 55, 243–276 (2005)
4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
5. Frank, U.: Towards a Pluralistic Conception of Research Methods in Information Systems Research. Institut für Informatik und Wirtschaftsinformatik (ICB) der Universität Duisburg-Essen, Essen (2006)
6. Garvin, D.A.: What Does Product Quality Really Mean. Sloan Management Review 26, 25 (1984)
7. Moody, D.L.: Metrics for Evaluating the Quality of Entity Relationship Models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 211–225. Springer, Heidelberg (1998)
8. Krogstie, J., Sindre, G., Jørgensen, H.: Process Models as Knowledge for Action: A Revised Quality Framework. European Journal of Information Systems 15, 91–102 (2006)
9. Frank, U.: Evaluation of Reference Models. In: Fettke, P., Loos, P. (eds.) Reference Modeling for Business Systems Analysis, pp. 118–140. Idea Group, Hershey (2007)
10. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–49. Springer, Heidelberg (2000)
11. Janiesch, C., Brelage, C., Holten, R.: Exploration of Conceptual Models - Application of the GoM Framework. In: Information Resources Management Association International Conference (IRMA), San Diego, CA, USA, pp. 254–257 (2005)
12. Ortner, E., Schienmann, B.: Normative Language Approach - A Framework for Understanding. In: Thalheim, B. (ed.) ER 1996. LNCS, vol. 1157, pp. 261–276. Springer, Heidelberg (1996)
13. Moody, D.L.: Validation of a Method for Representing Large Entity Relationship Models: An Action Research Study. In: Wrycza, S. (ed.) Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland, pp. 391–405 (2002)
14. Moody, D.L., Shanks, G.G.: Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework. Information Systems 28, 619–650 (2003)
15. Sedera, W., Rosemann, M., Gable, G.: Measuring Process Modelling Success. In: Wrycza, S. (ed.) Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland, pp. 331–341 (2002)
16. Schütte, R.: Evaluation of information models (in German). In: Heinrich, L.J., Häntschel, I. (eds.) Evaluation und Evaluationsforschung in der Wirtschaftsinformatik - Handbuch für Praxis, Lehre und Forschung, Oldenbourg, München, Wien, pp. 307–382 (2000)
17. Blommaert, J.: Discourse: a Critical Introduction. Cambridge Univ. Press, Cambr. (2005)
18. Potter, J., Wetherell, M.: Discourse and Social Psychology. Sage, London (2011)
19. Harris, Z., Mattick Jr., P.: Science Sublanguages and the Prospects for a Global Language of Science. Annals of the American Academy of Pol. and Social Science 495, 73–83 (1988)

20. Halpin, T.: Fact-Oriented Modeling: Past, Present and Future. In: Krogstie, J., Opdahl, A.L., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 19–38. Springer, New York (2007)
21. Sindre, G., Opdahl, A.L.: Capturing Dependability Threats in Conceptual Modeling. In: Krogstie, J., Opdahl, A.L., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 247–260. Springer, New York (2007)
22. Auramäki, E., Hirschheim, R., Lyytinen, K.: Modelling Offices Through Discourse Analysis: A Comparison and Evaluation of SAMPO with OSSAD and ICN. The Computer Journal 35, 492–500 (1991)
23. Auramäki, E., Hirschheim, R., Lyytinen, K.: Modelling Offices Through Discourse Analysis: The SAMPO Approach. The Computer Journal 35, 342–352 (1991)
24. Cimiano, P., Reyle, U., Sanic, J.: Ontology-driven discourse analysis for information extraction. Data & Knowl. Eng. 55, 59–83 (2005)
25. Ulrich, W.: Critically Systemic Discourse - A Discursive Approach to Reflective Practice in ISD (Part 2). The Journal of Information Technology Theory and Application (JITTA) 3, 85–106 (2001)
26. Ulrich, W.: Philosophical Staircase for Information Systems Definition, Design, and Development - A Discoursive Approach to Reflective Practice in ISD (Part 1). The Journal of Information Technology Theory and Application (JITTA) 3, 55–84 (2001)
27. Winograd, T.: A Language/Action Perspective on the Design of Cooperative Work. Human-Computer Interaction 3, 3–30 (1987-1988)
28. Hoppenbrouwers, S., Proper, H., van der Weide, T.: Formal Modelling as a Grounded Conversation. In: Goldkuhl, G., Lind, M., Haraldson, S. (eds.) Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP 2005), Kiruna, Sweden, pp. 139–155 (2005)
29. Cooper, H., Hedges, L.V.: Research Synthesis As a Scientific Enterprise. In: Cooper, H., Hedges, L.V. (eds.) The Handbook of Research Synthesis, pp. 3–14. Sage, New York (1994)
30. Agarwal, R., De, P., Sinha, A.P.: Comprehending Object and Process Models: An Empirical Study. IEEE Trans. Softw. Eng. 25, 541–556 (1999)
31. Bodart, F., Patel, A., Sim, M., Weber, R.: Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. Information Systems Research 12, 384–405 (2001)
32. Burton-Jones, A., Weber, R.: Understanding Relationships with Attributes in Entity-Relationship Diagrams. In: De, P., DeGross, J.I. (eds.) Proceedings of the International Conference on Information Systems (ICIS 1999), pp. 214–228. Charlotte, North Carolina (1999)
33. Burton-Jones, A., Meso, P.N.: Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis. Information System Research 17, 38–60 (2006)
34. Burton-Jones, A., Meso, P.N.: The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model. Journal of the AIS 9, 748–802 (2008)
35. Juhn, S., Naumann, J.: The Effectiveness of Data Representation Characteristics on User Validation. In: Proceedings of the International Conference on Information Systems (ICIS 1985), pp. 212–226 (1985)
36. Palvia, T., Lio, C., To, P.: The Impact of Conceptual Data Models on End User Performance. J. Database Manage. 3, 4–14 (1992)

37. Kim, Y.-G., March, S.T.: Comparing Data Modelling Formalisms. Communications of the ACM 38, 103–115 (1995)
38. Nordbotten, J.C., Crosby, M.E.: The Effect of Graphic Style on Data Model Interpretation. Information Systems Journal 9, 139–155 (1999)
39. Moody, D.L.: Complexity Effects on End User Understanding of Data Models: An Experimental Comparison of Large Data Model Representation Methods. In: Wrycza, S. (ed.) Proceedings of the 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland, pp. 482–496 (2002)
40. Moody, D.L.: Cognitive Load Effects on End User Understanding of Conceptual Models: An Experimental Analysis. In: Benczúr, A.A., Demetrovics, J., Gottlob, G. (eds.) ADBIS 2004. LNCS, vol. 3255, pp. 129–143. Springer, Heidelberg (2004)
41. Serrano, M., Calero, C., Trujillo, J., Lujan, S., Piattini, M.: Empirical Validation of Metrics for Conceptual Models of Data Warehouses. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 506–520. Springer, Heidelberg (2004)
42. Gemino, A., Wand, Y.: Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. Data & Knowledge Eng. 55, 301–326 (2005)
43. Sarshar, K., Loos, P.: Comparing the Control-Flow of EPC and Petri Net from the End-User Perspective. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 434–439. Springer, Heidelberg (2005)
44. Poels, G., Gailly, F., Maes, A., Paemeleire, R.: Object Class or Association Class? Testing the User Effect on Cardinality Interpretation. In: Akoka, J., Liddle, S.W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., van den Heuvel, W.-J., Kolp, M., Trujillo, J., Kop, C., Mayr, H.C. (eds.) ER Workshops 2005. LNCS, vol. 3770, pp. 33–42. Springer, Heidelberg (2005)
45. Khatri, V., Vessey, I., Ramesh, V., Clay, P., Park, S.-J.: Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. Information Systems Research 17, 81–99 (2006)
46. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007)
47. 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: Toleman, M., Cater-Steel, A., Roberts, D. (eds.) Proceedings of the 18th Australasian Conference on Inf. Systems, Toowoomba, Australia, pp. 356–366 (2007)
48. Serrano, M., Trujillo, J., Calero, C., Piattini, M.: Metrics for data warehouse conceptual models understandability. Inf. Softw. Technol. 49, 851–870 (2007)
49. De Lucia, A., Gravino, C., Oliveto, R., Tortora, G.: Data model comprehension an empirical comparison of ER and UML class diagrams. In: IEEE International Conference on Program Comprehension, Amsterdam, pp. 93–102 (2008)
50. Genero, M., Poels, G., Piattini, M.: Defining and validating metrics for assessing the understandability of entity-relationship diagrams. Data & Knowl. Eng. 64, 534–557 (2008)
51. Mendling, J., Strembeck, M.: Influence Factors of Understanding Business Process Models. In: Abramowicz, W., Fensel, D. (eds.) Business Information Systems. LNBIP, vol. 7, pp. 142–153. Springer, Berlin (2008)
52. Reijers, H.A., Mendling, J.: Modularity in Process Models: Review and Effects. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 20–35. Springer, Heidelberg (2008)

53. Vanderfeesten, I., Reijers, H.A., Mendling, J., van der Aalst, W.M.P., Cardoso, J.: On a Quest for Good Process Models: The Cross-Connectivity Metric. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 480–494. Springer, Heidelberg (2008)
54. Recker, J., Dreiling, A.: The Effects of Content Presentation Format and User Characteristics on Novice Developers Understanding of Process Models. Communications of the AIS 28, 65–84 (2011)
55. Reijers, H.A., Mendling, J.: A Study Into the Factors That Influence the Understandability of Business Process Models. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 41, 449–462 (2011)
56. Schalles, C., Creagh, J., Rebstock, M.: Usability of Modelling Languages for Model Interpretation: An Empirical Research Report. In: Bernstein, A., Schwabe, G. (eds.) Proceedings of the 10th International Conference on Wirtschaftsinformatik, Zurich, Switzerland, pp. 787–796 (2011)

Semiotic Considerations for the Design of an Agent-Oriented Modelling Language

Brian Henderson-Sellers[1], Graham Low[2], and Cesar Gonzalez-Perez[3]

[1] School of Software, University of Technology,
Sydney, P.O. Box 123, Broadway, NSW 2007, Australia
[2] UNSW School of Information Systems, Technology and Management,
Australian School of Business, University of New South Wales, Sydney,
NSW 2052, Australia
[3] Institute of Heritage Sciences (Incipit)
Spanish National Research Council (CSIC)
Santiago de Compostela, Spain
brian.henderson-sellers@uts.edu.au, g.low@unsw.edu.au,
cesar.gonzalez-perez@incipit.csic.es

Abstract. Building on published guidelines for good design practice as applied to the creation of modelling languages, we consider the creation of an appropriate notation for a domain-specific modelling language for supporting agent-oriented information systems design. We begin by analyzing extant metamodels, in particular that for FAML, in order to visualize these ontological concepts as a concrete syntax that adheres to semiotic principles and good design heuristics. We seek a notation that is easy to understand by industry users, is ontologically correct and is underpinned by some theory – expressed here as the FAML metamodel.

Keywords: concepts, modelling language, notation, agents.

1 Introduction

Information systems design and engineering can benefit from the use of a modelling language, either a general purpose language like ER or UML, or a domain-focussed DSML (domain-specific modelling language). A DSML can have a business domain as its focus e.g. a DSML for banking or for healthcare, usually linked to an ontology [1] or can focus on a particular technology such as services or agents, these latter two being closely linked since most service-oriented architectures are dependent upon agents for their implementation.

In this paper, we utilize published guidelines for good design practice as applied to the creation of modelling languages e.g. [2-5]. Building on these guidelines, we consider the DSML for supporting agent-oriented information systems design. The rationale is that following the rationalization of object-oriented notations in the late 1990s to early 2000s, the agent-oriented software engineering (AOSE) community has sought their own rationalization, in terms of methodology [6], in terms of metamodels

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 422–434, 2012.

[7] and in terms of notation [8]. Here, we focus on the modelling language aspects in terms of a metamodel to define the syntax and the semantics together with a notation to supply the concrete syntax. Of the various agent-oriented modelling languages (AOMLs) in existence (see, e.g., [9]), most are linked to one specific, marketed methodology. There are two schools of thought: (1) to create an AOML that extends the object-oriented UML (Unified Modeling Language) e.g. Agent UML [10] and (2) to recognize that agents have many incompatibilities with objects such that the use of UML would be inappropriate as a starting point e.g. [11].

One language that is not a UML extension (option 2) and is *not* aligned with a single methodological approach is FAML[1] [12], the metamodel of which was designed specifically by returning to basic principles and only using methodology-specific concepts sparingly and only when totally appropriate.

We thus start with the FAML metamodel [12] which, as noted above, aims to be a relatively-generic agent-oriented metamodel whose suitability for supporting modelling language development was demonstrated by evaluating it with respect to several existing methodology-specific metamodels including those of Adelfe, PASSI, Gaia, INGENIAS and Tropos. We endeavour to incorporate the notation of [8] while adhering to semiotic principles and good design heuristics [5]. We seek a notation that is easy to understand by industry users, is ontologically correct and is underpinned by some theory – expressed here as a metamodel [13].

In Section 2, we discuss general design considerations for the concrete syntax of a DSML; and in Section 3 we identify a good starting point in the published literature. In Section 4, we apply these ideas to the domain of agents. We then build on the outcomes by providing a firm proposal for the notation to complement one specific agent modelling language metamodel (FAML). This is then followed by a discussion of the resulting improvements made following a preliminary validation involving a small number of experts. Section 5 concludes with some planned and suggested future work.

2 Design Considerations for the Concrete Syntax of a Modelling Language

A number of authors have offered advice on the key aspects of a good quality modelling language. With the basic assumption (as adopted here) that an appropriate representation of the modelling language semantics uses a graphical syntax, Padgham et al. [8], quoting [14], suggest that such a language's notation should have the following characteristics:

1. Clear mapping of concepts to symbols
2. No overloading of symbols
3. Uniform mapping of concepts to symbols
4. Easy to draw by hand
5. Looks good when printed

[1] FAML is the FAME Agent-oriented Modelling Language. (FAME is an acronym for the Framework for Agent-oriented Method Engineering project.)

6. Must fax and copy well using monochrome images
7. Consistent with past practice
8. Self consistent
9. Distinctions not too subtle
10. Users can remember it
11. Common cases appear simple
12. Suppressible details.

Whilst this list is useful, it shows no influence from semiotics or usability studies. Indeed, when it was argued during an OMG meeting in Austin that the embryonic UML should be subject to pre-standardization usability tests, this proposal was not permitted to be put to the committee. Consequently, several UML symbols and annotations fail when subject to quality assessments e.g. [4] and, furthermore, the lack of provision of any design rationale for UML suggests that "this is acceptable practice even for the industry standard language" [5, p757].

Constantine and Henderson-Sellers [2,3] argue for the need to base symbols to be used in a modelling language on semiotic principles [15], differentiating between indexical signs (directly connected to the referent by physical association), iconic signs (having a likeness to the referent), and symbolic signs (having a connection by convention only). These ideas have been formalized more recently by Moody [5], who argues that, although there are many goals that a notation could aim to meet (e.g. simplicity, expressiveness, naturalness), the primary dependent variable for evaluating and comparing visual notations is cognitive effectiveness. He argues that cognitive effectiveness is not an emergent property but one that must be designed into the notation [16]. He notes that the representational *form* (i.e. the notational symbol) for a concept has equal, or even greater, influence on cognitive effectiveness than its content (including the underlying semantics) (see also [17]) – as confirmed by empirical studies (as referenced in [5]).

Table 1. Principles for designing effective visual notations

Name of Principle	Characteristics of Principle
1. Semiotic clarity	Ensure there is a 1:1 correspondence between semantic constructs and graphical symbols
2. Perceptual discriminability	Clearly distinguish between different symbols
3. Semantic transparency	Use visual representations whose appearance suggests their meaning
4. Complexity management	Include explicit mechanisms for dealing with complexity
5. Cognitive integration	Include explicit mechanisms to support integration of information from different diagrams
6. Visual expressiveness	Use the full range and capacities of visual variables
7. Dual coding	Use text to complement graphics
8. Graphical economy	Ensure the number of different graphical symbols is cognitively manageable
9. Cognitive fit	Use different visual dialects for different tasks and audiences

Moody [5] provides nine principles for designing effective visual notations (Table 1). Semiotic clarity (Principle 1) requires a one-to-one correspondence between construct and symbol. If this is not achieved, 'errors' such as symbol redundancy (several symbols used for one concept), overload (maps to more than one concept), excess (maps to no concept) or deficit (a concept has no symbolic representation) can occur (see [18] for an analysis of UML from this viewpoint). Symbol discriminability (Principle 2) is enhanced by ensuring that shapes likely to be juxtaposed are from different 'families' (e.g. curvilinear, polygonal), possibly also with additional use of colour or labels. Semantic transparency (Principle 3) uses cues to meaning, especially iconic signs rather than symbolic ones [2] and/or mnemonics. Good notational sets deal well with complexity (Principle 4) for which there are several options, although often none are used [5]. Various forms of decomposition are possible especially varying the abstraction levels by means of generalization/specialization and meronymic structures. Principle 5 (Table 1) encourages the use of systems of diagrams rather than one single diagram. Visual expressiveness (Principle 6) recommends the use of the large number of visual variables. Symbols can be differentiated in terms of eight variables [5,19] - see Table 2. Use of colour when possible (e.g. now for printouts on black & white printers) can enhance the speed of recognition by a factor of three e.g. [20]. Text can also be useful as a complement to graphics (Principle 7). Cognitive limits to the number of symbols are recognized in Principle 8, especially problematical for novices [21]. Finally, Principle 9 focusses on the application of cognitive fit theory e.g. [22] to visual software engineering notations. Cognitive differences exist between novices and experts such that a notation should have different subsets or dialects for such widely different user types.

Table 2. Eight characteristics of symbols of [19] from which an individual symbol can be constructed

Variable Kind	**Variable**
Planar	Horizontal position; Vertical position
Rational	Shape; Size; Colour; Brightness; Texture; Orientation

Moody [5] notes that the application of the nine principles of Table 1 can lead to some problems resulting from interactions between principles, possibly leading to the need to make trade-offs or, conversely, to benefit by selecting identified synergies.

The symbols used in the notational package are thus clearly of primary importance. These symbols are sometimes called the 'primary notation' in contrast to the 'secondary notation' that adds further visual clues that aid in comprehension (in particular, differentiating between novices and experts: [23]) whilst leaving the semantics unchanged [24]. These clues, which are called aesthetic guidelines in [25], include overall layout of model elements (investigated for ER diagrams in [26]), particularly focussing on line crossings, visual distance and back pointers [24]. Although clearly valuable, such concerns are outside the scope of our initial design of a domain-specific modelling language (here for agent-oriented system design).

3 Previous Proposal

As noted above, FAML offers a methodology-independent metamodel but no notation; in contrast, Padgham et al. [8] offer only a notation (but no metamodel) aimed at being methodology-independent, or at least useful for a small number (four) selected agent-oriented methodologies: O-MaSE, Tropos, Prometheus and PASSI. Thus, we start with this unified notation [8], as depicted in Fig. 1, as one major input and baseline notation to the creation of a notational component for FAML.

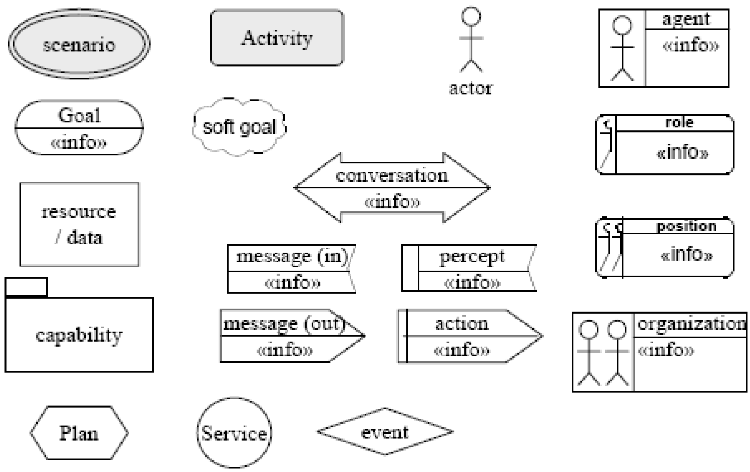


Fig. 1. Symbols proposed for AOSE notation (after [8])

Using the guidelines for language design discussed in Section 2, we can immediately identify some potential areas for improvement of this symbol set:

- The symbol proposed for role and position has *two* differentiating features – the roundangle shape and the half stick figure. It is generally the case that one differentiating feature will suffice.
- Activity, role and position all have a basic roundangle shape. Whilst the last two are of the same 'family', the visual similarity between role, position and activity could lead a novice to believe that an activity is some kind of agent.
- The shapes of the Activity icon and the Goal icon are topologically equivalent and hence easily mistaken
- Goal (an ellipse) and soft goal (a cloud) aren't obviously related visually. Also a metamodel (which is lacking in this study) would presumably argue for a supertype of Goal with two subtypes of Soft Goal and Hard Goal (at least if compatibility with Tropos is an aim as it is in this cited study).
- The shape chosen for Resource/data has little or no semiotic value, despite some semblance to the entity/class symbol in ER/UML, and should be improved or replaced
- Goal and service are too similar visually and are topologically equivalent

Despite these concerns, and in order to retain a community spirit, we propose utilizing the good elements of this notation in our proposal. In particular, we note an elegant consistency in terms of the use of angularity to symbols to all elements that relate to the interactions between agents and their environments (event, action, message, percept and conversation). However, this angularity is also seen in the hexagon used for Plan, which is definitely an agent-internal concept.

4 Initial Proposal for FAML Notation

The published version of FAML [12] is, as a metamodel, expressed primarily as a suite of four class diagrams: agent-external and agent-internal - both with one metamodel fragment for design and one for runtime. A list of the most important concepts, for which we seek a notational symbol here, are listed below in Table 3.

4.1 First Proposal for FAML Notation

Since the FAML metamodel consists of a significant number of concepts, mere attribution of a symbol to each concept would violate Principle 8 in Table 1. Consequently, we adopt the strategy used previously [27] in the development of a notation [28] for the International Standard 24744 [29]: Software Engineering Metamodel for Development Methodologies (SEMDM). This strategy involved identifying ‘families’ of concepts that may be linked visually in the notation. In the creation of the SEMDM notation, families were readily identified in terms of a small number of ‘top-level’ classes each of which has several subtypes i.e. it is structured more like an ontology than a domain model [30]. In contrast, FAML uses generalization very sparsely so that there are very few ‘top classes’ that can be highlighted as the root of a family set of symbols. We therefore turn to the textual definitions given in [12] to supplement the FAML metamodel diagrams in determining likely ‘family sets’.

In this first attempt at creating a notation that not only supports the concepts of FAML but also the concepts symbolized in Fig. 1, we proposed the following ‘family sets’ (Table 3) with the intention that within a family the symbols chosen will have same intrinsic geometric characteristics and (when used) the same colour and conversely that no two families will share characteristics (Principle 2). Colour is only used as an enhancer and the symbols chosen should equally be comprehensible if viewed in black and white. We also took into account all the semiotic advice discussed in Section 2. Overall, we have embodied Principles 1, 2 and 8 and to some degree Principles 3 and 6. As noted in Section 6, complexity management (Principle 4) and cognitive integration (Principle 5) only apply when diagram types rather than single symbols are discussed (see also [31, 32]). Finally, whilst supporting the ideas of Principle 9, empirical evidence resulting from extensive use of the proposed FAML notation is a pre-requisite for such future improvements.

Table 3. Initially proposed families and their members

Family	**Members**	**Shape**	**Colour (optional)**	**Source and/or influence for notation**
Agents and roles	Agent, role, group, position, organization	Circle atop mask or rectangle	Yellow	INGENIAS [33]
Tasks and plans	ActionSpecification, FAML task, PlanSpecification	Curvilinear	Green	ISO/IEC [28]
Messages	Conversation, MessageIn, MessageOut	Arrow heads	B/W	Padgham et al. [8]
Events and Resources	Event, Resource	Triangular	Blue/green	
Goals	Hard goal, Soft goal, Belief	Complex curvilinear	Brown	
Ontology	Ontology, Service, Capability	Polygonal	Dark blue	

The first group of symbols is that of agents and roles. The proposed notations for these concepts use the role mask for the Role and, for the agent representations, a rectangle, each of which is surmounted by one or more circles. The number of circles is suggestive of the scale (Fig. 2) and the concatenation of the circle/{rectangle or mask} has semiotic value in being perceived as being related to an active entity, such as a person or actor (Principle 3).

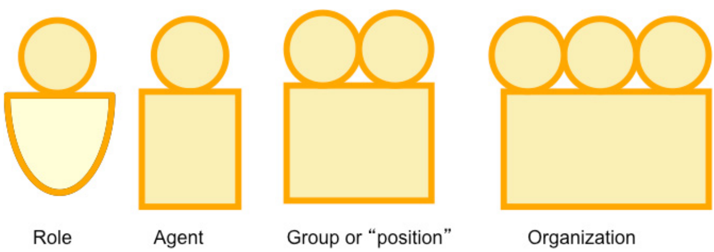


Fig. 2. First draft for Role and Agent icons

The basic shape for the second group is chosen to be analogous to the work unit symbols of ISO/IEC 24744, as is the colour: green. Fig. 3 shows these. Although all three icons are topologically equivalent, there is sufficient visual differentiation, endorsed by the differences in the shades of green when colour is utilized. Contrasting the symbols in Fig. 3 with those in Fig. 2 underlines our use of Principle 2 (as do all later diagrams in this paper).

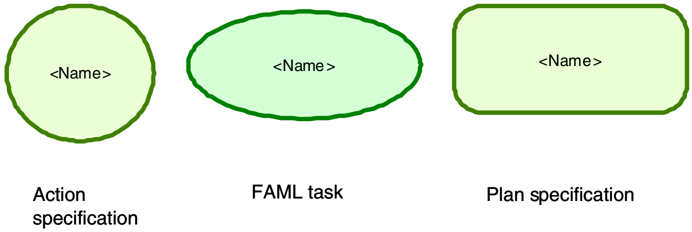


Fig. 3. First draft for Action, task and plan

Symbols for communication structures, shown in Fig. 4, use no colour but rather rely on semiotics for their visualization. Messages in and out have a triangle on their side indicating the direction (in or out) of the message. Two way messages or conversations (a.k.a. interaction protocols) are shown in an obvious manner with a bidirectional arrow (Principle 3 again).

Events and resources are coupled together rather tentatively into a family (Table 3). These are depicted in Fig. 5. Neither symbol has any obvious semiotic value but they do adhere to Principle 2 in being distinct from all other shapes proposed in this notational suite for FAML.

Fig. 6 shows goals (hard and soft) coupled with beliefs since an Agent has a Mental State in the FAML metamodel and this mental state has both goals and beliefs. The notation for both hard and soft goals derives from the i* notation [34] as used in the Tropos methodology [35]. The fill colour (when used) is brown (Table 3).

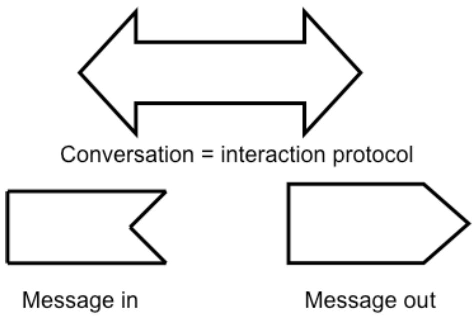


Fig. 4. Communication symbols

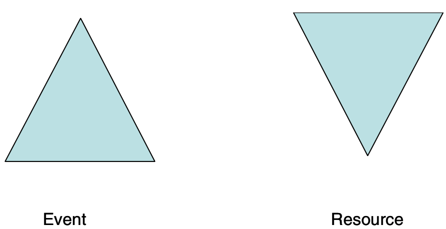


Fig. 5. Symbols for Event and Resource

The final family is that of ontology, service and capability (Fig. 7). These are grouped together because both Service and Ontology are linked to Role in FAML. Capability is a useful concept [8] and not obviously related to any of the previous families (Figs. 2-6). This family is represented by the same colour (blue) and a regular polygon of various designs.

Although not expressed as a family, there is a clear affinity between Scenarios (as well as use cases, often used in agent modelling) and Actors. Here, we simply adopt the symbols proposed in Fig. 1.

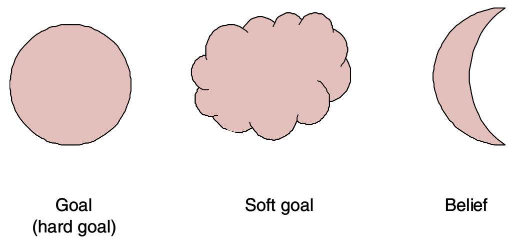


Fig. 6. Symbols for goals and beliefs

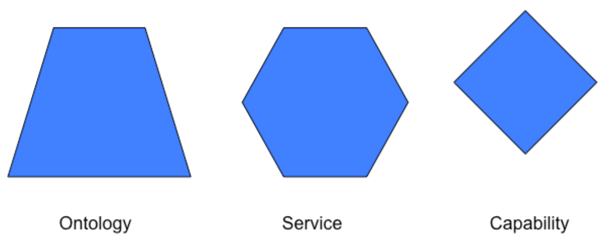


Fig. 7. Symbols for ontology, service and capability

4.2 Revised Proposal for FAML Notation

The original version of the FAML notation was evaluated by a number of experts in multi-agent systems design as well as Human-Computer Interaction (HCI) experts, notably Professor Mats Lind of the University of Uppsala and Dr Haris Mouratidis of the University of East London. As a result of their comments, it became clear that changes were needed. Their main concerns were as follows.

Fig. 2 equated the concepts of group (from INGENIAS) and position (which, in Tropos, represents a set of roles played by one agent). A challenge was made as to whether this is a valid assertion. This remains to be resolved.

Secondly, although ActionSpecification in Fig. 3 and HardGoal in Fig. 6 are clearly topologically equivalent, without colour or context they would be hard to discriminate between them. Consequently, in the revisions of the FAML notation subsequent to these HCI experts' comments, we changed the Action Specification symbol to that shown in Fig. 8. For the MessageIn and MessageOut symbols adopted from Fig. 1, it was noted that although the triangle (representing a directional arrowhead) was clearly discernible in MessageOut, this was less so in MessageIn. The modifications shown in Fig. 8 make this much clearer.

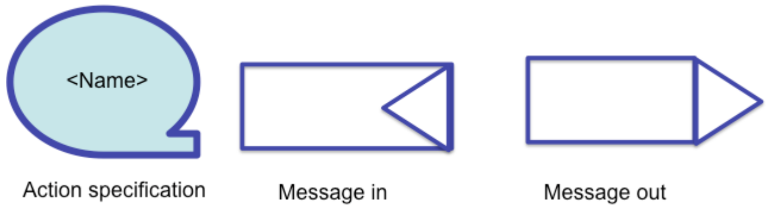


Fig. 8. Revised symbols for ActionSpecification, MessageIn and MessageOut

For Event and Resource, these icon shapes (Fig. 5) are not able to be discriminated under rotation. These need to be replaced, it was suggested. Shapes vaguely reminiscent of the WorkProduct family of SEMDM were selected and shown in Fig.9.

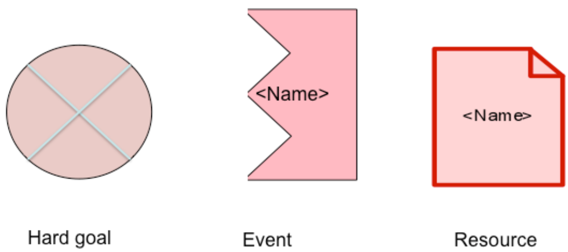


Fig. 9. Revised symbols for HardGoal, Event and Resource

4.3 Further Iterations and Possible Revisions

Since formulating these symbol sets, other (anonymous) reviews have been undertaken (in terms of the reviewers of the paper prior to its publication). Some comments received question whether the three symbols in Fig. 8 are really improvements on those in our first draft (Fig. 4). Another comment received was that the new Event symbol in Fig. 9 was now unnecessary since the ambiguity (from a rotational point of view) between Event and Resource (of Fig. 5) is obviated with the introduction of the new Resource symbol of Fig. 9. However, we should note that the family of Event (from Fig. 5) and Resource (as in Fig. 9) has now lost its coherence. This suggests further work is necessary.

Secondly, we would also propose re-opening the symbols for goals (Fig. 6 cf. Fig. 9). Goals come to most AOSE methodologies, including FAML, from i* [34], in which a hard goal is an ellipse and a soft goal a cloud. An improved semiotic value might be obtainable by using a cloud for both kinds of goal: with a solid outline for a hard goal and a dashed outline for a soft goal.

All these proposals, resulting from anonymous review and the further research engendered, merit further consideration, both from a theoretical viewpoint (in terms of the nine principles of Table 1) and by means of empirical studies (see suggestions in Section 5 below). Further insights are also sought in terms of visual design ideas presented in seminal texts such as [36].

5 Summary and Future Work

In Section 2, we discussed Moody's [5] nine principles of good notational design. Although the notation for FAML outlined in Section 3 follows these principles, it has not yet been tested in terms of likely diagram types e.g. [32] that require the superposition of several symbols and/or the inclusion of one symbol within another. We plan to identify diagram types relevant to AOSE and FAML. This will likely lead to the superposition of symbols – our analysis of ISO/IEC [28] notation for SEMDM [31] is planned to be mirrored in our further development to the FAML notation. In parallel, we also are planning an experiment in which creative design students are asked to supply appropriate symbols for the FAML concepts. We intend to conduct a comprehensive evaluation of the enhanced FAML notation using a large group (20 plus) of experts followed by a usability study in a real world case study.

Secondly, we have only offered individual symbols. As suggested above, an obvious next step is to evaluate *diagram types*, which combine several different symbols, in terms of potential ambiguities (semantic or visual) and with respect to possible mechanisms for hiding/showing detail i.e. supporting diagrams at different granularities.

Our third line of future enquiry is to evaluate the possible utility of reformulating the FAML metamodel by use of powertypes [37, 38]. The current incarnation of the FAML metamodel contains several cases of class pairs that can be satisfactorily described as powertype patterns, namely Agent and AgentDefinition, Facet and FacetDefinition, Plan and PlanSpecification, Action and ActionSpecification. This means that, for each of these pairs, one class represents entities that are subtypes of the other class. For example, focusing on the Agent/AgentDefinition powertype pattern, specific agent definitions (i.e. instances of AgentDefinition) are also subtypes (i.e. subclasses) of Agent, being dual entities called clabjects [39]. Powertype patterns would serve to keep a tighter link between the system and environment domains in FAML, since each class in a class pair pertains to a different domain. Such a revision of the metamodel is a topic of future work in the light of significant current deliberations regarding the most appropriate approach to multilevel metamodelling [1, 40, 41]; it would also allow for a better alignment of FAML with ISO/IEC 24744, which uses the powertype pattern as its underlying architecture.

Acknowledgements. This is paper number 12/03 of the Centre for Object Technology Applications and Research within the Centre for Human Centred Technology Design of the University of Technology, Sydney. Thanks to all those who commented on early drafts of the FAML notation, especially Professor Mats Lind and Dr Haris Mouratidis. BH-S and GCL also wish to acknowledge the support of the Australian Research Council through grant DP0878172.

References

1. Henderson-Sellers, B., Gonzalez-Perez, C.: Mathematical Relationships Between Domain-specific Modelling Languages, Ontologies and Metamodels. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) Research Directions in Domain Engineering. Springer, Berlin (2012)

2. Constantine, L.L., Henderson-Sellers, B.: Notation Matters: Part 1 - Framing the Issues. Report on Object Analysis and Design 2(3), 25–29 (1995)
3. Constantine, L.L., Henderson-Sellers, B.: Notation Matters: Part 2 - Applying the Principles. Report on Object Analysis and Design 2(4), 20–23 (1995)
4. Moody, D.L., van Hillegersberg, J.: Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In: Proc. First Int'l Conf. Software Language Eng. (2008)
5. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Eng. 35(6), 756–779 (2009)
6. Henderson-Sellers, B.: Creating a Comprehensive Agent-oriented Methodology - Using Method Engineering and the OPEN Metamodel. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Methodologies, ch.13, pp. 368–397. Idea Group, Hershey (2005)
7. Bernon, C., Cossentino, M., Gleizes, M.-P., Turci, P., Zambonelli, F.: A Study of Some Multi-agent Meta-models. In: Odell, J.J., Giorgini, P., Müller, J.P. (eds.) AOSE 2004. LNCS, vol. 3382, pp. 62–77. Springer, Heidelberg (2005)
8. Padgham, L., Winikoff, M., DeLoach, S., Cossentino, M.: A Unified Graphical Notation for AOSE. In: Luck, M., Gomez-Sanz, J.J. (eds.) AOSE 2008. LNCS, vol. 5386, pp. 116–130. Springer, Heidelberg (2009)
9. Henderson-Sellers, B., Giorgini, P. (eds.): Agent-Oriented Methodologies, p. 413. Idea Group (2005)
10. Bauer, B., Müller, J.P., Odell, J.: Agent UML: A Formalism for Specifying Multiagent Software Systems. In: Ciancarini, P., Wooldridge, M.J. (eds.) AOSE 2000. LNCS, vol. 1957, pp. 91–103. Springer, Heidelberg (2001)
11. Silva, V., Garcia, A., Brandão, A., Chavez, C., Lucena, C., Alencar, P.: Taming Agents and Objects in Software Engineering. In: Garcia, A., et al. (eds.) SELMAS 2002. LNCS, vol. 2603, pp. 1–26. Springer, Heidelberg (2003)
12. Beydoun, G., Low, G., Henderson-Sellers, B., Mouratidis, H., Gomez-Sanz, J., Pavon, J., Gonzalez-Perez, C.: FAML: A Generic Metamodel for MAS Development. IEEE Trans. Software Eng. 35(6), 841–863 (2009)
13. Dupuy-Chessa, S.: Quality in Ubiquitous Information System Design. In: Flory, A., Collard, M. (eds.) Procs. Third International Conference on Research Challenges in Information Science (RCIS 2009), pp. 343–352. IEEE, Los Alamitos (2009)
14. Rumbaugh, J.: Notation Notes: Principles for Choosing Notation. Journal of Object Oriented Programming 9(2), 11–14 (1996)
15. Peirce, C.S.: Collected Papers v. 2, paragraphs 243-263, written (circa 1903)
16. Larkin, J.H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11(1), 65–100 (1987)
17. Siau, K.: Informational and Computational Equivalence in Comparing Informational Modelling Methods. J. Database Management 15(1), 73–86 (2004)
18. Opdahl, A., Henderson-Sellers, B.: Ontological Evaluation of the UML using the Bunge-Wand-Weber Model. Software and Systems Modelling 1(1), 43–67 (2002)
19. Bertin, J.: Semiology of Graphics: Diagrams, Networks, Maps. Univ. Wisconsin Press, Madison (1983)
20. Treisman, A.: Perception Grouping and Attention in Visual Search for Features and for Objects. J. Experimental Psychology: Human Perception and Performance 194–214 (1982)
21. Nordbotten, J.C., Crosby, M.E.: The Effect of Graphic Style on Data Model Interpretation. Information Systems J. 9(2), 139–156 (1999)

22. Vessey, I.: Cognitive Fit: A Theory-based Analysis of the Graphs Versus Tables Literature. Decision Science 22, 219–240 (1991)
23. Petre, M.: Why Looking isn't Always Seeing: Readership Skills and Graphical Programming. Comms ACM 38(6), 33–44 (1995)
24. Schrepfer, M., Wolf, J., Mendling, J., Reijers, H.A.: The Impact of Secondary Notation on Process Model Understanding. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 161–175. Springer, Heidelberg (2009)
25. Unhelkar, B., Henderson-Sellers, B.: Applying Syntax, Semantics and Aesthetic Checks to Verifying and Validating the Quality of UML Models. In: Procs. IRMA 2005. Idea Group, Hershey (2005)
26. Hay, D.C.: Data Model Patterns. Conventions of Thought. Dorset House Publishing Company (1996)
27. Gonzalez-Perez, C., Henderson-Sellers, B.: Metamodelling for Software Engineering, p. 210. J. Wiley & Sons, Chichester (2008)
28. ISO/IEC: 24744 Software Engineering – Metamodel for Development Methodologies Annex A – Notation, International Organization for Standardization/International Electrotechnical Commission, Geneva (2010)
29. ISO/IEC,: Software Engineering – Metamodel for Software Development. ISO/IEC 24744, Geneva, Switzerland (2007)
30. Gonzalez-Perez, C., Henderson-Sellers, B.: An ontology for software development methodologies and endeavours. In: Calero, C., Ruiz, F., Piattini, M. (eds.) Ontologies in Software Engineering and Software Technology, ch.4, pp. 123–152. Springer, Heidelberg (2006)
31. Sousa, K., Vanderdonckt, J., Henderson-Sellers, B., Gonzalez-Perez, C.: Evaluating a Graphical Notation for Modelling Software Development Methodologies. Journal of Visual Languages and Computing (in press, 2012)
32. Henderson-Sellers, B.: Consolidating Diagram Types from Several Agent-oriented Methodologies. In: Fujita, H. (ed.) New Trends in Software Methodologies, Tools and Techniques - Proceedings of the 9th SoMeT_10. Frontiers in Artificial Intelligence and Applications, vol. 217, pp. 293–345. IOS Press, The Netherlands (2010)
33. Pavon, J., Gomez-Sanz, J.J., Fuentes, R.: The INGENIAS Methodology and Tools. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Methodologies, ch. IX, pp. 236–276. Idea Group, Hershey (2005)
34. Yu, E.S.-K.: Modelling Strategic Relationships for Process Reengineering. unpubl. PhD thesis, University of Toronto, p. 124 (1995)
35. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopolous, J., Perini, A.: Tropos: An Agent-oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems 8(3), 203–236 (2004)
36. Ware, C.: Visual Thinking for Design, p. 197. Morgan Kaufmann, Amsterdam (2008)
37. Odell, J.: Power Types. Journal of Object-Oriented Programming 7(2), 8–12 (1994)
38. Gonzalez-Perez, C., Henderson-Sellers, B.: A Powertype-based Metamodelling Framework. Software and Systems Modeling 5(1), 72–90 (2006)
39. Atkinson, C.: Supporting and Applying the UML Conceptual Framework. In: Bézivin, J., Muller, P.-A. (eds.) UML 1998. LNCS, vol. 1618, pp. 21–36. Springer, Heidelberg (1999)
40. Atkinson, C., Gutheil, M., Kennel, B.: A Flexible Infrastructure for Multilevel Language Engineering. IEEE Trans. Software Eng. 35(6), 742–755 (2009)
41. Eriksson, O., Henderson-Sellers, B., Ågerfalk, P.: Ontological and Linguistic Metamodelling Revisited – A Language Use Approach. J. AIS (2011) (submitted)

Technology-Enhanced Support for Learning Conceptual Modeling

Gayane Sedrakyan and Monique Snoeck

Katholieke Universiteit Leuven,
Management Information Systems,
Naamsestraat 69, 3000 Leuven
{gayane.sedrakyan,monique.snoeck}@econ.kuleuven.be

Abstract. This paper describes an optimized didactic environment to support and improve learning achievements for conceptual modeling. In particular, it describes computer-aided techniques to address various learning challenges observed in the teaching process such as: hybrid background of students, enrollment of a large number of students, the complexity of industrial tools and difficulties in abstract thinking. The didactic environment has been developed and subsequently optimized in the context of the course Architecture and Modeling of Management Information Systems. It includes 1) diagnostic testing with automated feedback 2) an adapted modeling tool 3) an MDA based simulation feature. The didactic tools were evaluated positively by the students and a positive impact was observed on the student's capabilities to construct object-oriented conceptual models.

Keywords: teaching business domain modeling, conceptual model, enterprise modeling, computer aided modeling, modeling tool, automated consistency control, managing knowledge diversity, automated feedback, model driven architecture, simulation, prototyping, executable models.

1 Educational Context

The main goal of the course "Architecture and Modeling of Management Information Systems"[1] is to familiarize the students with modern methods and techniques of Object-Oriented Analysis and Design for Business Information Systems, to let them understand the relation between an information system and the organisational aspects of an enterprise, and to let them acquire sufficient skills of developing an enterprise model as basis of a business information system. This paper describes a didactic environment developed and subsequently optimized based on our experiences from observations the student achievements in the course over a period of 5 years, similar issues found in related research, and constant feedback from 300 students overall.

[1] The course's page can be found on
`http://onderwijsaanbod.kuleuven.be/syllabi/e/D0I71AE.htm`

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 435–449, 2012.

It proposes an extension to the techniques previously presented by Snoeck et al. [1]. The methodology used is based on the concepts of Merode[2].

The remainder of the paper is structured as follows. The second section presents the challenges observed in a learning process. Section 3 gives an overview of the related work and subsequently formulates our research question. Sections 4 to 6 describe optimization techniques for addressing the observed challenges. Section 4 describes a diagnostic testing environment allowing to detect various gaps in students' prerequisite knowledge in a self-regulating manner. Section 5 defines the goals of the adapted modeling tool and gives the overview of modeling techniques that support these goals as far as the modeling process and associated difficulties are concerned. Section 6 describes the prototyping feature that allows for model simulation and concludes with explaining how the tool improves the learning cycle through automated feedbacks and increased transparency between a model and its prototype. Section 7 then reports on our experiences and the evaluation of the didactic tools by the students. Finally, section 8 suggests some future directions.

2 Problem Domain

In view of the high demand for skilled IT professionals with skills in Enterprise Architecture, the faculty of Business and Economics decided to offer a one year program conducting a non-technical training in information management, with an emphasis on the efficient and effective application and management of information technology in various business contexts. The program admits students with an academic bachelor degree of any field and has been very successful in attracting students. As a result of the high number of students with diverse backgrounds, several challenges were observed hindering the realization of the course objectives which was previously taught to a small homogenous group of students with a bachelor in Information Systems. The major challenges observed include: A) hybrid backgrounds of students resulting in various gaps in prerequisite knowledge, subsequently leading to different levels of motivation; B) enrollment of a large number of students making it time-consuming to provide frequent personalized feedback; up to more conceptual issues that include C) the complexity of industry modeling tools making these less effective in supporting a teaching process; D) theoretical knowledge being at high level of abstraction which makes it difficult to bridge the gap between abstract model and a concrete Information System, especially for the students who have never programmed before and thus lack in technical insight. In this paper we focus on computer-aided techniques to remediate the major difficulties of a learning cycle while also striving to improve satisfactory learning outcomes. Our approach includes an optimized didactic environment that is well adapted to learning goals and is intelligent enough to guide throughout a modeling process. More specifically, the rest of this paper aims to describe *how the learning*

[2] Merode is an Object Oriented Enterprise Modeling method. Its name is the abbreviation of Model driven, Existence dependency Relation, Object oriented DEvelopment. Cfr. `http://merode.econ.kuleuven.be`

cycle of a novice up to medium experienced learner can benefit from the combination of adaptation, automation and simulation techniques in the context of addressing the targeted learning challenges.

3 Related Work

Several authors have pointed out the difficulties that come along with teaching a heterogeneous group of students in the context of analysis and design of business information systems [2-6] indicating that not handling substantial differences in prior knowledge between students will result in large differences in motivation and learning outcomes. To address this [8] proposes adapting teaching according to cumulative data on students' understanding and difficulties, while also highlighting the advantage of exploiting peer knowledge in addressing the diversity issue. Many researchers and practitioners also point to problems related to the UML [2-4] which has become a widely accepted standard for modeling. According to the complexity metric presented by [4] due to human cognitive load limitations the diversity of constructs and diagrams of the UML score from 2 to 11 times more complex than those of other methods. Furthermore, [9] points out that most of the modeling languages are too "noisy" with various concepts, that can result in misusing concepts and creation of unintended models, i.e. models that use the language concepts in a way not intended for the modeling domain. As proposed by the constructivist approach [7] the method of dialogue is the optimal way to address learning difficulties by delivering personalized feedback. However, as it can be concluded from the above-mentioned, teaching with industry UML tools would make it time-consuming in terms of addressing how-to-use questions, thus creating a risk to potentially mask the real objectives of the course. Our approach proposes automation technique to deliver feedbacks throughout the learning cycle. The use of a technology-enabled dialogue can improve the learning process in two ways. Firstly, the didactic tools can perform some routine tasks which were previously done by the instructor, leaving the instructor more time for in-depth discussions with the students (cfr. challenges A, B, C). Secondly and more importantly, the calculating power of a computer enables feedback and testing possibilities a human instructor cannot provide for (cfr. challenge B). Among the other fundamental deficiencies of UML is that it is unclear how to combine interactive, structural and behavioral aspects together in a single view [10]. As observed, this can lead to difficulties in mastering the behavioral aspect of modeling (cfr. challenge D). One of the effective approaches for improving the learning process of dynamic aspect includes the use of simulation technique. Benefits of simulation in optimizing the feedback cycle for learning modeling are demonstrated by [16]. To apply the technique we use the model-driven approach. Besides being able to transform a model into its executable form within the shortest development cycle which is essential in the context of limited timeframe of the course, model driven development also allows to enhance the understanding of the rationale behind system design decisions, as well as improve the correctness and completeness of a model [18]. We will therefore apply the adaptation technique to build a modeling environment that is 1) able to support model-driven transformation 2) can alleviate the problem of "noisiness" of UML by limiting to a restricted UML with a limited number of views 3) is intelligent enough to guide the student throughout modeling process.

Table 1. Coverage matrix of techniques used per challenge

	Diversity (A)	Large number of students (B)	Complexity of UML tools (C)	Abstract thinking (D)
Diagnostic Test	automated feedback	automated feed-back	automated feed-back	
Modeling tool	adaptation	automated feed-back	adaptation	
Prototyping tool	simulation	automated feed-back	simulation	simulation / au-tomated feedback

Finally, to operationalize these techniques, we propose a didactic environment that includes 1) automated diagnosing tests allowing to deal with prerequisite knowledge gaps in a self-regulating manner through automated feedbacks; 2) a modeling tool adapted to the learning goals with built-in intelligence; 3) a simulation feature providing increased transparency between a model and its dynamic behavior. Table 1 shows the coverage matrix of techniques used per challenge within the didactic environment.

4 Diagnostic Testing

To achieve the objectives of the course students need to complete a set of modeling exercises. Exercises start with a textual description of some user requirements, which should be interpreted by the student and transformed into an object-oriented conceptual model. Given the fact that such user requirements can and will be interpreted in different ways (because of their informal aspect) by students, students should receive individual feedback on their solution. Unfortunately, both the number of students enrolled (cfr. challenge B) and the use of paper and pencil exercises hamper the realizations of these (course) objectives. Furthermore, students have very different levels of prior knowledge at the beginning of the course (cfr. challenge A). To address the problems of various gaps in students' prerequisite knowledge (cfr. challenge A) a self-regulating approach was stimulated among students to detect and recover possible gaps. This was achieved by developing a testing environment to motivate and help students to discover to which extent they lack prerequisite knowledge. A sample diagnostic test examining knowledge in ER modeling is shown in Fig. 1. Through the tests, students are guided by immediate automated feedbacks as well as tips for improvements. Examples of automated feedbacks to student are shown in Fig. 2. The method showed positive impact on the learning process in terms of stimulating a self-regulated manner towards prerequisite knowledge.

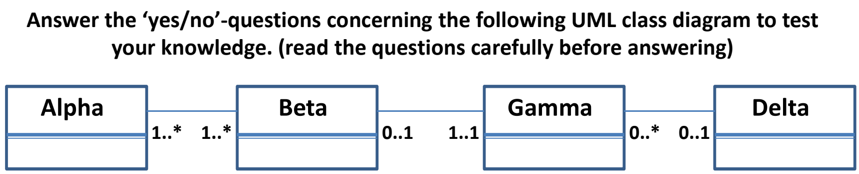


Fig. 1. Example of a diagnostic test

Students with deficient prerequisite were offered extra sessions at the start of the course. Subsequently, the smaller number of (the right) students attending these sessions facilitated the dialogue with the teacher. With the diagnostic environment, however, the number of participants increased, and multiple self-testing cycles were registered by the system for the same student indicating gradually improving results with each cycle.

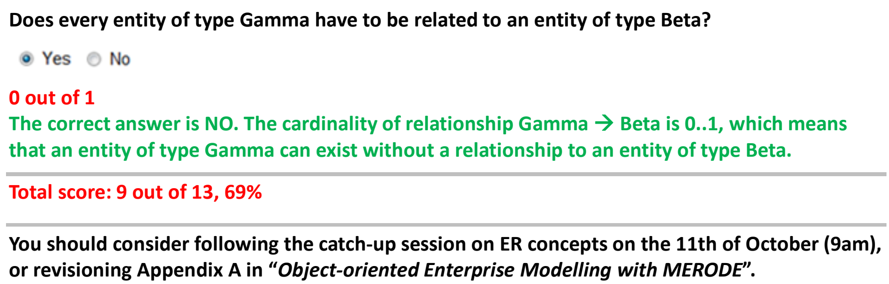


Fig. 2. Example of an automated feedback of a diagnostic test

Additionally, the diagnostic environment allows to store cumulative data about the students prerequisite knowledge. Having data about students understanding, skills and difficulties, in turn, helps to adapt teaching accordingly [8]. In addition to the diagnostic tests, throughout the semester peer dialogue was used as an alternative source for feedbacks in the context of group assignments by means of review sessions among the groups. Diversity in this context, as observed by several studies, can create an advantage for exploiting a "peer expertise". Furthermore, peer dialogues can affect the learning process in a variety of ways by 1) creating an opportunity for broadening student's mind with alternative perspectives and tactics, 2) communication being conducted on the "same language" that makes concepts easy to grasp and critique less stressful to accept, 3) enabling to revise or reject hypothesis, 4) developing detachment of judgment (about work in relation to standards) by commenting on the works of peers [8]. Furthermore, peer reviews for group solutions allow to teach students skills "to read models, not only to write them" [3]. In addition to the offline peer reviews, peer communication was also supported by enabling technologies, such as Wikis.

5 Adapted Modeling Tool

Development of an own adapted modeling environment was motivated by the need for addressing the learning challenges related to the complexity of industry tools as well as the need to support an MDE-based prototyping feature allowing to visualize the dynamic aspect of a model (see section 6 on prototyping). Advantages of own environment over the industry tools include: 1) modeling techniques adapted to learning goals 2) restricted number of easily traceable views allowing to complete a modeling task through incremental and iterative steps, 3) automated feedbacks and built-in model checking mechanism that guides a student throughout a modeling process, 4) a prototyping feature allowing to visualize/validate a model based on its dynamic aspect (cfr. challenge D).

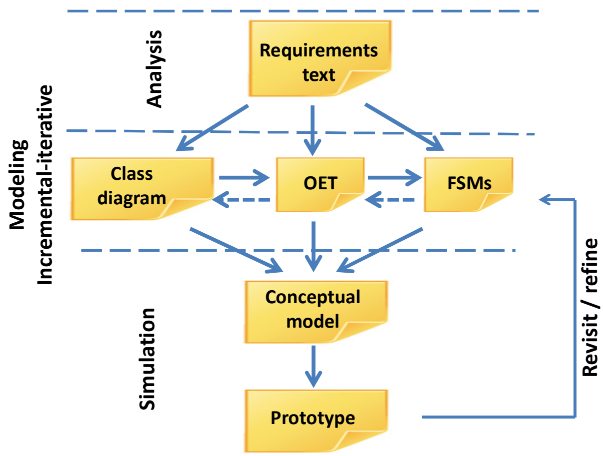
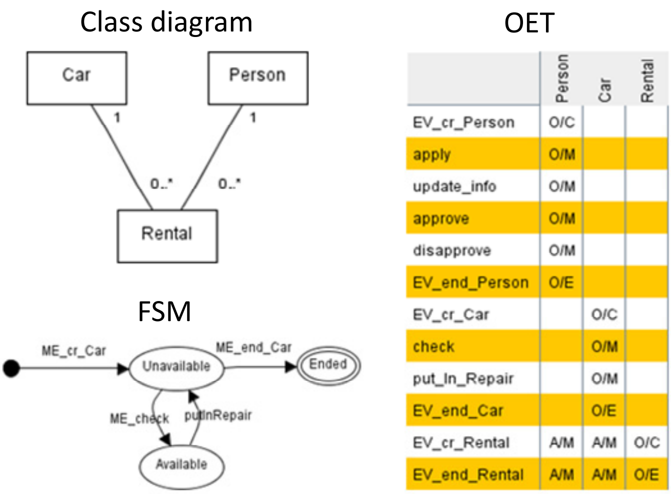


Fig. 3. Artefacts and modeling cycle with Merode

Fig. 3. depicts the artefacts and modeling cycle with Merode within the proposed adapted environment. An object-oriented business model typically consists of several views that together define a platform independent model that is a formal representation of the user requirements. In the modeling method used in this course, an enterprise domain model consists of a class diagram, an interaction model and a number of state charts. The class diagram is a restricted form of UML class diagram: the types of associations are limited to binary associations, with a cardinality of 1 to many or 1 to 1. Many to many associations need to be converted to an intermediate class. The interaction model consists of an Object-Event Table (OET), created according to the principles of Merode [11]. It represents a kind of CRUD-matrix[3], a

[3] CRUD-matrix is a table containing functions of an application and with each function entry indicating the type of interaction with a database for that function: Create, Read, Update or Delete.

technique borrowed from Information Engineering [12]. In Merode, "business events" represent atomic actions from the real world in which one or more domain objects can participate. Each business event is assigned an owner class indicated by an "O/" preceding the kind of involvement (**C**reate, **M**odify, **E**nd). The other participants are considered as "Associated" participants and have the C, M or E preceded by "A/".



	Person	Car	Rental
EV_cr_Person	O/C		
apply	O/M		
update_info	O/M		
approve	O/M		
disapprove	O/M		
EV_end_Person	O/E		
EV_cr_Car		O/C	
check		O/M	
put_In_Repair		O/M	
EV_end_Car		O/E	
EV_cr_Rental	A/M	A/M	O/C
EV_end_Rental	A/M	A/M	O/E

Fig. 4. Examples of class diagram, Object-Event Table (OET) and a Finite State Machine (FSM) supported by JMermaid modeling tool

The finite state machines allow the object type to impose sequence constraints on the business events it is involved in. Multiple Finites State Machines (FSMs) allow to model independent aspects as parallel machines.

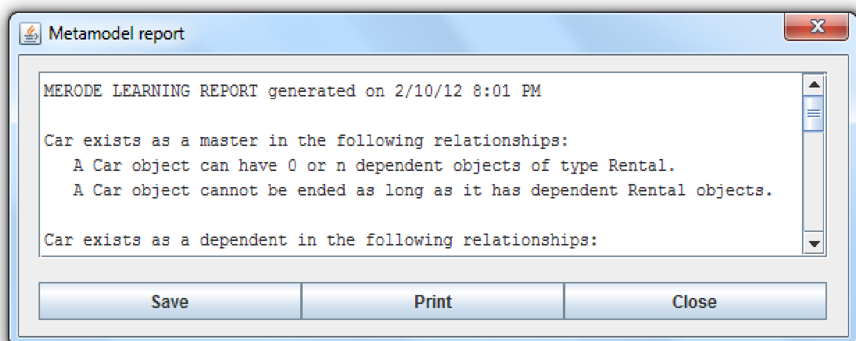


Fig. 5. Example of learning report

While modeling, student should receive interactive feedback on the quality of the developed model. The graphical editors of the adapted JMermaid[4] tool have built-in intelligence that prevent from entering inconsistent specifications. In order to provide students with an elementary form of feedback, a model-to-text feature which converts a model to plain English has been provided. In previous years we experienced that "reading" the data view aloud (like "This model says that… Is that what you meant?"), very often was sufficient to make students realize obvious mistakes. Especially students who are totally unfamiliar with modeling benefit from this feature. This model-to-text facility is called the "learning report" and has been implemented for the data view (see Fig. 5). Students can request for a learning report to pop up each time when drawing an association between classes, or can request a learning report for the data view as a whole.

A second feature concerns the tool's intelligence for managing consistency between the three views of the universe of discourse: the data view, the behavioral view and the interaction view. In its initial form, the tool followed a "consistency-by-construction" approach [13], [14]. In this approach, each time when entering specifications in one view, specifications that can be derived for other views are automatically generated by the tool. As an example, one of the design guidelines states that when defining a class, one should provide at least one method to create instances of that class and one method to terminate instances. Also the behavioral modeling with finite state machines has been augmented with a number of verification tools. The first set of tools act on a finite state machine and report about anomalies in the diagram: forward and backward inaccessible state, non-determinism and missing methods. Next, if multiple finite state machines are used to model parallel aspects of a single class, the student can request the tool to calculate the global behavior implied by the parallel composition of the individual finite state machines. Finally, the tool allows a student to run a check over the complete model to generate a tip-based report on the overall internal consistency of a model.

6 Validation through Simulation

One of the major goals of the course is that students achieve the capability to mentally transform the abstract concepts into a concrete Information System (D). We delegated this to the method of simulation because of the recognized advantage of the latter to improve feedback cycles for learning the dynamic aspect of a model. The advantage of learning based on system dynamics is emphasized by many studies. In particular, among other benefits, simulation makes it possible to 1) trace from model to its effects in the concrete system, 2) conduct experiments and test the system for several alternatives with "what-if" scenarios, 3) increase model accuracy through detection of errors at early phases of engineering, and finally 4) it motivates the use of a simulator with the need to gain enough experience that usually cannot be gained by reading or lecturing alone [15], [16].

[4] Cfr. `http://merode.econ.kuleuven.be/mermaid.aspx`

6.1 MDA-Based Prototyping with Merode

Although many tools are able to generate fairly easily simple data-oriented prototypes no tool is capable of generating prototypes that are able to simulate both the data-aspects and the behavioral aspects contained in the finite state machines. To master the behavioral aspect of a domain model, the student needs to grasp how the overall behavior of a system results from the individual behavior of objects, all running in parallel while behaving according to the finite state machine associated with their class and synchronizing when jointly involved in business events. Since the behavioral aspects of the model is reported as the most difficult subject to master, being able to simulate also the behavioral aspects is of utmost importance. Hence, in order to assist the student in building a mental model of both the data and the behavioral view, the prototyping feature generates a Java application using a code generator based on the Model Driven Architecture (MDA) [17]. Transformation to prototype code can be achieved through a single click: the student receives a ZIP-archive, containing both a compiled application in executable JAR format and the source-code.

6.2 Relationships between Model and Prototype

Before we discuss the transformed behavior of model we give a short overview of the structure of a generated prototype application. It consists of three layers, a graphical user interface (GUI) layer, an event handling layer and a persistence layer.

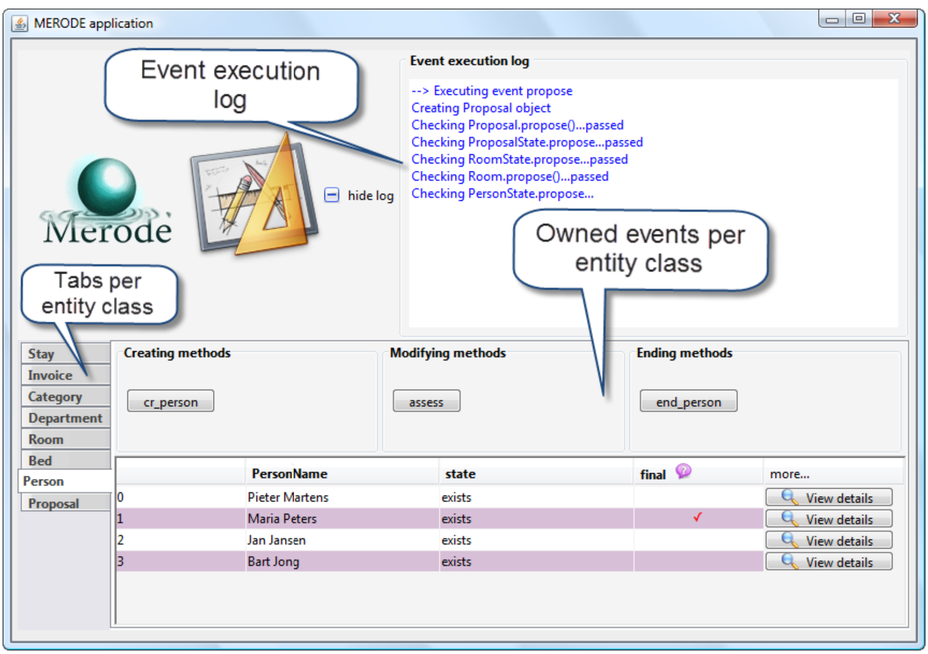


Fig. 6. Main window of a prototype application

The graphical user interface has only basic functionality like triggering the creating and ending of objects, and triggering other business events. The GUI layer is built on top of the event handling layer. The task of the latter layer is to handle all events correctly by managing the appropriate interactions with the objects in the persistence layer. Student interacts with the generated application through the graphical user interface (GUI).

The event handling layer consists of a collection of so called event handlers. The working of an event handler can be described in four steps: 1) Upon an event execution call the event handler 'asks' every participating object (the participants to a business event that have been specified in the Object-Event Table) whether all preconditions set by the object are met. For example, associations between classes will lead to preconditions to maintain referential integrity; 2) Similarly to the previous step the event handler retrieves from every participating object its current state (or reference to the corresponding state object) and checks whether that state allows further processing of the event; 3) If all results of the tasks in step 1 and 2 are positive, the event handler invokes the methods in the participating objects, i.e. corresponding event triggered in response to processing the originally called event in the specific object; 4) next, if all results of previous steps are positive, the event handler executes the method in all participating objects retrieved in step 2 to implement the state modifications (according to the triggered event). While executing a business event in a prototype application users (or students) can follow in an event execution log frame what is happening in the upper right corner of the generated application.

6.3 Increased Transparency between Model and Prototype

One of the methods of promoting better insight on the dynamic aspect of modeling was considered to be the increased transparency of the links between the model and its generated prototype application. During the course it was observed that students experience significant difficulties in linking the knowledge on validation rules gained from theory with the background process of validation in the prototype application. The connection proved to be not obvious with warning and error messages packaged with the generated prototype. Initially those violations were reported to students in the form of error popups (e.g. "Can't execute event: object in wrong state.") which were not easily interpreted by students, causing confusions up to the feeling that the prototype is not behaving as expected. This was partially because students also tend to draw links between the prototype application and its desirable behavior as would be expected in reality, at that level not being skillful enough to link to the errors coming from the model they designed themselves one reason being that during testing the prototype, the model was well hidden from the sight and mind. To make an analogy: a student walking in a (prototype) house discovering that there is no stair leading to the upper floor would simply conclude that the prototype was wrongly built, whereas (s)he should then check whether (s)he actually designed a stair on the plan for the house. Thus, while considering automated feedbacks for a prototype, the major emphasis was put on the failures caused by invalid calls of events. We believed that this would significantly improve the learning process, since the improved system of error reporting, in turn, would motivate to use the prototype application for better understanding, and subsequently for improving learning achievements with a purpose to detect defects. This will subsequently stimulate several improvement cycles of revisiting and refining a model while also allowing to gain a solid knowledge on dynamic behavioral aspect and mastering the necessary skills for modeling.

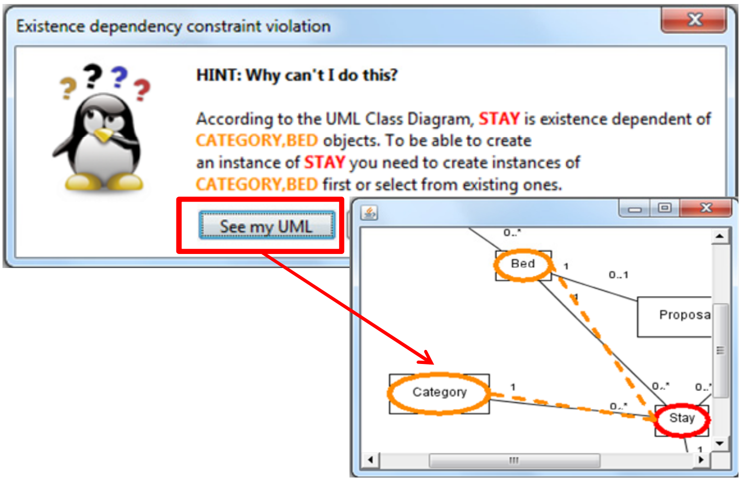


Fig. 7. Example of an automated feedback on 'mandatory one' rule violation

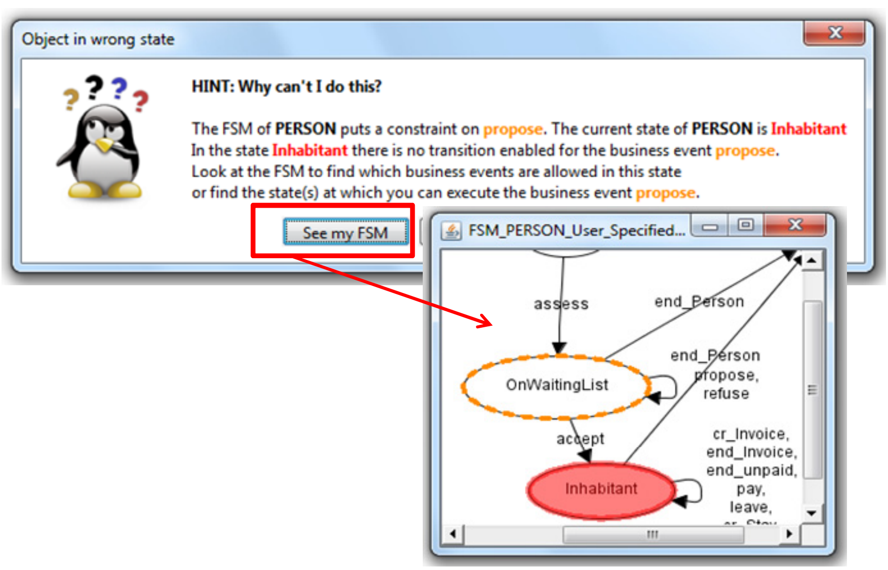


Fig. 8. Automated feedback on event execution being refused by object state

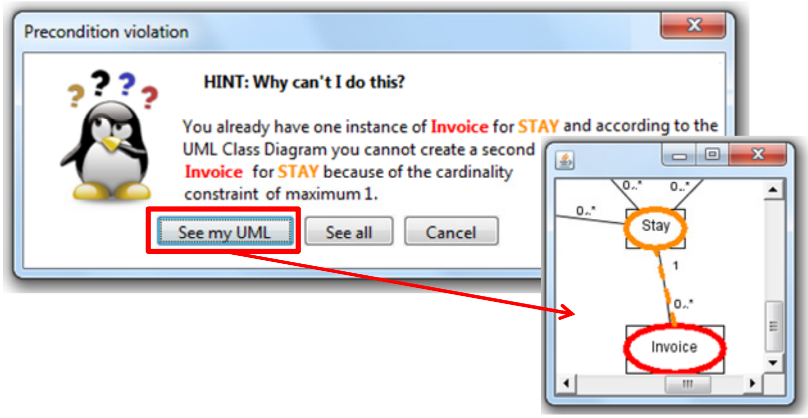


Fig. 9. Example of feedback on cardinality constraint violation

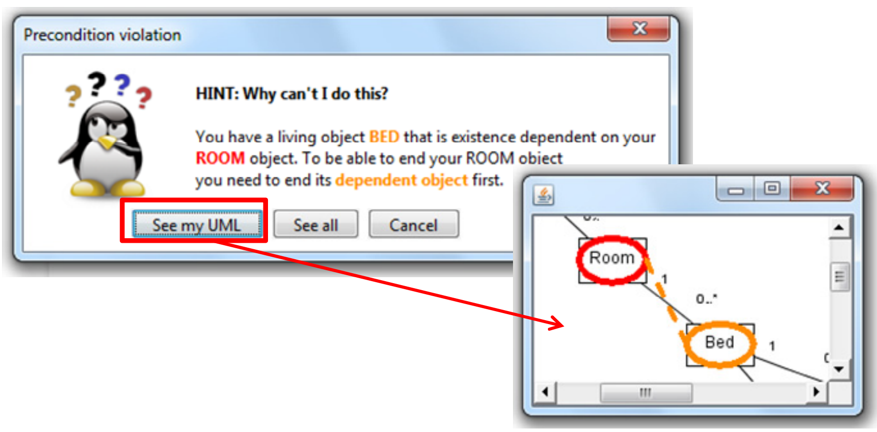


Fig. 10. Example of feedback on ending event failure due to living referring object

With the implementation of tip-based and visualized error reports students are gently guided into the causes of errors through: 1) more descriptive explanations in the error popups with an indication of the names of objects and/or events causing the problem; the explanations are reported in the coloring format linking to their visual representations 2) visual representation of the links between the errors and designed model 3) an image gallery that adds extra comfort by linking to the conceptual model and making it easy to navigate between a prototype and snapshot views of a model.

The set of implemented visualizations include feedbacks on failures that can result from: 1) Mandatory one cardinality violation: an object is attempted to be created before the objects it needs to refer to are created or associated (see Fig. 7). 2) wrong state of the claiming object: event execution fails because the state of the owner object or one of the associated participating objects refuse the execution of an event (see Fig. 8). 3) Cardinality violation: create-event execution fails due to a cardinality constraint of maximum 1 (see Fig. 9). 4) Referential Integrity: ending-event execution fails due to existing referring objects. Example of automated report is shown on Fig. 10. The image gallery adds extra comfort by linking to all the views of the conceptual model, thus eliminating the need to switch between the application view and JMermaid modeling tool views.

7 Evaluation and Experiences

The optimization of the didactic environment is based on constant feedbacks from students, experiments including proposed features and observations of a progress curve of delivered results from tasks and exams. The students were requested to evaluate each feature of the didactic tools on a range of 1 (not useful) to 5 (very useful). Table 2 shows the results of the evaluation for three years of teaching based on the answers of 92 participated students. The empty cells indicate that the feature was not available at the time of evaluation.

Table 2. Evaluation of didactic environment by students

How helpful were the following features ? (1 = Not ○ 5 = very helpful)	Year 1	Year 2	Year 3
Diagnostic Tests	-	-	4,42
Peer Feedbacks	-	-	3,29
Show learning dialog after inserting dependency	3,6	3,6	3,94
Generate creating / ending events automatically	4,6	4,6	4,18
Generate creating / ending methods automatically	4,7	4,5	4.22
Create default finite state machines	3,8	3,8	3,95
Generate model report	3,7	3,3	3,7
Verify All: Consistency Report	4,1	4,2	3,97
OET: Check methods tool	4,4	4,5	3,54
FSM: Check FSM tool	3,5	4,3	3,97
FSM: Calculated FSM tool	2,6	4	3,39
Is the use of JMermaid tool in general helpful in learning the method?	3,9	4,5	4,14
Code Generator / Rapid Prototyping	-	3,7	3,46
Graphical visualizations of errors in prototype	-	-	3,87

With slight variations from year to year, all the features score largely above 3 (neutral), meaning that they are perceived as useful. The only concern was the popularity of the prototyping feature among the students. Previously the simulation was achieved through a chain of several transformation and execution steps before being able to run the prototype. The prototyping process was in addition complicated by an extra dependency of a generated prototype on an application server. Furthermore, the graphical visualizations of errors were implemented as an optional plugin students could extend their prototypes with. Due to their low technical skills, students experienced various difficulties throughout the simulation process chain, which made the major part of students reluctant in using the feature mostly resulting in “didn’t use” answer while evaluating the feature. Despite these early problems the prototyping and errors’ visualizations were rated above average (3,46 and 3,87 out of 5). Furthermore, a little experiment conducted with students before and after the use of simulated model resulted in the positive correction of 1,16 in the interpretation of a model, increasing from 7,63 to 8,59 in a range of 0-10. In the meantime, the problems with the simulation chain have been solved by providing students with an all-in-one package allowing to generate and start a prototype with a single click from the student side. We therefore expect this tool to score even better in 2012 resulting in a much higher positive correction. Yet the growth in average grade on exams is another indicator of improved learning achievements along with significantly increased difficulty level of tasks that students are capable to handle.

8 Future Directions

Although the applied techniques showed improvement in learning achievements, the didactic environment can be improved further and several issues still remain unaddressed. In the future we intend to further increase the intelligence of our didactic tools. Some possible directions include allowing increased interaction with a prototype generator to promote better technical insight. Taking into consideration the differences in prior knowledge the code generator can be extended to support different levels of technical expertise (cfr. challenge A) such as novice, intermediate, and advanced. This may be achieved by allowing a novice user to modify a simple GUI component or a popup message for the generated prototype, followed by intermediate level interaction such as a built-in Query Tool enabling manual queries to be sent to the prototype database , up to more advanced level management, such as (re)defining the structure of generated application, altering the database storage, modifying or adding templates in the template pool of the generator, e.g. a student may be allowed to add invariants and constraints through a template manager interface . We believe those can be good incentives motivating the students to model and experiment with the prototype while mastering skills that are not only related to the architecture and modeling aspects but also involve the domain of other courses taught in parallel, such as database systems, object-oriented programming, etc. thus stimulating the use of integrated knowledge which in turn will establish a more pragmatic and comprehensive focus among the students. Automation technique can be further extended to provide guidance throughout the requirements analysis phase. In parallel with experiments in an educational environment we consider validation and improvement of the proposed environment from the industry perspective. Yet another direction could include be the integration of UML and JMermaid, enabling the latter to process an output of a standard UML tool.

References

1. Snoeck, M., Haesen, R., Buelens, H., De Backer, M., Monsieur, G.: Computer Aided Modelling Exercises. Journal Informatics in Education 6(1), 231–248 (2007)
2. Siau, K., Loo, P.-P.: Identifying difficulties in learning UML. Information Systems Management 23(3), 43–51 (2006)
3. Moisan, S., Rigault, J.-P.: Teaching Object-Oriented Modeling and UML to Various Audiences. In: Ghosh, S. (ed.) MODELS 2009. LNCS, vol. 6002, pp. 40–54. Springer, Heidelberg (2010)
4. Erickson, J., Siau, K.: Can UML Be Simplified? Practitioner Use of UML in Separate Domains. In: Proceedings EMMSAD 2007, Trondheim, Norway, pp. 87–96 (2007)
5. Wang, S.: Teaching the object-oriented approach for Business Information Systems Analysis and Design. Journal of Informatics Education and Research 4(1) (2002)
6. Secundo, G., Margherita, A., Passiante, G.: Hybrid Knowledge, Process and Technology: A Way to Managing Learners' Diversity, Research, Reflections and Innovations in Integrating ICT in Education. FORMATEX, Badajoz (2009)
7. Hadjerrouit, S.: Object-Oriented Software Development Education: a Constructivist Framework. Informatics in Education 4(2), 167–192 (2005)
8. Nicol, D.J., Macfarlane-Dick, D.: Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. Studies in Higher Education 31(2), 199–218 (2006)
9. Buckl, S., Matthes, F., Schweda, C.M.: A Meta-language for EA Information Modeling – State-of-the-Art and Requirements Elicitation. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 169–181. Springer, Heidelberg (2010)
10. Gustas, R.: Conceptual Modeling and Integration of Static and Dynamic Aspects of Service Architectures. In: Sicilia, M.-A., Kop, C., Sartori, F. (eds.) ONTOSE 2010. LNBIP, vol. 62, pp. 17–32. Springer, Heidelberg (2010)
11. Snoeck, M., Dedene, G.: Existence dependency: the key to semantic integrity between structural and behavioural aspects of object types. IEEE Trans. Software Eng. 24(4), 233–251 (1998)
12. Martin, J.: Strategic Data Planning Methods. Prentice-Hall, Inc., Englewood Cliffs (1982)
13. Snoeck, M., Michiels, C., Dedene, G.: Consistency by Construction: The Case of MERODE. In: Jeusfeld, M.A., Pastor, Ó. (eds.) ER Workshops 2003. LNCS, vol. 2814, pp. 105–117. Springer, Heidelberg (2003)
14. Haesen, R., Snoeck, M.: Implementing Consistency Management Techniques for Conceptual Modeling. Accepted for UML2004: 7th Conference in the UML Series, Lisbon, Portugal, October 10-15 (2004)
15. Merrill, D., Collofello, J.S.: Improving Software Project Management Skills Using a Software Project Simulator. Frontiers in Education Conference (1997)
16. Neu, H., Becker-Kornstaedt, U.: Learning and Understanding a Software Process through Simulation of Its Underlying Model. In: Proceedings of LSO, pp. 81–93 (2002)
17. OMG, Model-Driven Architecture, `http://www.omg.org/mda/`
18. Zikra, I., Stirna, J., Zdravkovic, J.: Analyzing the Integration between Requirements and Models in Model Driven Development. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 342–356. Springer, Heidelberg (2011)

Author Index