Ilia Bider   Khaled Gaaloul
John Krogstie   Selmin Nurcan
Henderik A. Proper   Rainer Schmidt
Pnina Soffer (Eds.)

LNBIP 175

Enterprise, Business-Process and Information Systems Modeling

15th International Conference, BPMDS 2014
19th International Conference, EMMSAD 2014
Held at CAiSE 2014, Thessaloniki, Greece, June 16–17, 2014
Proceedings

Springer

Lecture Notes in Business Information Processing 175

Series Editors

Ilia Bider Khaled Gaaloul
John Krogstie Selmin Nurcan
Henderik A. Proper Rainer Schmidt
Pnina Soffer (Eds.)

Enterprise, Business-Process and Information Systems Modeling

15th International Conference, BPMDS 2014
19th International Conference, EMMSAD 2014
Held at CAiSE 2014, Thessaloniki, Greece, June 16-17, 2014
Proceedings

Springer

Volume Editors

Ilia Bider
DSV Stockholm University, Sweden
E-mail: ilia@dsv.su.se

Khaled Gaaloul
Public Research Centre Henri Tudor, Luxembourg
E-mail: khaled.gaaloul@tudor.lu

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

Selmin Nurcan
Université de Paris 1 Panthéon - Sorbonne, France
E-mail: selmin.nurcan@univ-paris1.fr

Henderik A. Proper
Public Research Centre Henri Tudor, Luxembourg
E-mail: erik.proper@tudor.lu

Rainer Schmidt
Munich University of Applied Sciences, Germany
E-mail: rainer.schmidt@hm.edu

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

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-662-43744-5 e-ISBN 978-3-662-43745-2
DOI 10.1007/978-3-662-43745-2
Springer Heidelberg New York Dordrecht London
Library of Congress Control Number: 2014940058

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 15th International Conference on Business Process Modeling, Development and Support (BPMDS 2014), and the 19th International Conference on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2014). The two working conferences are introduced below.

BPMDS 2014

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: http://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 (but they still need to be explicitly relevant to BPMDS). The focus theme for BPMDS 2014 idea papers was "The Human Perspective in Business Processes." The human perspective is of essence since it is humans who drive business processes rather than computers. This perspective plays a major role in all phases of BPMDS. The human perspective in business process modeling relates to the individual who creates a process model, to the communication among people that is facilitated by the business process model during and after the modeling process, and to the social process of collaborative modeling. The human perspective in business process design relates to the kind of interaction/collaboration/coordination/cooperation that should be implemented in the business process or to specific human-related aspects of the business process itself and their representations in models. Human perspective in business process support relates to all social aspects of the business process and its management in an organization.

BPMDS 2014 received 48 submissions from 23 countries (Australia, Austria, Belgium, Brazil, Cameroon, Canada, Cuba, Estonia, Finland, France, Germany, Greece, Italy, Latvia, Morocco, The Netherlands, Portugal, Spain, Sweden, Tunisia, Turkey, UK, Venezuela). Management of the paper submission and reviews was supported by the EasyChair conference system. Each paper received at least three reviews. Eventually, 20 high-quality papers were selected; among them one experience report and five 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 Modeling as a Human-Driven Process
- Representing the Human Perspective of Business Processes
- Supporting Humans in Business Processes
- Variability-Enabling Process Models
- Various Models for Various Process Perspectives
- BPMDS in Practice

We wish to thank all the people who submitted papers to BPMDS 2014 for having shared their work with us, as well as the members of the BPMDS 2014 Program Committee, who made a remarkable effort reviewing the large number of submissions. We also thank the organizers of CAiSE 2014 for their help with the organization of the event, and IFIP WG8.1 for the support.

April 2014

Ilia Bider
Selmin Nurcan
Rainer Schmidt
Pnina Soffer

EMMSAD 2014

The field of information systems analysis and design includes numerous information modeling methods and notations (e.g., ER, ORM, UML, ArchiMate, EPC, BPMN, DEMO to mention a few) that are typically evolving. Even with some attempts toward standardization (e.g., UML for object-oriented software design), new modeling methods are constantly being introduced, many of which differ only marginally from previous approaches. These ongoing changes significantly impact the way information systems, enterprises, and business processes are being analyzed and designed in practice.

The EMMSAD conference focuses on exploring, evaluating, and enhancing modeling methods and methodologies for the analysis and design of information systems, enterprises, and business processes. Although the need for such studies is well recognized, there is a paucity of such research in the literature. The objective of the EMMSAD conference series is to provide a forum for researchers and practitioners interested in modeling methods for systems analysis and design to meet and exchange research ideas and results. It also provides the participants an opportunity to present their research papers and experience reports, and to take part in open discussions.

Whereas modeling techniques traditionally have been used to create intermediate artifacts in systems analysis and design, modern modeling methodologies take a more active approach. For instance in business process management (BPM), model-driven software engineering, domain-specific modeling (DSM), enterprise architecture (EA), enterprise modeling (EM), interactive models and active knowledge modeling, the models are used directly as part of the information system of the organization. At the same time, similar modeling techniques are also used for sense-making and communication, model simulation, quality assurance, and requirements specification in connection to more traditional forms of information systems and enterprise development. Since modeling techniques are used in such a large variety of tasks with different goals, it is hard to assess whether a model is sufficiently good to achieve the goals. To provide guidance in this process, knowledge for understanding the quality of models and modeling languages is needed.

EMMSAD 2014 was the 19th in a series of events, previously held in Heraklion, Barcelona, Pisa, Heidelberg, Stockholm, Interlaken, Toronto, Velden, Riga, Porto, Luxembourg, Trondheim, Montpellier, Amsterdam, Hammamet, London, Gdansk, and Valencia.

This year we had 27 papers submitted with authors from 21 countries and five continents (Austria, Belgium, Brazil, Canada, Colombia, Egypt, Estonia, France, Germany, Greece, India, Israel, Luxembourg, The Netherlands, Norway, Portugal, South Africa, Spain, Sweden, Tunisia, UK). After an extensive review process by a distinguished international Program Committee, with each paper receiving at least three reviews, we accepted the ten full and three short papers that appear in these proceedings. Congratulations to the successful authors!

Apart from the contribution by paper authors, the quality of EMMSAD 2014 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 of CAiSE and WG 8.1. Other co-sponsors this year were the Enterprise Architecture Network, the ORM Foundation, and AIS-SIGSAND

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

April 2014

John Krogstie
Khaled Gaaloul
Erik Proper

BPMSD 2014 Organization

Program Committee

Joao Paulo Almeida	Federal University of Espirito Santo, Brazil
Eric Andonoff	Centre national de la recherche scientifique, France
Renata Araujo	UNIRIO, Brazil
Judith Barrios Albornoz	University of Los Andes, Colombia
Ilia Bider	Stockholm University, Sweden
Karsten Boehm	FH Kufstein Tirol - University of Applied Science, Austria
Pere Botella	Universitat Politecnica de Catalunya, Spain
Dirk Fahland	Technische Universiteit Eindhoven, The Netherlands
Luciano García-Bañuelos	University of Tartu, Estonia
Johny Ghattas	Smart Path Ltd.,
Claude Godart	Université de Lorraine, France
Renata Guizzardi	Universidade Federal do Espírito Santo, Brazil
Paul Johannesson	Royal Institute of Technology, Sweden
Marite Kirikova	Riga Technical University, Latvia
Agnes Koschmider	Karlsruhe Institute of Technology, Germany
Marcello La Rosa	Queensland University of Technology, Australia
Jan Mendling	Wirtschaftsuniversität Wien, Austria
Bela Mutschler	University of Applied Sciences Ravensburg-Weingarten, Germany
Michael Möhring	University of Jena, Germany
Jens Nimis	University of Applied Sciences Karlsruhe, Germany
Selmin Nurcan	Université de Paris 1 Panthéon - Sorbonne, France
Oscar Pastor Lopez	Polytechnical University of Valencia, Spain
Gil Regev	Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Manfred Reichert	University of Ulm, Germany
Hajo A. Reijers	Eindhoven University of Technology, The Netherlands
Iris Reinhartz-Berger	University of Haifa, Israel
Stefanie Rinderle-Ma	University of Vienna, Austria
Colette Rolland	Université de Paris 1 Panthéon - Sorbonne, France
Michael Rosemann	Queensland University of Technology, Australia

Shazia Sadiq	The University of Queensland, Australia
Rainer Schmidt	Munich University of Applied Sciences, Germany
Pnina Soffer	University of Haifa, Israel
Lars Taxen	Linköping University, Sweden
Roland Ukor	FirstLinq Ltd.
Barbara Weber	University of Innsbruck, Austria
Matthias Weidlich	Imperial College London, UK
Jelena Zdravkovic	Stockholm University, Sweden

Additional Reviewers

Cristina Cabanillas
Claudio Di Ciccio
Amin Jalali
Jens Kolb
Andreas Lanz
Maria Leitner
Henry Roa Marin
Nicolas Mundbrod
Artem Polyvyanyy
Suriadi Suriadi
Wei Steve Wang
Stefan Zugal

EMMSAD 2014 Organization

Organizing Committee

John Krogstie	Norwegian Institute of Science and Technology, Norway
Khaled Gaaloul	Public Research Centre Henri Tudor, Luxembourg
Erik Proper	Public Research Centre Henri Tudor, Luxembourg, and Radboud University Nijmegen, The Netherlands

Advisory Committee

Terry Halpin	INTI International University, Malaysia
Keng Siau	Missouri University of Science and Technology, USA

Program Committee

Stephen Aier	University of St. Gallen, Switzerland
Antonia Albani	Delft University of Technology, The Netherlands
Raian Ali	Bournemouth University, UK
David Aveiro	Madeira University, Portugal
Annie Becker	Florida Institute of Technology, USA
Giuseppe Berio	University of Torino, Italy
Andy Carver	INTI International University, Malaysia
Sybren de Kinderen	University of Luxembourg
Olga De Troyer	Vrije Universiteit Brussel, Belgium
John Erickson	University of Nebraska-Omaha, USA
Peter Fettke	DFKI, Saarbrücken, Germany
Marcela Genero	University of Castilla-La Mancha, Spain
Wided Guedria	Public Research Centre Henri Tudor, Luxembourg
Sérgio Guerreiro	Universidade Lusófona de Humanidades e Tecnologias, Lisbon, Portugal
Brian Hendersson-Sellers	University of Technology, Sydney, Australia
Stijn Hoppenbrouwers	Radboud University Nijmegen, The Netherlands

Philip Huysmans	University of Antwerp, Belgium
Paul Johannesson	Stockholm University, Sweden
Bogdan Lent	University of Applied Sciences, Zurich, Switzerland
Pericles Loucopoulos	Loughborough University, UK
Florian Matthes	Technical University München, Germany
Raimundas Matulevičius	University of Tartu, Estonia
Graham McLeod	Promis Solutions, Switzerland
Jan Mendling	Humboldt-Universität zu Berlin, Germany
Haris Mouratidis	University of East London, UK
James Nelson	Southern Illinois University, Carbondale, USA
Antoni Olive	UPC, Barcelona, Spain
Andreas Opdahl	University of Bergen, Norway
Stefan Oppl	Johan Kepler University Linz, Austria
Sietse Overbeek	Delft University of Technology, The Netherlands
Hervé Panetto	University of Lorraine, France
Oscar Pastor	UPV, Valencia, Spain
Barbara Pernici	Politecnico di Milano, Italy
Anne Persson	University of Skövde, Sweden
Mario Piattini	University of Castilla-La Mancha, Spain
Georgios Plataniotis	Public Research Centre Henri Tudor, Luxembourg
Michael Prilla	Ruhr-Universität Bochum, Germany
Jolita Ralyté	University of Geneva, Switzerland
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
Guttorm Sindre	IDI, NTNU, Trondheim Norway
Monique Snoeck	KULeuven, Belgium
Janis Stirna	Royal Institute of Technology, Stockholm, Sweden
Dirk van der Linden	Public Research Centre Henri Tudor, Luxembourg
Yair Wand	University of British Columbia, Canada
Barbara Weber	University of Insbruck, Austria
Carson Woo	University of British Columbia, Canada
Jelena Zdravkovic	University of Stockholm, Sweden
Pär Ågerfalk	Uppsala University, Sweden

Additional Reviewers

Ralf Abraham
Stefan Bischoff
Tom Thaler
Timo Kucza
Klym Shumaiev
Bernhard Waltl
Mario Lezoche
Eduardo Portela
Marija Bjeković
Constantinos Giannoulis

Systems and Business Process Modeling: Are We Confusing the Map with the Territory? (Keynote)

Gil Regev[1,2]

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL), School of Computer and Communication Sciences, CH-1015 Lausanne, Switzerland
gil.regev@epfl.ch

[2] Itecor, Av. Paul Cérésole 24, cp 568, CH-1800 Vevey 1, Switzerland
g.regev@itecor.com

Abstract. Both EMMSAD and BPMDS are concerned with the modeling of people and machines, operating together in organizations. In the vast majority of modeling techniques people are represented only through the roles they play in that interaction. This has permitted very efficient modeling by abstracting from the particularities of different people and different machines, even making them interchangeable. We model human beings as if they were machines, through their externally visible behavior. Peoples' state of mind while interacting with a system or executing a business process is not externally visible, but nevertheless has a large influence on their behavior. Hesitations, changes of mind, debates, inconsistencies and other such frequent human behaviors are rarely if ever represented in our models. Most often our models are based on interviews where the interviewees are disconnected from their everyday work and therefore provide us with sweeping generalities. And yet we tend to think that our models truly represent reality. We then force people to work with our maps instead of making the effort to understand their territory, leading people to be alienated from the processes that are supposed to help them. If we want to take the human aspect seriously, we need to create modeling languages and methods that more accurately represent peoples' everyday work. Techniques such as scenarios, stories and personas are useful but are not sufficient. They must be based on direct observation in the field. We need to leave our research labs from time to time and meet people on their work premises in order for our models to better account for their territory.

Table of Contents

Business Process Modeling as a Human-Driven Process

Representing the Human Perspective of Business Processes

Supporting Humans in Business Processes

Variability-Enabling Process Models

Various Models for Various Process Perspectives

BPMDS in Practice

Conceptual Modeling

Requirements Modeling

Business Process Modeling

Goal and Language Action Modeling

Enterprise and Business Modeling

New Approaches (Short Papers)

The Modeling Mind: Behavior Patterns in Process Modeling*

Jakob Pinggera, Stefan Zugal, Marco Furtner, Pierre Sachse, Markus Martini, and Barbara Weber

University of Innsbruck, Austria
{jakob.pinggera,stefan.zugal,marco.furtner,pierre.sachse, markus.martini,barbara.weber}@uibk.ac.at

Abstract. To advance the understanding of factors influencing the quality of business process models, researchers have recently begun to investigate the way how humans create process models—the process of process modeling (PPM). In this idea paper, we subscribe to this human–centered perspective of process modeling and present future research directions pursued in the vision of Modeling Mind. In particular, we envision to extend existing research toward PPM behavior patterns (PBP) that emerge during the creation of process models. Thereby, we explore PBPs by triangulating several quantitative and qualitative research methods, i.e., integrating the modeler's interaction with the modeling environment, think aloud data, and eye movement data. Having established a set of PBPs, we turn toward investigating factors determining the occurrence of PBPs, taking into account modeler–specific and task–specific factors. These factors manifest as modeling expertise, self–regulation, and working memory capacity. In a next step, we seek to investigate the connection between PBPs and process model quality in terms of syntactic, semantic, and pragmatic quality. These findings, in turn, will be used for facilitating the development of customized modeling environments, supporting the process modeler in creating process models of high quality. Through this idea paper, we would like to invite researcher to join our research efforts to ultimately arrive at a comprehensive understanding of the PPM, leading to process models of higher quality.

Keywords: Process of Process Modeling, PPM Behavior Patterns, Business Process Modeling.

1 Introduction

For decades, conceptual models have been used to facilitate the development of information systems and to support practitioners when analyzing business domains [1]. Recently, particularly *business process models*, or process models for

* Modeling Mind is a collaborative research effort of the Institute of Computer Science and the Institute of Psychology at the University of Innsbruck. Modeling Mind is funded by Austrian Science Fund (FWF): P26609–N15.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 1–16, 2014.

short, have raised significant interest due to their critical role for the management of business processes [2]. For instance, business process models are used to support the development of process–aware information systems, inter–organizational workflows, service–oriented architectures and web services [3]. Additionally, the growing importance of business process management has influenced how conceptual modeling is taught, as business process management has been adopted in today's university curricula [4]. Considering the intense usage of business process modeling, the relevance of process models has become obvious. Yet, industrial process models display a wide range of problems [5], confirming that an in–depth understanding of factors influencing process model quality is required.

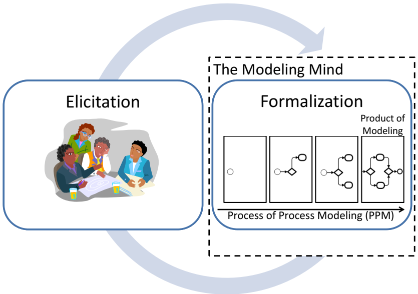


Fig. 1. Model Development Lifecycle

In response to the demand of process models of high quality, researchers recently have begun to take into account the processes involved in its creation. In general, as illustrated in Fig. 1, the *process model development lifecycle* involves several stakeholders, such as domain experts and system analysts, who drive the creation of the process model in *elicitation phases* and *formalization phases* [6]. In the *elicitation phase*, information from the domain is extracted by *domain experts* and used in the *formalization phase* by *system analysts* (*process modelers* in our context) for creating a formal process model [7]. Since requirements evolve over time, model development usually comprises *several iterations* of elicitation and formalization, resulting in an evolving process model.

For the creation of process models that satisfy stringent quality requirements, such as correctness, comprehensibility and maintainability, significant modeling skills are indispensable [8]. These quality demands, in turn, have sparked significant research regarding process model formalization, mostly focusing on the *product of a process modeling endeavor* (e.g., [9,10]). Recently, researchers have started to broaden their perspective from the *product*, i.e., the process model, toward the *process modeling act* (e.g., [11,12]). Thereby, research focuses on the *formalization*, in which a process modeler constructs a process model reflecting a given domain description—denoted as *process of process modeling (PPM)* (cf. [7]). So far, research on the PPM has focused on recording the modeler's interactions with the modeling environment. For instance, researchers have observed differences in the way process modelers create process model [12] and suggested the existence of *patterns of behavior* (denoted as PPM Behavior Patterns, or PBPs for short) [12,13]. For instance, it has been observed that some modelers start modeling immediately by adding elements to the model, while other modelers invest more time in understanding the modeling task before adding model elements [12]. Likewise, [13] observed that modelers working on specific, bounded parts of the model and finishing them before working on another part, tend to produce process models of higher quality [13]. Hence, we argue that the PPM provides an emerging, yet promising research direction for obtaining a better understanding of factors influencing the quality of process models.

1.1 Problem Statement

While prior research has advanced the understanding on how process models are created, several challenges remain. First, insights are solely based on analyzing the modeler's interactions with the modeling environment (e.g., adding, deleting, and moving nodes or edges), which allows only for a partial understanding of patterns of behavior [12]. Particularly, phases of inactivity with the modeling environment, e.g., when understanding the problem, can only be approximated. Thus, strategies modelers use during such phases (e.g., understanding the problem and mapping domain knowledge to modeling constructs [14]) cannot be investigated. Also, validation activities where modelers check whether the created process model complies with the requirements remain undiscovered. Therefore, challenge C_1 describes the need to identify patterns of behavior during the PPM.

Second, while the results reported in [12] give indications that the PPM is influenced by *modeler–specific characteristics* (inherent to a modeler irrespective of the modeling task) and *task–specific characteristics* (depending on the modeling task), the exact factors determining differences are hardly understood. Additionally, *contextual factors*, e.g., management approval (cf. [15]), might influence the creation of process models. In Modeling Mind, we focus on the formalization of process models based on a textual description, with the purpose of documenting the specification. Therefore, only contextual factors that might influence the formalization of process models are considered, e.g., modeling purpose [2]. *Modeler–specific characteristics* include process modeling expertise, domain knowledge, but also working memory capacity and the modeler's personality. *Task–specific characteristics*, on the contrary, are specific to the modeling task (e.g., the task's inherent complexity or the presentation of the task to the modeler) and determine the task's *cognitive load*. In case that a task's cognitive load exceeds the modeler's working memory capacity, errors are likely to occur [16]. Typically, the cognitive load caused by a modeling task is measured once at the end of the modeling task using self–rating. However, since challenges faced during modeling might influence the occurrence of PBPs and eventually lead to errors once working memory gets overloaded, knowing the evolution of cognitive load over time seems to be essential. Challenge C_2 therefore relates to the identification of factors determining the occurrence of PBPs. In the vision of Modeling Mind, we aim to investigate these challenges in the form of two research objectives:

Research Objective RO_1: Identify and Assess PBPs for Process Model Creation. Identifying PBPs considering different ways of analysis, i.e., model interactions, think aloud data, and eye movement data.

Research Objective RO_2: Understand Factors Determining the Occurrence of PBPs. Understanding how factors, including modeler–specific factors and task–specific factors, influence the occurrence of the identified PBPs.

Summarized, the goal of Modeling Mind is to obtain an in–depth understanding of PBPs. For this purpose, Modeling Mind utilizes different perspectives on the PPM, such as the modelers' interactions with the modeling environment, think aloud data, and eye movement analysis. In addition, Modeling Mind aims for understanding how these patterns relate to process model quality and for deriving a set of modeling styles bundling commonly co–occurring PBPs. Moreover, Modeling Mind seeks to understand factors determining the occurrence of PBPs covering modeler–specific factors and task–specific factors. By providing a theoretical model describing PBPs during the PPM and factors influencing modeling styles, Modeling Mind can facilitate the development of customized modeling environments, better supporting the individual modeler. Further, a thorough understanding of the PPM enables the development of effective teaching methods helping future students to become skilled in the craft of modeling.

The remainder of the paper is structured as follows. Section 2 describes the state–of–the–art relevant for RO_1 and RO_2. Section 3 describes how we envision to investigate RO_1 and RO_2, whereas the paper is concluded in Section 4.

2 Relation to State–of–the–Art

In this section, the state–of–the–art is discussed. First, the focus is put on PBPs presented in literature, forming a basis for RO_1. Then, a list of factors potentially determining the occurrence of PBPs is presented for RO_2.

2.1 Research on PPM Behavior Patterns

Existing work on the PPM has focused on analyzing interactions with the modeling environment. For this purpose, Cheetah Experimental Platform (CEP) [17] has been developed, providing a basic modeling editor recording the modeler's interactions with the modeling environment. By capturing the interactions, the creation of the process model can be *replayed* at any point in time.[1] Subsequently, we discuss PBPs described in literature to build a starting point for the investigations on RO_1. While the presented list is certainly not exhaustive, it constitutes a starting point for the investigations in Modeling Mind.

PBP_1 Planning. [12] describes differences regarding the time it takes modelers to start working on the process model. Some modelers start right away adding model elements (PBP_{1a}), while others invest more time in gaining an understanding of the modeling task (PBP_{1b}). To operationalize this aspect, [12] defines the measure of *initial comprehension duration*, capturing the time before

[1] A replay demo is available at `http://cheetahplatform.org`

modelers start adding content to the process model. It seems that differences in initial comprehension duration are modeler dependent, i.e., modelers that spend much time on an initial comprehension phase for one task do the same when working on a different task.

***PBP_2* Detours.** While some modelers create a process model in an efficient manner without detours (i.e., superfluous modeling interactions), others require several attempts (including the deletion and re–creation of content) [18]. To operationalize this aspect, [18] suggests a measure called *process deviations* that is calculated as the sum of all delete operations and those adding operations that deal with the re–creation of content [19].

***PBP_3* Layout Behavior.** In [12,20] we observed modelers who carelessly put nodes on the canvas and draw straight connecting edges, resulting in poor layout and a low number of layout operations (PBP_{3a}). Also, [12,20] reports on three strategies to come up with an appealing layout. The first strategy involves modelers that place elements at strategic places right from the beginning, making subsequent layout interactions unnecessary (PBP_{3b}). The second and third strategy involve modelers who carelessly put nodes on the canvas and perform layout operations later on, placing and arranging nodes and edges to achieve an appealing layout. Laying out is either performed continuously, leading to a high number of layout phases with a small number of layout operations each (PBP_{3c}) or toward the end of the modeling process all at once, resulting in a small number of layout phases with a high number of layout operations (PBP_{3d}). It could be shown that modelers with the desire to invest into good layout will persist in this intent [12], suggesting that layout preferences are also modeler–specific.

2.2 Research on Factors Determining the Occurrence of PBPs

The creation of a process model can be classified as a problem solving task [12,20], an area of vivid research for decades in cognitive psychology. Therefore, we turn to cognitive psychology as a starting point for understanding the factors determining PBPs. Subsequently, a list of factors potentially influencing the occurrence of PBPs as well as measurements are presented. The presented list cannot be considered complete, but should rather provide a starting point for future investigation. Only contextual factors impacting the formalization of process models are considered, e.g., modeling purpose [2].

Modeler–Specific Factors. In this section, a list of factors specific to the modeler are presented. Modeler–specific factors are inherent to the modeler, but not independent of the modeling task. For instance, existing domain knowledge is inherent to the modeler, but obviously only a factor if the domain of the modeling task matches or conflicts with the modeler's prior knowledge.

Process Modeling Expertise. Even though the influence of modeling expertise on the PPM has not been investigated, research has demonstrated that modeling expertise (i.e., process modeling experience and knowledge) positively influences process model comprehension (e.g., [21,22]). To assess modeling experience, often self-assessment questionnaires rating participants' knowledge on process modeling and their experience in process modeling are used [21]. Since the validity of self–assessment on theoretical modeling knowledge has been questioned [22], it is often complemented with a test on process modeling [23].

Domain Knowledge and Conceptual Modeling. Moreover, research has demonstrated the importance of domain knowledge for the understanding of conceptual models [24]. We speculate that similar effects also occurs for process modeling tasks. To assess domain knowledge, [24] suggests the usage of self–assessment questionnaires where participants rate their familiarity with the domain.

Working Memory and Complex Problem Solving Tasks. Working memory (WM) represents a construct that maintains and manipulates a limited amount of information for goal directed behavior [25]. WM defines a main construct in human information processing and is a central component for an understanding of inter–individual differences in process modeling. More specifically, during the PPM, WM is responsible for the representation and integration of information for an iterative construction of a mental, and in the following physical, process model. The capacity of WM (WMC) can be measured via complex span tasks [26]. There is strong empirical evidence that WMC predicts performance in tasks like, e.g., language comprehension [27], reasoning [28], and the integration of preexisting domain knowledge [29]—fundamental cognitive abilities relevant for the PPM.

Self–Regulation and Complex Problem Solving Tasks. Self regulation consists of two basic modes: *locomotion* and *assessment* [30]. Locomotion is characterized by instantaneous, straight, and action oriented behavior (*"Just doing it"*). Assessment refers to the critical, strategical cognitive planning and evaluation of a given situation (e.g., goals, given means to reach them, and alternatives, *"Doing the right things"*). A person with high locomotion and low assessment acts like a *"headless chicken"* (trial and error). A person with low locomotion and high assessment will put only little into action. Therefore, for high achievement performance, balancing locomotion and assessment is necessary [31]. Self–regulation can be measured with the Locomotion–Assessment–Questionnaire [30].

Task–Specific Characteristics. Creating a process model from a given process description is not only influenced by modeler–specific characteristics, but also by characteristics of the modeling task. This influence is described by Cognitive Load Theory [16] as *cognitive load* on the person solving the task. The cognitive load of a task is determined by its *intrinsic load*, i.e., the inherent difficulty associated with a task and its *extraneous load*, i.e., the load generated by the task's representation [32]. In our context, *intrinsic load* is determined by the process to be modeled. It can be characterized by the size and complexity of the model structure and constructs [33]. In contrast, *extraneous load* concerns, for example, the presentation of the task to the modeler [10]. Additionally, the properties of the notational systems, i.e., the modeling notation and the modeling environment, affect the difficulty of the modeling task, e.g., due to notational deficiencies [34]. Moreover, related work has demonstrated the impact of activities' labeling style [35], secondary notation (i.e., layout) [21], and the influence of modularity [36,37] on understandability. While the impact of various task extraneous characteristics on model understanding has been widely addressed, it is less understood how these factors impact the creation of process models. In general, research has shown that a high cognitive load increases the probability of errors, especially when the WMC is exceeded [16]. Cognitive load is typically operationalized as *mental effort* [32] and various measurement techniques can be applied, such as the measurement of the diameter of the eyes' pupil (*pupillometry*), heart–rate variability, and rating scales [32]. Especially pupillometric data and rating scales (i.e., self–rating mental effort) have been shown to reliably measure mental effort and are widely adopted [38].

3 Vision of Modeling Mind

This section describes how we *envision* the detection of PBPs within PPM instances[2]. First, techniques for detecting PBPs are presented in Section 3.1. Then, possible PBPs are presented in Section 3.2, addressing RO_1. Finally, the data analysis procedure for identifying the influencing factors for PBPs are described in Section 3.3, addressing RO_2. Prior research on PBPs has focused on analyzing the interactions with the modeling environment. While several PBPs could be observed, this rather narrow focus limits the options for analyzing the PPM. For instance, the distinction between understanding the problem and mapping the problem to modeling constructs [14] cannot be detected. Therefore, we intend to complement the analysis of the modeler's interactions with the modeling environment with *think aloud data* and *eye movement data*. Additionally, feedback from modelers is collected. This way, we hope to triangulate toward a more comprehensive understanding of the PPM.

3.1 Widening the Perspective of the PPM

In Modeling Mind we seek to widen the perspective on the PPM by taking additional methods of investigation into account. First, we adopt think aloud, asking modelers to verbalize their thoughts during modeling [39]. This allows us to draw inferences on how modelers arrive at their conclusion. For instance, using think aloud, the distinction between understanding the problem and mapping the problem to the modeling constructs can be investigated. Second, apparently modelers rely on visual perception for reading the task description and designing

[2] A PPM instance constitutes one specific instantiation of the PPM.

the process model when creating a formal process model. To investigate visual perception, we plan to adopt eye movement analysis, allowing to further investigate internal mental processes involved in modeling. For instance, it is known that high–resolution visual information input, which only occurs during fixations [40] (the modeler fixates the area of interest with the fovea), is necessary for reading text or identifying model elements. This, in turn allows for identifying specific areas of interest the modeler is focusing attention on, e.g., the task description, features of the modeling environment, or modeling constructs. Within these areas, eye fixations can be measured to identify the parts of the modeling environment modelers focus on, potentially pointing to challenging situations.

Visualization of Model Interactions, Eye Movement Data, and Think Aloud Data. To support the identification of PBPs, we develop visualizations enabling an integrated analysis of model interactions, eye movement data, and think aloud data. Subsequently, several potential visualizations are proposed.

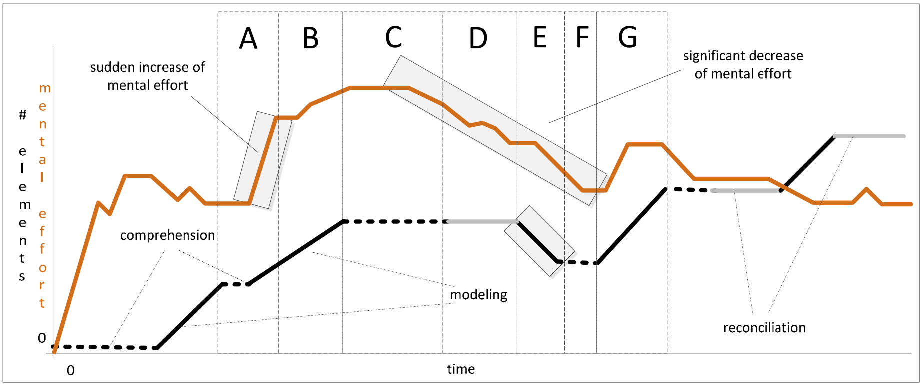


Fig. 2. MPD Including Mental Effort Visualization

Modeling Phase Diagrams with Mental Effort. Modeling Phase Diagrams (MPDs) were proposed to visualize the interactions with the modeling environment to gain an overview of PPM instances [20]. For this, different activities during the formalization of process models are considered. Activities indicating modeling, such as adding content by creating nodes and edges, are mapped to *modeling* phases. Similarly, activities indicating clean–up, such as laying out the process model, are mapped to *reconciliation* phases. Finally, phases of inactivity usually indicate cognitive activities like understanding the problem, and hence are mapped to *comprehension* [20]. MPDs can be extended for displaying the modeler's mental effort, which was measured using pupillometry (cf. Fig. 2). Such a visualization allows for a quick overview of challenges faced during the PPM. For example, the PPM instance in Fig. 2 shows a sudden increase of mental effort in phase A. When looking at the corresponding visualization of modeling phases, it can be observed that this increase occurs in a comprehension phase. Fig. 2 further shows that mental effort remains high throughout the subsequent modeling phase B. The MPD suggests that the high mental effort might have caused a modeling error and the deletion of modeling elements in phase E and respective corrections during phase G (i.e., subsequent increase of number of model elements). It can be further observed that the mental effort started to decrease in phase C before the modeler started with the corrections and continued to decrease in phase D during which the modeler performed reconciliation interactions. This visualization could be complemented with an overview of significant changes of mental effort within a predefined timeframe (or timeframes with particularly high mental effort). Further, the visualization allows to jump to the corresponding part of the PPM instance using the replay feature of CEP.

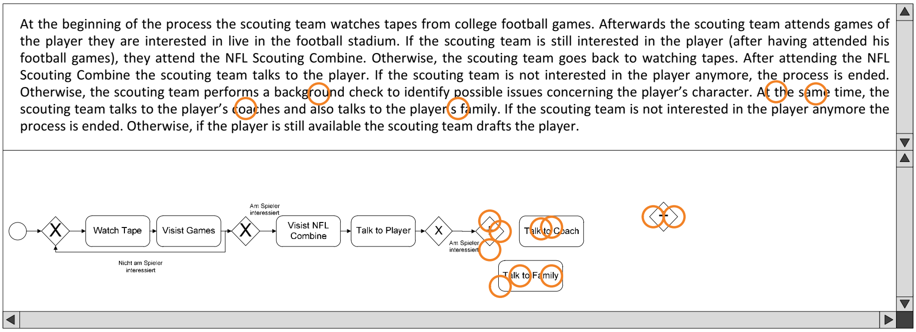


Fig. 3. Visualization of Fixations

Visualizing Fixations and Model Elements. An increase in mental effort only indicates that the modeler perceived the modeling task more difficult in the current situation. Still, it does not allow for identifying the reasons for the problem. Therefore, mental effort analysis can be complemented with a visualization of the modeler's fixations. For this, an overlay for the modeling editor is designed, displaying fixations within, e.g., the last 10 seconds prior to the current position within the replay of the PPM instance. Fig. 3 illustrates the visualization, assuming that an increase in mental effort was identified using the visualization in Fig. 2. Then, CEP can be utilized for navigating to the position within the PPM instance with increased mental effort. The fixations within the last seconds reveal that few fixations were on the textual description, but several fixations were on the modeling canvas. This might indicate that the modeler had problems with the modeling constructs required for translating the information extracted from the textual description to the formal process model. Since the position of modeling elements as well as the textual description can be computed in an automated way, fixations can be automatically mapped. This, in turn, could be used to analyze whether modelers solely focus on the process model in reconciliation phases or whether they access the textual description to perform reconciliation.

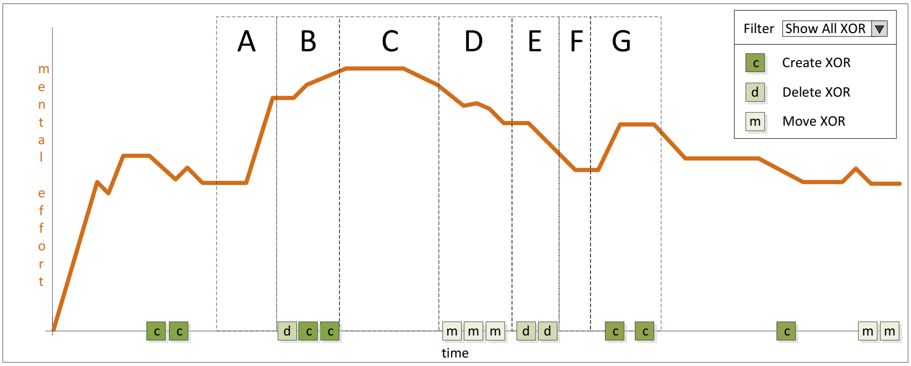


Fig. 4. Interactions with Mental Effort

Visualizing Mental Effort for Slices of the PPM. Another visualization could—instead of dividing the PPM into different phases—focus on the the type of interaction with the modeling environment and slice PPM instances according to the modeling elements, e.g., XOR gateways. As illustrated in Fig. 4, insights can be gained into relations between mental effort and specific types of model interaction. Similar to Fig. 2, the visualization shows mental effort in a continuous manner. Further, interactions with the modeling environment passing a customizable filter, i.e., all interactions involving XOR gateways in Fig. 4, are displayed on a timeline. For instance, the example in Fig. 4 shows that two XOR gateways are created prior to the increase in mental effort in phase A. After a steep increase in mental effort in phase A, mental effort remains high in phase B. Further, in phase B a XOR gateway is removed and two new XOR gateways are created. This could indicate that the increase in mental effort is related to the XOR gateways. Next, several XOR gateways are moved in phase D, while mental effort decreases. In phase E, two XOR gateways are removed, while in phase G new XOR gateways are added to the process model causing an increase in mental effort. No major increases in mental effort related to the XOR gateways toward the end of the PPM instance can be observed.

Again, the visualization could be extended with measures linking the mental effort with modeling elements. For instance, timeframes surrounding the creation of the model element could be used for computing the average mental effort required for creating the model element. The selection of appropriate timeframes, however, needs to investigated first.

Visualizing Think Aloud Protocols. Think aloud protocols can be visualized by utilizing the synchronized timestamps for positioning the verbal utterances within the MPD (cf. Fig. 5). By clicking on the visualization, the corresponding verbal utterance is displayed. Further, the verbal utterances can be filtered according to a previously manually constructed the coding.

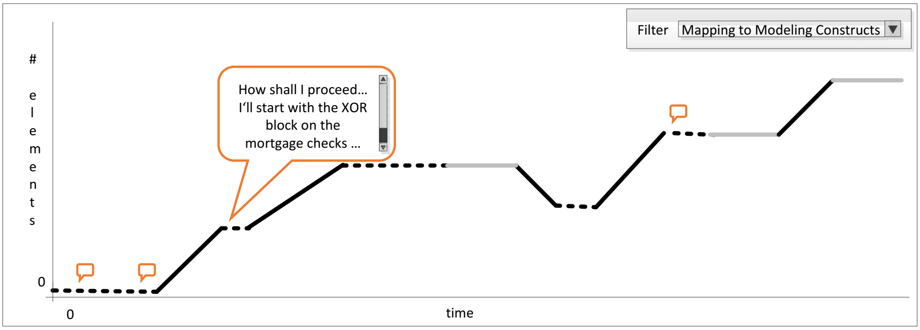


Fig. 5. Visualization of Verbal Utterances

3.2 RO_1: Identify and Assess PBPs for Process Model Creation

So far, we described how Modeling Mind intends to develop new visualizations of the PPM. Next, we describe how we approach the identification PBPs.

Identifying PBPs. As starting point for identifying PBPs, the initial set outlined in Section 2.1 is used. We plan to extend this set by analyzing the modeler's interactions with the modeling environment and by considering think aloud data and data regarding the modeler's eye movements. The results of the data exploration stage are then used to propose additional PBPs resulting in an extended catalog of PBPs. We are envisioning patterns similar to the following:

Candidate Pattern 1 (CP_1) Goal Orientation. Considering the visualization of think aloud protocols, one might be able to detect differences between modelers in terms of the type of thoughts uttered, e.g., modelers that are highly goal–oriented and conduct a strategic planning phase before starting with the modeling (CP_{1a}), in contrast to modelers that show less goal–oriented behavior and immediately start with modeling (CP_{1b}).

Candidate Pattern 2 (CP_2) Causes of Confusion. Similarly, considering the visualization in Fig. 3, one might see different PBPs depending on whether challenges stem from difficulties in creating an internal representation of the domain to be modeled (i.e., high average fixation duration on the textual description; CP_{2a}) or from difficulties in mapping the internal representation to modeling elements (i.e., high average fixation duration on the process model; CP_{2b}).

Assessing PBPs. In a next step, we investigate in how far the identified patterns are related to the quality of the resulting process models. This requires operationalizing patterns and statistically analyzing the relation to quality characteristics, e.g., using correlation analysis. As quality measures we consider *syntactic errors* (e.g., violations of the soundness property) and *semantic errors* referring to the validity (i.e., statements in the model are correct and related to the problem) and completeness (i.e., the model contains all relevant and correct statements to solve this problem) of the model. *Pragmatic quality* is typically related to the understandability of the model [41]. This includes, e.g., evaluating the process model's secondary notation. For assessing syntactic errors, we rely on existing automated techniques (e.g., [42]). For assessing semantic quality, we follow a semi–automated approach, since no fully automated solution exists [14]. We plan to utilize the ICoP framework [43] in combination with features provided by jBPT [44]. Similarly, a semi–automated approach is utilized for assessing pragmatic quality. Complementary to these approaches, we use expert assessment (e.g., in form of an iterative consensus building process [45]).

From PBPs to Modeling Styles. Finally, combinations of PBPs are be combined to define modeling styles. To discover clusters of co–occurring patterns (i.e., modeling styles) we plan to use quantitative methods like correlation analysis. The identified clusters can then serve as the basis for implementing personalized modeling environments or tailored teaching materials.

3.3 RO_2: Understand Factors Determining the Occurrence of PBPs

To understand factors determining the occurrence of PBPs, we follow a two–step procedure. First, data on modeler–specific factors (cf. Section 2.2) is recorded in addition to the modeler's interactions, think aloud data, and eye movement data. Modeling tasks are planned to cover different complexity; other task–specific factors presumably affecting the occurrence of PBPs (task representation, modeling notation, and tool support) are controlled. Second, we investigate factors determining pattern occurrence. Subsequently, the expected impact of factors is described for a subset of PBPs. Respective expectations serve as starting point to be refined and extended for the newly identified PBPs. To differentiate between modeler–specific characteristics and task–specific characteristics we conduct between–modeler comparisons, between–task comparisons and within–task comparisons. Between–modeler comparisons focus on comparing modelers with different characteristics, i.e., WMC, self–regulation, domain knowledge, and process modeling expertise, who individually work on the same modeling task. Between–task comparisons, in turn, compare PBPs of the same modeler working on several tasks with different task–characteristics. For within–task comparisons, we extract slices of PPM instances and compare these slices regarding PBPs. This way, different aspects of model creation within a PPM instance can be compared. Subsequently, we present *examples* for factors that might influence the patterns described in Section 2.1.

PBP_1 Planning. PBP_1 is related to initial comprehension duration and seems to depend on modeler–specific characteristics. In particular, locomotion ("Just doing it") and assessment ("Doing the right things") might play a role. Modelers scoring high on assessment might have longer initial comprehension phases

(PBP_{1a}) when compared to modelers scoring high on locomotion (PBP_{1b}). In addition, self–regulation might play a role, i.e., modelers scoring high on self–regulation presumably have long initial comprehension phases containing strategic planning activities (PBP_{1a}). Additionally, WMC might play a moderating role since building a mental model of the task requires a high WMC (PBP_{1a}). Moreover, modelers with high modeling expertise might use pattern PBP_{1a} more frequently than less experienced modelers, since WM is used more efficiently.

PBP_2 Detours. Regarding PBP_2, preliminary insights suggest that a modeler's WMC has a measurable impact on the number of detours taken during modeling [19]. Modelers with higher WMC are able to create the solution more efficiently, i.e., take less detours. Similar effects might occur for locomotion and assessment. For instance, high locomotion in combination with low assessment is expected to result in the highest number of detours because of modelers running into dead ends, since strategic behavior patterns are missing.

Summarized, in Modeling Mind we seek for identifying patterns in the modeler's behavior and plan to investigate the underlying factors determining the occurrence of PBPs, leading to a comprehensive understanding of the PPM.

4 Summary

This idea paper presents future research directions pursued by the Modeling Mind, intended to develop a comprehensive set of PBPs, which describe how modelers interact with the modeling environment. PBPs are explored by triangulating quantitative and qualitative research methods. For instance, we intend to analyze the modeler's interactions with the modeling environment, think aloud data, and eye movement data. Having established a set of PBPs, the research direction turns toward investigating factors determining the occurrence of PBPs. Presumably, influencing factors can be partitioned into modeler–specific factors and task–specific factors. For investigating modeler–specific factors, we intend to use established questionnaires from cognitive psychology to measure, e.g., modeling expertise, and working memory capacity. Further, the connection between process model quality in terms of syntactic, semantic, and pragmatic quality is investigated. This way, we intend to develop a model describing how modelers create process models. In this context, ignoring elicitation by assuming the existence of a complete domain description cannot be considered representative for modeling in practice. Still, the formalization of process models constitutes a sub–part of professional modeling activities. However, generalizations regarding the professional modeling community need to made with care.

By providing a theoretical model describing PBPs during the PPM and factors influencing modeling styles, we hope to facilitate the development of customized modeling environments. For instance, sketchpads in combination with a touch devise could be useful for understanding the problem, while tools for laying out process models might support the improvement of the process model's understandability. Further, a thorough understanding of the PPM enables the development of effective teaching methods helping future students to become skilled in the craft of modeling. For instance, if it turns out that specific aspects of modeling, e.g., the creation of loops, co–occur with increased mental effort, this might be an aspect teachers should specifically focus on during their instructions. This paper focuses on business process modeling, but similar investigations for other areas of conceptual modeling might be envisioned. Through this idea paper, we would like to invite researcher to join our research efforts to ultimately arrive at a comprehensive understanding of the creation of conceptual models, leading to models of higher quality.

References

1. 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 Association for Information Systems 9, 748–802 (2008)
2. 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)
3. Zugal, S.: Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models. PhD thesis, University of Innsbruck, Department of Computer Science (2013)
4. Recker, J.C., Rosemann, M.: Teaching business process modelling: Experiences and recommendations. Communications of the Association for Information Systems 25, 379–394 (2009)
5. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry 62, 467–486 (2011)
6. Hoppenbrouwers, S.J., Proper, E.H., van der Weide, T.P.: Formal Modelling as a Grounded Conversation. In: Proc. LAP 2005, pp. 139–155 (2005)
7. Hoppenbrouwers, S., Proper, H., Weide, T.: A Fundamental View on the Process of Conceptual Modeling. In: Proc. ER 2005, pp. 128–143 (2005)
8. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction and Guidelines for Correctness. Springer (2008)
9. Recker, J.C., Rosemann, M., Green, P., Indulska, M.: Do ontological deficiencies in modeling grammars matter? MIS Quarterly 35, 57–79 (2011)
10. Pinggera, J., Zugal, S., Weber, B., Fahland, D., Weidlich, M., Mendling, J., Reijers, H.: How the Structuring of Domain Knowledge Can Help Casual Process Modelers. In: Proc. ER 2010, pp. 231–237 (2010)
11. Pinggera, J., Furtner, M., Martini, M., Sachse, P., Reiter, K., Zugal, S., Weber, B.: Investigating the Process of Process Modeling with Eye Movement Analysis. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 438–450. Springer, Heidelberg (2013)
12. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H., Mendling, J.: Styles in business process modeling: An exploration and a model. Software & Systems Modeling (2013), doi:10.1007/s10270-013-0349-1
13. Claes, J., et al.: Tying Process Model Quality to the Modeling Process: The Impact of Structuring, Movement, and Speed. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 33–48. Springer, Heidelberg (2012)

14. Soffer, P., Kaner, M., Wand, Y.: Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 357–369. Springer, Heidelberg (2012)
15. Wasana, S., Rosemann, M., Doebeli, G.: A process modelling success model: Insights from a case study. In: Proc. ECIS 2003, pp. 1–11 (2003)
16. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12, 257–285 (1988)
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. Weber, B., Pinggera, J., Torres, V., Reichert, M.: Change Patterns in Use: A Critical Evaluation. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 261–276. Springer, Heidelberg (2013)
19. Sachse, P., Martini, M., Pinggera, J., Weber, B., Reiter, K., Furtner, M.: Das Arbeitsgedächtnis als Nadelöhr des Denkens. In: Psychologie Menschlichen Handelns: Wissen & Denken—Wollen & Tun. Pabst (2013)
20. Pinggera, J., Zugal, S., Weidlich, M., Fahland, D., Weber, B., Mendling, J., Reijers, H.A.: Tracing the Process of Process Modeling with Modeling Phase Diagrams. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 370–382. Springer, Heidelberg (2012)
21. 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)
22. 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)
23. Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension—Findings from a series of experiments. DSS 53, 195–206 (2012)
24. Khatri, V., Vessey, I.: Information Use in Solving a Well-Structured IS Problem: The Roles of IS and Application Domain Knowledge. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 46–58. Springer, Heidelberg (2010)
25. Baddeley, A.: Working Memory: Theories, Models, and Controversies. Annu. Rev. Psychol. 63, 1–29 (2012)
26. Lewandowsky, S., Oberauer, K., Yang, L.X., Ecker, U.K.: A working memory test battery for MATLAB. Behavior Research Methods 42, 571–585 (2010)
27. Just, M.A., Carpenter, P.A.: A capacity theory of comprehension: Individual differences in working memory. Psychological Review 99, 122–149 (1992)
28. Oberauer, K., Süß, H.M., Wilhelm, O., Wittman, W.W.: The multiple faces of working memory: Storage, processing, supervision, and coordination. Intelligence 31, 167–193 (2003)
29. Hambrick, D.Z., Engle, R.W.: Effects of domain knowledge, working memory capacity, and age on cognitive performance: An investigation of the knowledge-is-power hypothesis. Cognitive Psychology 44, 339–387 (2002)
30. Kruglanski, A.W., Thompson, E.P., Higgins, E.T., Atash, M., Pierro, A., Shah, J.Y., Spiegel, S.: To "do the right thing" or to "just do it": locomotion and assessment as distinct self-regulatory imperatives. Journal of Personality and Social Psychology 79, 793–815 (2000)

31. Higgins, E.T., Kruglanski, A.W., Pierro, A.: Regulatory mode: Locomotion and assessment as distinct orientations. Advances in Experimental Social Psychology 35, 293–344 (2003)
32. Paas, F., Renkl, A., Sweller, J.: Cognitive Load Theory and Instructional Design: Recent Developments. Educational Psychologist 38, 1–4 (2003)
33. Figl, K., Laue, R.: Cognitive Complexity in Business Process Modeling. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 452–466. Springer, Heidelberg (2011)
34. Figl, K., Recker, J., Mendling, J.: A study on the effects of routing symbol design on process model comprehension. Decision Support Systems 54, 1104–1118 (2013)
35. Mendling, J., Reijers, H., Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. Information Systems 35, 467–482 (2010)
36. Zugal, S., Pinggera, J., Mendling, J., Reijers, H., Weber, B.: Assessing the Impact of Hierarchy on Model Understandability—A Cognitive Perspective. In: Proc. EESSMod 2011, pp. 123–133 (2011)
37. Zugal, S., Soffer, P., Haisjackl, C., Pinggera, J., Reichert, M., Weber, B.: Investigating expressiveness and understandability of hierarchy in declarative business process models. SoSym, 1–23 (2013), doi:10.1007/s10270-013-0356-2
38. Zugal, S., Pinggera, J., Reijers, H., Reichert, M., Weber, B.: Making the Case for Measuring Mental Effort. In: Proc. EESSMod 2012, pp. 37–42 (2012)
39. Ericsson, K.A., Simon, H.A.: Protocol analysis: Verbal reports as data. MIT Press (1993)
40. Posner, M.I.: Attention in cognitive neuroscience. In: The Cognitive Neurosciences, pp. 615–624. MIT Press (1995)
41. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: A revised quality framework. EJIS 15, 91–102 (2006)
42. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
43. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP Framework: Identification of correspondences between process models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)
44. Polyvyanyy, A., Weidlich, M.: Towards a compendium of process technologies: The jbpt library for process model analysis. In: Proc. CAiSE Forum 2013, pp. 106–113 (2013)
45. Recker, J., Safrudin, N., Rosemann, M.: How Novices Design Business Processes. Information Systems 37, 557–573 (2012)

How Advanced Change Patterns Impact the Process of Process Modeling*

Barbara Weber[1], Sarah Zeitelhofer[1], Jakob Pinggera[1], Victoria Torres[2], and Manfred Reichert[3]

[1] University of Innsbruck, Austria
firstname.lastname@uibk.ac.at, sarah.zeitlhofer@student.uibk.ac.at
[2] Universitat Politècnica de València, Spain
vtorres@pros.upv.es
[3] University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. Process model quality has been an area of considerable research efforts. In this context, correctness-by-construction as enabled by change patterns provides promising perspectives. While the process of process modeling (PPM) based on change primitives has been thoroughly investigated, only little is known about the PPM based on change patterns. In particular, it is unclear what set of change patterns should be provided and how the available change pattern set impacts the PPM. To obtain a better understanding of the latter as well as the (subjective) perceptions of process modelers, the arising challenges, and the pros and cons of different change pattern sets we conduct a controlled experiment. Our results indicate that process modelers face similar challenges irrespective of the used change pattern set (core pattern set versus extended pattern set, which adds two advanced change patterns to the core patterns set). An extended change pattern set, however, is perceived as more difficult to use, yielding a higher mental effort. Moreover, our results indicate that more advanced patterns were only used to a limited extent and frequently applied incorrectly, thus, lowering the potential benefits of an extended pattern set.

Keywords: Process Model Quality, Process of Process Modeling, Change Patterns, Controlled Experiment, Problem Solving.

1 Introduction

Due to the important role they play for process–aware information systems, process models have become increasingly important for many years [1]. In this context, it was shown that process model understandability has a measurable impact on whether or not a process modeling initiative is successful [2]. Still, process models exhibit a wide range of quality problems, which not only hamper comprehensibility but also affect maintainability [3,4]. For example, [3] reports on error rates between 10% and 20% in collections of industrial process models.

* This research is supported by Austrian Science Fund (FWF): P23699-N23.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 17–32, 2014.

To improve process model quality, change patterns appear promising. They combine change primitives, e.g., to add nodes or edges, to high-level change operations [4]. In particular, change patterns enable correctness-by-construction [5] by providing only those change patterns to the modeler, which ensure that process models remain sound after applying model transformations.

Recently, the creation of process models based on change primitives has received considerable attention resulting in research on the *process of process modeling (PPM)* [6,7,8]. This research focuses on the *formalization* phase of process model creation, i.e., the interactions of the process modeler with the modeling environment. The PPM utilizing change patterns, in turn, is still hardly understood. In previous work we presented an exploratory study to investigate re-occurring challenges when using change patterns for process modeling [9]. The study revealed that process modelers did not face major problems when using change patterns for constructing simple process fragments. When being confronted with more complex process fragments, however, difficulties increased observably. Building respective structures efficiently (i.e., without *detours* in the PPM) requires process modelers to look ahead, since patterns cannot be always combined arbitrarily. This need for looking ahead is a fundamental difference compared to process model creation based on change primitives and was perceived as both challenging and restrictive by subjects. Further, [9] emphasizes that the basic set of change patterns, which allows creating control flow structures like sequence, exclusive branchings, parallel branchings, and loops, is not sufficient for efficient model creation. In particular, the study observed that patterns for moving process fragments might help to resolve detours efficiently.

On one hand, an extended set of change patterns (including move patterns) offers more flexibility. On the other, it increases complexity, especially when mapping the mental model of the process to be created to the available pattern set. As a result, the extended change pattern set might make the modeling environment more difficult to use. This raises the question whether the expected benefits of an extended pattern set can be materialized in a practical setting. To obtain an in-depth understanding of the impact an extended pattern set has on the PPM, we implement a modeling editor offering two different change pattern sets based on Cheetah Experimental Platform (CEP) [10]. Using this editor, we conduct a controlled experiment with 42 process modelers. Our results indicate that an extended pattern set yields higher mental effort for modelers and is perceived as more difficult to use. At the same time, the expected benefits in terms of increased problem solving efficiency did not materialize, suggesting to focus on a core pattern set. The results provide a contribution toward a better understanding on how tool features (like change patterns) impact the PPM, but also give advice on how effective tool support should be designed.

Sect. 2 introduces backgrounds. Sect. 3 describes the controlled experiment. Sect. 4 presents the subjective perception of change pattern use. Sect. 5 deals with the impact of change patterns on problem solving efficiency and Sect. 6 details on the actual and potential use of patterns. Limitations are presented in Sect. 7. Related work is presented in Sect. 8. Sect. 9 concludes the paper.

2 Process Model Creation Based on Change Patterns

Most environments for process model creation are based on change primitives, e.g., `add/delete activity` or `add/delete edge`. Process model adaptations (i.e., transformation of a model S into model S') may require the joint application of multiple change primitives. Imagine process model S_1 in Fig. 1 without the colored fragment. To transform this model into S1 (including the colored fragment) 19 change primitives are needed: deleting the edge between activity `D` and the parallel gateway, adding `D`,`E`, and `F` to the process model, adding the conditional branch around `C` (including transition conditions), and adding the edges connecting the newly added elements with the process model. When applying change primitive, soundness of the resulting process model cannot be guaranteed and must be explicitly checked after every model transformation. In turn, change patterns imply a different way of interacting with the modeling environment. Instead of applying a set of change primitives, high-level change operations are used to realize the desired model transformation. Examples of change patterns include the insertion of process fragments, their embedding in conditional branches or loops, or the updating of transition conditions. A catalog of change patterns can be found in [4], while their semantics of these patterns are described in [11]. To conduct the described transformation with change patterns (i.e., obtain S_1 from a model where the colored fragment is missing), 6 pattern applications are needed (i.e., serial insert of activity `E`, parallel insert of activity `F`, serial insert of activity `C`, embed activity `C` in conditional branch, and two updates of conditions). As opposed to change primitives, change pattern implementations typically guarantee model correctness after each transformation [5] by associating pre-/post-conditions with high-level change operations. In process modeling environments supporting the correctness–by–construction principle (e.g., [12]), usually process modelers only have those change patterns available for use that allow transforming a sound process model into another sound one. For this purpose, structural restrictions on process models (e.g., block structuredness) are imposed. This paper investigates the impact of two different change pattern sets on the PPM.

3 Experiment

This section describes research questions and the design of the experiment.

Research Questions. Our goal is to obtain an in-depth PPM understanding when using change patterns. More specifically, we want to understand how modelers experience their interaction with the modeling environment depending on the available change pattern set.

> RQ1: What is the impact of the change pattern set available to process modelers on their subjective perception during model creation?

In addition to the subjective perception of modelers, we are interested in the challenges faced by process modelers during the PPM depending on the used change pattern set. Respective challenges can result in modeling errors that persist in the final model, but also detours on the way to a complete process model, negatively affecting problem solving efficiency.

> RQ2: What is the impact of the change pattern set available to process modelers on the challenges faced during model creation?

Finally, we want to understand how well the additional patterns of the extended pattern set was adopted (i.e., in their actual use) as well as the potential benefits that could have been achieved through proper pattern usage.

> RQ3: What was the actual use of the additional change patterns compared to the potential of using those patterns?

Factors and Factor Level. The experiment considers a single factor, i.e, the pattern set used to conduct the modeling task with factor levels: *core* and *extended*. The core pattern set comprises a minimum change pattern set (see [4] for the full pattern set) that allows modelers to create basic control-flow structures (i.e., sequences, parallel, conditional branchings, and loops): patterns AP1 (Insert Process Fragment), AP2 (Delete Process Fragment), AP8 (Embed Fragment in Loop), AP10 (Embed Process Fragment in Conditional Branch), and AP13 (Update Condition). Concerning pattern AP1, two variants were provided: Serial and Parallel Insert. In addition, process modelers could rename activities. In turn, the extended pattern set comprises all patterns included in the core pattern set plus an advanced pattern for moving process fragments (AP3). To be able to trace back the impact to single change patterns, we intentionally decided to only add one additional pattern from which we expect a considerable impact on problem solving efficiency to the extended pattern set. Similar to AP1, two variants are provided: Serial and Parallel Move. While the core pattern set is complete in the sense that all control-flow structures can be created, it does not allow for arbitrary model transformations. In particular, in [9] we observed that detours could have been addressed more efficiently with an extended pattern set. In particular, we observed that patterns for moving process fragments would have helped with many of the detours. Frequently, process modelers had to undo or delete considerable parts of the model, which could have been resolved with the application of a single move pattern. Consider, for example, the two models in Fig. 1. When transforming S_1 to S_2 without move patterns, the modeler must perform a detour of 7 steps to delete the colored parallel branch and to re-insert it after activity B (cf. problem solving path $P_{1,2}$). On the contrary, using move patterns, transforming S_1 into S_2 just requires the application of one change pattern, i.e., Serial Move, saving a total of 6 pattern applications.
Modeling Tasks. The modeling task is a slight adaption of the task used in [9] and describes a process of the "Task Force Earthquakes" of the German Research Center for Geosciences [13] (cf. Fig. 2—labels are abstracted for readability). The task comprises 15 activities; all main control–flow structures like sequence,

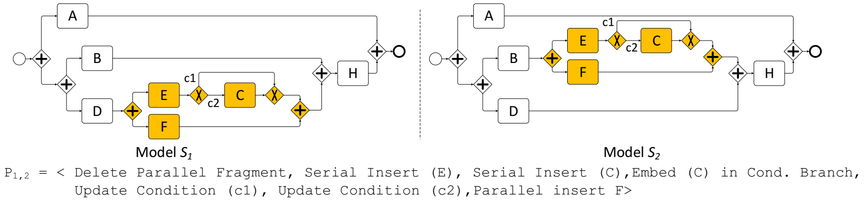


Fig. 1. Detour to go from S_1 to S_2 when no Move Patterns are available

parallel and conditional branchings, and loops are present. The model has a nesting depth of 4. Subjects received an informal requirements description as well as the solution of the modeling task (i.e., a process model). Their task was to re–model the process using change patterns. To model the process a minimum number of 28 change patterns are required with both the core and the extended change pattern set. Since subjects had the correct solution available, the challenge lies in determining the patterns for re-constructing the model and in combining the available patterns effectively.

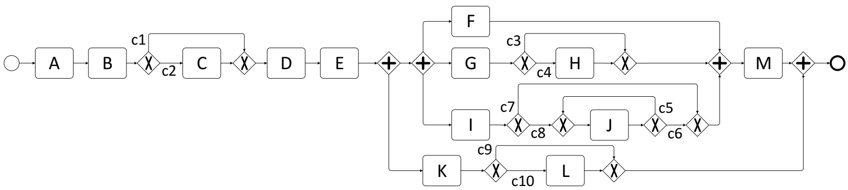


Fig. 2. Solution Model S_S

Subjects. Novices and experts differ in their problem solving strategies. Considering that industrial process modelers are often not expert modelers, but rather casual modelers with a basic amount of training [14], subjects participating in our experiment are not required to be experts. In previous research with software engineering students it has been shown that students can provide an adequate model for the professional population [15,16,17]. Thus, we relied on students (instead of professionals) in our experiment. To avoid difficulties due to unfamiliarity with the tool, rather than the modeling task, we require some prior experience with process modeling as well as change patterns. To ensure that the subjects are sufficiently literate in change pattern usage, subjects are provided with theoretical backgrounds. Further the subjects obtain hands-on experience in the creation of process models using change patterns in terms of a familiarization task.

Experimental Setup. The experiment consists of four phases. (1) collecting demographic data, (2) familiarization with the change patterns editor, and (3) performing a modeling task. Subjects were divided into two groups. While *Group A* receives the core pattern set, *Group B* conducts the same task based on the extended pattern set. During modeling, all interactions with the modeling environment are recorded by CEP [10]. This allows us to replay the creation of the process model step-by-step [10], addressing RQ2 and RQ3. After completing the modeling tasks, (4) *mental effort* as well as *Perceived Ease of Use (PEU)* and *Perceived Usefulness (PU)* of the *Technology Acceptance Model* [18] are assessed, addressing RQ1.

Experimental Execution. Prior to the experiment a pilot was conducted to ensure usability of the tool and understandability of the task description. This led to improvements of CEP and minor updates of the task description. The experiment was conducted by 42 graduate and postgraduate students from the Universities of Innsbruck, Ulm, and Valencia. Subjects were randomly assigned to groups, with an equal number of subjects for each group.

Data Validation. To obtain a valid data set, we checked for completeness of the created process models. Unfortunately, 8 of the participants had to be removed due to incomplete models. As, a result, 34 subjects remained in the data set, which were equally distributed over the two groups. Since we did not consider process modeling knowledge and experience as a factor in our experiment, we screened the participants for prior knowledge regarding BPMN and change patterns. For this, a questionnaire similar to [19] was used to verify that subjects were equally distributed to the two groups. (cf. Table 1). The questionnaire used Likert scale ranging from strongly disagree (1) to strongly agree (7). To test for differences between the two groups, t-tests were run for normally distributed data. For non–normally distributed data, the Mann-Whitney test was used. No significant differences were identified between the two groups. Consequently, we conclude that no differences could be observed between the two groups.

Table 1. Demographic Data

Question	Group	Min	Max	M	SD
Familiarity with BPMN	A	2	7	5.12	0.99
	B	2	7	5.53	1.28
Confidence in understanding BPMN	A	3	7	5.53	1.33
	B	4	7	6.24	0.75
Competence using BPMN	A	3	7	5.06	1.14
	B	3	7	5.59	1.06
Familiarity change patterns	A	2	7	4.76	1.44
	B	2	7	4.53	1.46
Competence using change patterns	A	2	7	4.59	1.33
	B	2	6	4.41	1.28

4 Subjective Perception of Model Creation

This section addresses research question RQ1 dealing with the subjective perception of process modelers when using change patterns. In particular, it investigates how the used change pattern set (core vs. extended) impacts mental effort. Further, we investigate the perceived ease of use and perceived usefulness.

4.1 Mental Effort

Descriptive Statistics. The results related to mental efforts are displayed in Table 2. Mental effort was measured using a 7-point Likert scale ranging from 'very low' (1) to 'very high' (7). For Group A the mean mental effort was 3.35, corresponding approx. to 'rather low' (3). In turn, for Group B the mental effort was higher with a mean of 4.25, corresponding to 'medium' (4).

Table 2. Subjective Perception

Scale	Group	Min	Max	M	SD
Mental Effort	A	2	5	3.35	1.06
	B	3	7	4.25	1.00
Perceived Ease of Use	A	5.18	6.06	5.81	0.33
	B	4.53	5.82	5.25	0.47
Perceived Usefulness	A	4.13	4.75	4.38	0.21
	B	3.87	4.87	4.27	0.36

Hypothesis Testing. When using change patterns for process modeling, plan schemata on how to apply change patterns need to be developed in order to create complex process fragments. In this context, we investigate how the mental effort of modelers is affected by utilizing a larger change pattern set. While the extended change pattern set allows modelers to recover faster from detours, it also requires them to develop additional plan schemata on how to apply the move change patterns. Therefore, an extended change pattern set might impose higher demands on the modeler's cognitive resources. Especially, move change patterns require modelers to imagine how the process model looks like after applying change patterns. This might put additional burden on them, requiring to manipulate an internal representation of the process model in working memory. In the light of the cognitive background, we expect the extended pattern set to yield a significantly higher mental effort compared to the core pattern set.
Hypothesis H1. *The usage of an extended change pattern set significantly increases the mental effort required to accomplish the modeling task.*

Since the data was normally distributed, a t-test was used for testing the differences between the two groups ($t(31) = -2.50, p = 0.02$). The result allows us to accept hypothesis H1.

Descriptive Statistics. In order to assess how far process modelers with moderate process modeling knowledge consider the change pattern editor as easy to use and useful, we asked them to fill out the *Perceived Ease of Use (PEU)* and the *Perceived Usefulness (PU)*. Both scales consist of 7-point Likert items, ranging from 'extremely unlikely' (1) over 'neither likely nor unlikely' (4) to 'extremely likely' (7). Regarding the PEU, the mean value was 5.81 for Group A (core pattern set), corresponding approx. to 'quite likely' (6). In turn, for Group B (extended pattern set) the mean value was 5.25, corresponding approx. to 'slightly likely' (5). Finally, regarding the PU, the observed mean value was 4.38

for Group A and 4.27 for Group B, corresponding approx. to 'neither likely nor unlikely' (4) for both groups. Three participants indicated that they could not answer the questions on PU. Hence, they were removed for the analysis of PU.

Hypothesis Testing. As stated for mental effort already, the extended pattern set requires modelers to develop additional plan schemata in order to apply the change patterns properly. Accordingly, one would expect that an extended change pattern set is more difficult to use. However, these should also be perceived as more useful since the extended pattern set helps to resolve detours quicker compared to the core pattern set, i.e., when allowing to move a misplaced process fragment based on a respective pattern.

Hypothesis H2. *The usage of an extended change pattern set significantly lowers the perceived ease of use.*

Hypothesis H3. *The usage of an extended change pattern set significantly increases the perceived usefulness.*

Since none of the groups are normally distributed, we apply the Mann-Whitney U-Test to test for differences regarding PEU and PU. While significant differences in terms of PEU ($U = 4010.50, p = 0.00$) allow us to accept hypothesis H2, no statistically significant differences in terms of PU ($U = 3639.00, p = 0.06$) were observed.

Discussion. Our results indicate that the core pattern set leads to a significantly lower mental effort for modelers and its use is perceived as being significantly easier compared to the extended pattern set. This seems reasonable since modelers need to devote additional cognitive resources in order to use the move change patterns. Regarding PU, against our expectations, we could not obtain any statistically significant result. When looking at the descriptive statistics, the participants of Group B tend to perceive change patterns as less useful compared to Group A. We might conclude that the move change patterns provided for Group B are not as useful as expected (at least for the task assigned to the subjects). Alternatively, the subjects of Group B might have struggled with the usage of change patterns due to the additional patterns. In turn, this might have foiled potential positive effects of the additional patterns. The results presented in Sec. 5 support the latter explanation suggesting that process modelers had considerable problems with the use of the move patterns.

5 Challenges when Modeling with Change Patterns

This section addresses research question RQ2 aiming to obtain an in-depth understanding how the chosen pattern set impacts the challenges faced by modelers.

5.1 Data Analysis Procedure

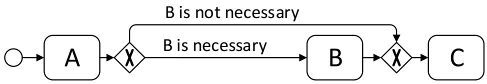
Step 1: Determine Solution Model, Distance, and Optimal Problem Solving Paths. First, we create a model representing the correct solution (i.e., S_S) for the modeling task. Subjects had to work on a re-modeling task as described in Sect. 3, i.e., in addition to an informal textual description they obtained the solution to the modeling task in the form of a graphical model. Thus, the goal state of the modeling task was clearly defined and unique, i.e., subjects should create a graphical representation of the process that exactly looks like the solution model. To be able to assess not only how closely subjects reached the goal state (i.e., how similar their resulting model is to the solution model), but also how efficiently their problem solving process was, we determine the *distance* for transforming an empty model S_0 to S_S, i.e., the minimum number of change patterns required for the respective model transformation. Generally, there are several options to create the solution model S_S by starting from S_0 and applying a sequence of model transformations. From a cognitive perspective, each sequence of change patterns that leads to S_S without detours constitutes an *optimal problem solving path*. Starting from S_0 the process fragment depicted in Fig. 3 can be created with 6 change patterns; e.g., S_S can be created by first inserting A and then B, next embedding B in a conditional branch, then updating the two transition conditions, and finally inserting C (P_0 in Fig. 3).

Step 2: Determine Deviations from Solution Model and Optimal Problem Solving Path.

To quantify the efficiency of the problem solving strategy used by the subjects to accomplish the re-modeling task, their problem solving path is analyzed. To be more specific, using the replay feature of CEP we compare the subject's problem solving path P with the optimal one and capture deviations from it. For this, every superfluous change pattern application a subject performs is counted as a *process deviation*. To quantify how close subjects reached the goal state (i.e., how similar their resulting model is to the solution model S_S) we consider *product deviations* that measure the number of incorrect change pattern applications leading to deviations between the final models created by the subjects and the solution model S_S.

Fig. 3 shows the problem solving path P_0 of one modeler who managed to model the depicted fragment correctly (i.e., 0 process deviations and 0 product deviations). Problem solving path P_2, in turn, leads to a correct goal state (i.e., 0 product deviations). However, the modeler made a detour of 2 change patterns before reaching the solution (i.e., solution path P_2 comprises 2 superfluous change patterns summing up to 2 process deviations). Now assume that the modeler, who took a detour when creating the process model, did not correct the introduced error ending up with an incorrect process model (cf. path $P1$ in Fig. 3). The application of the Embed in Loop pattern (instead of Embed in Conditional Branch) constitutes 1 product deviation (i.e., the modeler applied one incorrect change pattern that led to an incorrect goal state).

Since not every subject reached the goal state (i.e., their models contain product deviations), the direct comparison of process deviations might favor modelers that left out parts that were difficult to model and where other subjects produced a high number of process deviations. To decrease this bias we consider a second measure for operationalizing problem solving efficiency. In addition to the process deviations described above this measure considers the effort needed to correct



P_0 = < Serial Insert (A), Serial Insert (B), Embed (B) in Cond. Branch,
Update Condition (B is not necessary),
Update Condition (B is necessary), Serial Insert (C) >

P_1 = < Serial Insert (A), Serial Insert (B), Embed (B) in Loop,
Update Condition (B is not necessary),
Update Condition (B is necessary), Serial Insert (C) >

P_2 = < Serial Insert (A), Serial Insert (B), ~~Embed (B) in Loop~~,
~~Undo Embed (B) in Loop~~, Embed (B) in Cond. Branch,
Update Condition (B is not necessary),
Update Condition (B is necessary), Serial Insert (C) >

Fig. 3. Process Deviations, Product Deviations, and Fixing Steps

an incorrect process model (denoted as *fixing steps*), i.e., the steps needed to transform the created model into S_S. For example, to correct the model that resulted from $P1$ in Fig. 3, 5 fixing steps are needed, irrespective of whether or not the core or the extended change pattern set is used. First the fragment embedded in the loop has to be deleted. Next, B has to be re-inserted and embedded in a conditional branch, and then the two transition conditions must be updated. Fixing steps and process deviations are then combined in a single measure called *total process deviations*. This measure does not only consider detours (i.e., process deviations), but also model transformations that would be needed to correct the process model (i.e., resolving product deviations).

Step 3: Detection of Outliers. In order to limit the influence of modelers who are experiencing severe difficulties during the creation of the process model, we test for outliers regarding the number of process deviations. For this purpose, we utilize the Median Absolute Deviation (MAD) to detect outliers. More specifically, we apply a rather conservative criterion for removing outliers by removing values that differ at least 3 times the MAD from the median [20]. As a result, one PPM instance was removed from Group B regarding further analysis.

5.2 Results

Descriptive Statistics. To create a correct solution model 28 operations are needed. Overall, 123 process deviations (i.e., detours in the modeling process) and 44 product deviations (i.e., deviations of the final models from the solution model) were identified (cf. Table 3). From the 123 process deviations 60 can be attributed to Group A (3.53 deviations per subject), while 63 were found for Group B (3.94 deviations per subject). In terms of product deviations they were equally distributed among the two groups, i.e., 22 product deviations per group (1.29 deviations per subject in Group A and 1.38 deviations per subject in Group B). In order to resolve the product deviations, 45 fixing steps are required for the models of Group A and 29 fixing steps for Group B resulting in 105 and 92 total process deviations respectively.

Table 3. Overview of Deviations

Scale	Group A	Group B
Process deviations	60	63
Average Process deviations per modeler	3.53	3.94
Product deviations	22	22
Average Product deviations per modeler	1.29	1.38
Fixing steps	45	29
Average fixing steps per modeler	2.65	1.81
Total process deviations	105	92
Average total process deviations per modeler	6.18	5.75

Hypothesis Testing. We test for significant differences between the two groups regarding process deviations and total process deviations. We expect that the modelers using the extended pattern set have significantly fewer process deviations, because the extended pattern set allows them to resolve detours with fewer steps. Moreover, we expect an impact on the total process deviations, since the extended pattern set allows transforming the model created by the modelers with fewer steps into the solution model.

Hypothesis H4. *The usage of an extended change pattern set significantly decreases the number of process deviations.*

Hypothesis H5. *The usage of an extended change pattern set significantly decreases the number of total process deviations.*

To test for differences in terms of process and total process deviations, we apply the t-test since the data was normally distributed. No statistical difference could be observed for process deviations ($t(31) = -0.24, p = 0.82$) or total process deviations ($t(31) = 0.25, p = 0.81$).

Discussion. Our results did not yield statistically significant differences between the two groups. This indicates that the usage of an extended change pattern set might not have an impact on both process deviations and total process deviations. An alternative explanation could be that process modelers did not use the provided patterns frequently enough to obtain statistically significant differences (i.e., pattern adoption was low). Another explanation could be that subjects did not use the patterns effectively, canceling out a potential positive impact. To investigate these alternative explanations in more detail Sect. 6 analyzes the actual use of the move change patterns.

6 Actual and Potential Use of an Extended Pattern Set

This section addresses research question RQ3 which deals with the actual use of the additional change patterns compared to the potential usage of those patterns. The analysis of invocations of the move change patterns revealed that the serial move pattern was only applied 3 times (by 3 different participants), whereas the parallel move pattern was used 18 times (by 7 different participants). This indicates that the subjects adopted the move patterns only to a limited extend.

Out of the 21 move pattern applications, 11 were correct; i.e., they led to correct intermediate models, either directly through the application of the pattern or the application of the pattern in combination with additional patterns. In turn, 10 applications of the parallel move pattern were incorrect and either led to an undesired model or did not yield any changes of the model. This indicates that subjects had difficulties when applying the move change patterns.

Though the actual use of the move patterns was limited, we investigate their theoretical potential. For this, we analyze the number of fixing steps required to correct a model with product deviations (i.e., to transform it into S_S). We further analyze how the availability of an extended pattern set impacts this measure. In a second step, we analyze the potential of an extended pattern set for reducing process deviations, i.e., by enabling a faster resolution of detours.

Table 4. Potential Use of the Move Change Pattern

Scale	Group A	Group B
Fixing steps with move	45	64
Fixing steps without move	25	29
Saved operations	20	35
Process Deviations	60	63
Unnecessary Operations	-	15
Saved operations	9	0
Potential process deviations	51	48

To show the potential of an extended pattern set for resolving product deviations, Table 4 depicts the number of fixing steps, when using the core pattern set and for the extended pattern set. For Group A, 45 fixing steps are required to correct all product deviations that occurred. By making the extended pattern set available to Group A, this number could be reduced to 25 (i.e., 20 fixing steps could be saved). In turn, for Group B the number of observed fixing steps is 29. Without the extended pattern set, however, 64 fixing steps would be needed. This indicates the theoretical potential of the extended pattern set for reducing the number of fixing steps and, thus, the number of total process deviations.

To investigate the potential for reducing process deviations for Group A, we analyze whether process deviations could have been reduced when using move change patterns. In turn, for Group B we focused on the number of operations that would have been saved if move patterns were always applied correctly. As illustrated in Table 4, 9 operations could be saved if the move pattern had been available for Group A resulting in 51 potential process deviations. Regarding Group B, 15 operations could have been saved through correct pattern application resulting in 48 potential process deviations.

Discussion. These results suggest that a theoretical potential for using move change patterns exists. However, the subjects used the move change patterns only to a limited extent and had troubles with their correct application. As a consequence the potential of the additional patterns could not be fully exploited. Since mental effort and perceived ease of use is lower with the core pattern set it might be more favorable to use the core pattern set for process modelers that are only moderately familiar with process modeling and are no experts in the usage of change patterns. We might speculate that the extended pattern set could be promising for more experienced users (who are literate in pattern usage).

7 Limitations

As with every other research, this work is subject to several limitations. Certainly, the relatively small sample size constitutes a threat regarding the generalization of our results. Using students instead of professionals poses another threat regarding external validity. In previous research with software engineering students it has been shown that students may provide an adequate model for the professional population [15,16,17]. Still, generalizations should be made with care. Moreover, since we used subjects who were moderately familiar with process modeling and change patterns results cannot be generalized to expert modelers. It can be assumed that process modelers experienced with the usage of change patterns will presumably face less problems during model creation and will be able to apply patterns more effectively. Another limitation relates to the fact that we used only one modeling task in our study. The potential benefit of move patterns, however, depends on the structure of the process model to be created. For more complex process models with higher nesting depth the potential usefulness might be higher. Thus, it is questionable in how far results may be generalized to models with different characteristics. As a consequence, we plan further experiments testing the impact of model structure on challenges regarding change pattern usage. Moreover, this work compares two particular change pattern sets. Using an extended change pattern set with different patterns (e.g., a pattern to change a conditional fragment into a parallel fragment or to change a conditional fragment to a loop) might lead to different results. Another limitation regarding the external validity relates to the process modeling notation (i.e., BPMN) and the modeling tool used (i.e., CEP). Results might be different when using other modeling languages or different modeling tools.

8 Related Work

The presented work relates to research developed in the context of the creation of process models and process model creation patterns.

Research on the creation of process models builds on observations of modeling practice and distills normative procedures for steering the process of modeling toward successful completion. To do so, [21,22] deal with structured discussions among different parties (system analysts, domain experts). In this line of research, [23] analyzes the procedure of developing process models in a team, while [24] discusses participative modeling. Complementary to these works, whose focus is on the effective interaction between the involved stakeholders, our work focuses is on the *formalization* of the process model.

Researchers have also focused on the interactions with the modeling environment, i.e., the PPM. [8] identified three distinct modeling styles, whereas [6,25] suggest different visualization techniques for obtaining an overview of the PPM; [7] demonstrates that a structured modeling style leads to models of better quality. [26] investigates the PPM using eye movement analysis. While these works focus on interactions with the modeling environment based on change primitives, this paper investigates the use of change patterns.

Change patterns for process model creation have been investigated as well; e.g., AristaFlow allows modeling a sound process schema based on an extensible set of change patterns [12]. [27] describes a set of pattern compounds, comparable to change patterns, allowing for the context-sensitive selection and composition of workflow patterns. Complementary to these works, which have a strong design focus, this paper provides empirical insights into the usage of change patterns. More precisely, it builds upon the results obtained in [9], which describes recurring challenges modelers face during the PPM using change patterns.

9 Summary

While recent research has contributed to a better understanding regarding the PPM, little is known about this process when utilizing change patterns. In this experiment we investigate the impact of the available patterns on the PPM and the modeler's perception. The results indicate that an extended change pattern set puts an additional burden on modelers who perceive them as more difficult to use. In addition, when using these patterns, subjects faced considerable difficulties. Therefore, (against our expectations) our data does not indicate an increased problem solving efficiency, i.e., the expected benefits of using the extended change pattern set did not materialize. This indicates that the change pattern set should be selected with care, especially for modelers with limited experience. Future research should include investigations on new change pattern sets having a (theoretical) potential for reducing process deviations, e.g., a pattern to change a conditional fragment into a parallel or a loop fragment.

References

1. 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)
2. Kock, N., Verville, J., Danesh-Pajou, A., DeLuca, D.: Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. Decision Support Systems 46, 562–575 (2009)
3. Mendling, J., Verbeek, H.M.W., van Dongen, B.F., van der Aalst, W.M.P., Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model. Data and Knowledge Engineering 64, 312–329 (2008)
4. Weber, B., Reichert, M., Rinderle, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering 66, 438–466 (2008)

5. Casati, F.: Models, Semantics, and Formal Methods for the design of Workflows and their Exceptions. PhD thesis, Milano (1998)
6. Pinggera, J., Zugal, S., Weidlich, M., Fahland, D., Weber, B., Mendling, J., Reijers, H.A.: Tracing the Process of Process Modeling with Modeling Phase Diagrams. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 370–382. Springer, Heidelberg (2012)
7. Claes, J., et al.: Tying Process Model Quality to the Modeling Process: The Impact of Structuring, Movement, and Speed. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 33–48. Springer, Heidelberg (2012)
8. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H., Mendling, J.: Styles in business process modeling: An exploration and a model. Software & Systems Modeling, 1–26 (2013)
9. Weber, B., Pinggera, J., Torres, V., Reichert, M.: Change Patterns in Use: A Critical Evaluation. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 261–276. Springer, Heidelberg (2013)
10. 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)
11. Rinderle-Ma, S., Reichert, M., Weber, B.: On the formal semantics of change patterns in process-aware information systems. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 279–293. Springer, Heidelberg (2008)
12. Dadam, P., Reichert, M.: The ADEPT project: A decade of research and development for robust and flexible process support. Comp. Scie. - R&D 23, 81–97 (2009)
13. Fahland, D., Woith, H.: Towards process models for disaster response. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008 Workshops. LNBIP, vol. 17, pp. 254–265. Springer, Heidelberg (2009)
14. Pinggera, J., Zugal, S., Weber, B., Fahland, D., Weidlich, M., Mendling, J., Reijers, H.A.: How the Structuring of Domain Knowledge Can Help Casual Process Modelers. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 445–451. Springer, Heidelberg (2010)
15. Höst, M., Regnell, B., Wohlin, C.: Using students as subjects-a comparative study of students and professionals in lead-time impact assessment. Empirical Software Engineering 5, 201–214 (2000)
16. Porter, A.A., Votta, L.G.: Comparing detection methods for software requirements inspections: A replication using professional subjects. Empirical Software Engineering 3, 355–379 (1998)
17. Runeson, P.: Using students as experiment subjects—an analysis on graduate and freshmen student data. In: Proc. EASE 2003, pp. 95–102 (2003)
18. Davis, F.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13, 319–340 (1989)
19. 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)
20. Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L.: Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology 49, 764–766 (2013)
21. Frederiks, P., Weide, T.: Information modeling: The process and the required competencies of its participants. Data and Knowledge Engineering 58, 4–20 (2006)

22. Hoppenbrouwers, S.J.B.A., Proper, H.A(E.), 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)
23. Rittgen, P.: Negotiating Models. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 561–573. Springer, Heidelberg (2007)
24. 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)
25. Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H.A., Weber, B., Poels, G.: Visualizing the Process of Process Modeling with PPMCharts. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 744–755. Springer, Heidelberg (2013)
26. Pinggera, J., Furtner, M., Martini, M., Sachse, P., Reiter, K., Zugal, S., Weber, B.: Investigating the Process of Process Modeling with Eye Movement Analysis. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 438–450. Springer, Heidelberg (2013)
27. Gschwind, T., Koehler, J., Wong, J.: Applying patterns during business process modeling. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 4–19. Springer, Heidelberg (2008)

A Participative End-User Modeling Approach for Business Process Requirements

Agnès Front, Dominique Rieu, and Marco Santorum

Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
CNRS, LIG, F-38000 Grenoble, France
{Agnes.Front,Dominique.Rieu,Marco.Santorum}@imag.fr

Abstract. A business process can be characterized by multiple perspectives (intentional, organizational, operational, functional, interactional, informational, etc.). Business process modeling must allow different stakeholders to analyze and represent process models according to these different perspectives. This representation is traditionally built using classical data acquisition methods together with a process representation language such as BPMN or UML. These techniques and specialized languages can easily become hard, complex and time consuming. In this paper we propose ISEA, a participative end-user modeling approach that allows the stakeholders in a business process to collaborate together in a simple way to communicate the business process requirements in an accurate and understandable manner. Our approach covers the organizational perspective of business processes, exploits the information compiled during the simulation of the processes in the organizational perspective and touches lightly an interactional perspective allowing users to create customized interface sketches to test the user interface navigability and the coherence within the processes. Thus, ISEA can be seen as a participative end-user modeling approach for business process requirements.

Keywords: business process management, requirements engineering, domain modeling, user interfaces modeling, participative approach.

1 Introduction

Business Process Management is an important best practice that is critical for the long-term success of an organization and provides important benefits to organizations [18]. Modeling business processes may have different goals: align the organizational processes with users' needs, explain or automatize the different processes, evolve the conduct of the business in order to adapt it more rapidly to change, etc. Business process modeling techniques must enable the different stakeholders to analyze and represent business processes according to different and adapted perspectives (intentional, functional, organizational…) [8, 21, 24, 29].

Moreover, business process representations are traditionally built using classical data acquisition methods (interviews, observations, transcription of activities, text analysis, etc.) together with a process representation language such as BPMN or

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 33–47, 2014.

UML. These techniques and specialized languages can easily become hard, complex and time consuming particularly if the organization does not have formal and clear process description documents or if the stakeholders proceed mechanically without real conscience of the task.

On the contrary, participative approaches for business process improvement recommend a strong implication of the users [20, 28]. These approaches improve time and quality needed for the acquisition of the useful information to understand and improve the processes. However the obtained representations are not enough formalized, they don't correspond to models conformed to a formal modeling language.

Aware of these facts, we adopted an iterative and end-user centered design approach to involve functional actors of specific business processes throughout the representation of the useful process perspectives. End-users are indeed the ones that have the knowledge and have to use the system in the end, thus they should really know what is expected. Our approach called ISEA[1] can be seen as a participative end-user modeling approach for business process requirements in order to obtain sketches of models convertible in standard languages, all of them elaborated in a consensus-based manner. It is particularly adapted to existing business processes which need to be improved. Although it was developed and evaluated in the context of university business processes, it is generic and can be suited to different business processes domains to model and improve existing processes.

In the following, section 2 gives an overview of ISEA and describes by a MAP [10] the three perspectives yet covered by ISEA. Section 3 focuses on a particular path of ISEA, detailing some participative modeling activities proposed in the method. Section 4 presents an experimental research method used to co-construct and validate ISEA, this experimental research method is based on a user centered experimental validation cycle. Section 5 compares our approach to some related works in multi-perspective business process modeling and participative approaches for business process improvement. Finally, section 6 concludes the paper and describes the further works to be carried out.

2 ISEA a Participative End-User Modeling Method

The ISEA method allows modeling business processes following organizational, informational and interactional perspectives. We emphasized at first these three perspectives that are particularly suited for modeling and improvement of existing business processes. The informational perspective is based on the information compiled during the simulation of the process in the organizational perspective. The interactional perspective, based on organizational and informational perspectives, allows users to create customized interface sketches to test the user interface navigability and the coherence with the process. Figure 1 describes by a MAP [10] these three perspectives. Each perspective is characterized by three goals:

[1] ISEA: Identification, Simulation, Evaluation, Amelioration (http://www.iseamethod.com/)

- Model elicitation allows representing each view of the process using a Domain Specific Language (DSL) adapted to business process modeling [25]. Such DSL are use in similar works like PICTURE [31]. The strategies for model elicitation are participative and use role-playing simulation games [26]. For example, section 3 will describe the DSL as simple languages and graphical representations such as trees that we use in order to make them understandable by the functional actors of a process.
- Model evaluation highlights the process difficulties and dysfunctions. This evaluation is realized from the end-user process models.
- Model transformation aims at transforming the different perspectives into standardized or common languages. The transformed models called analysis models are obtained from end-user models; they are quite poor, representing the concepts identified by the users. The development team should be enriched them in order to be automated.

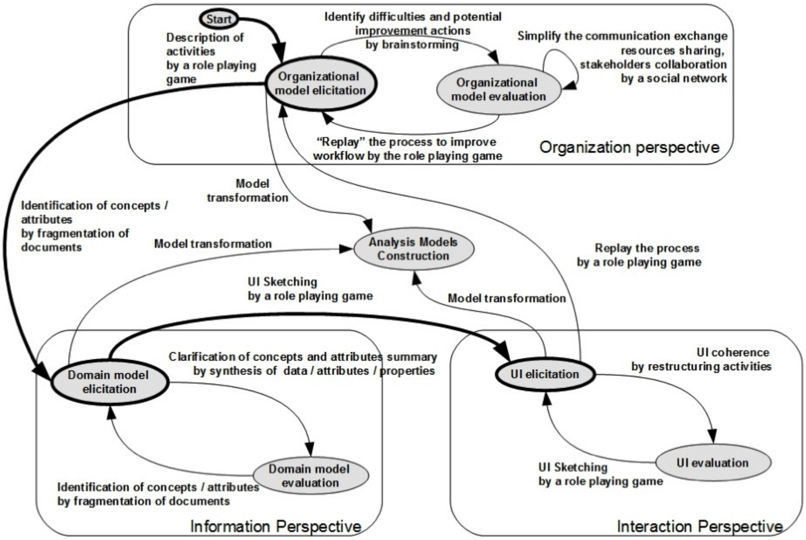


Fig. 1. The ISEA method

In an initiative of business process cartography, all the goals are not necessary reached. If the goal is only to facilitate the communication in the business team by a better understanding of each actor's role, the first purpose (elicitation) is sufficient. The second goal (evaluation) is essential to improve the processes. The third goal (transformation) is necessary to automate the processes. In this case, the end-user models constitute consensual requirements models allowing the development of process-aware information systems.

Figure 1 also highlights the strategies for achieving these three goals. All the strategies have in common to be based on participative and playful approaches. Each strategy is supported by a partially ordered set of individual or collective activities. For example, the strategy "Description of activities by a role playing game" is supported by several activities. The main activity is collaborative: the functional actors of a business process, with the help of an animator, collectively elaborate the description of the business activities and the exchanged documents. A particular path of the map, in fat-line in figure 1, is detailed in the following section.

3 Exploring a Path in ISEA

This section illustrates one possible path of the ISEA map, this path (in fat line in Figure 1) allows:

- Elicitation of the organizational perspective by a role playing game in which each participant plays its own role,
- Elicitation of the informational perspective by individual and collective activities dedicated to the exploitation of the documents identified in the organizational perspective,
- Elicitation of the interactional perspective by individual and collective activities dedicated to the sketching of the business activities identified in the organizational perspective.

3.1 Elicitation of the Organizational Perspective

We illustrate here the strategy starting from "Start" to "Organizational model elicitation". This strategy corresponds to the first phase of the ISEA method where end-users collaborate around the creation and maintenance of existing process cartographies. The goal is to elicit an organizational model corresponding to a business process expressed using a very simple domain specific language and representing all the activities and documents exchanges. In this strategy as in the whole ISEA method, all stakeholders are involved, and more particularly the end-users, who are the domain experts and possess the necessary knowledge of how the processes should operate, which tasks have to been carried out, which business rules need to be enforced, the validation checks to perform, etc.

The DSL is composed of graphic elements (see Figure 2), which are involved in the construction of the organizational perspective:

- A yellow post-it represents an activity, which may be decomposed in several actions. An action consists of a verb conjugated in the first person singular (e.g. "I ask") and a medium (e.g. "by email") or document if needed (e.g. "a quote").
- A pink post-it represents the intervention of an external actor in the process.
- The colored lines show the flow between activities.
- A “loop symbol” represents a repetitive activity, a “clock symbol” is a timer event that executes an activity at a specific time or at a given time and a “stop symbol” represents the end of an actor participation in the process.
- A “document symbol” is used to represent a document produced or used by an activity. The documents of the same process have different colors. A document is described by a short description (document name, abbreviation). A pdf file which corresponds to the real document, can be attached.

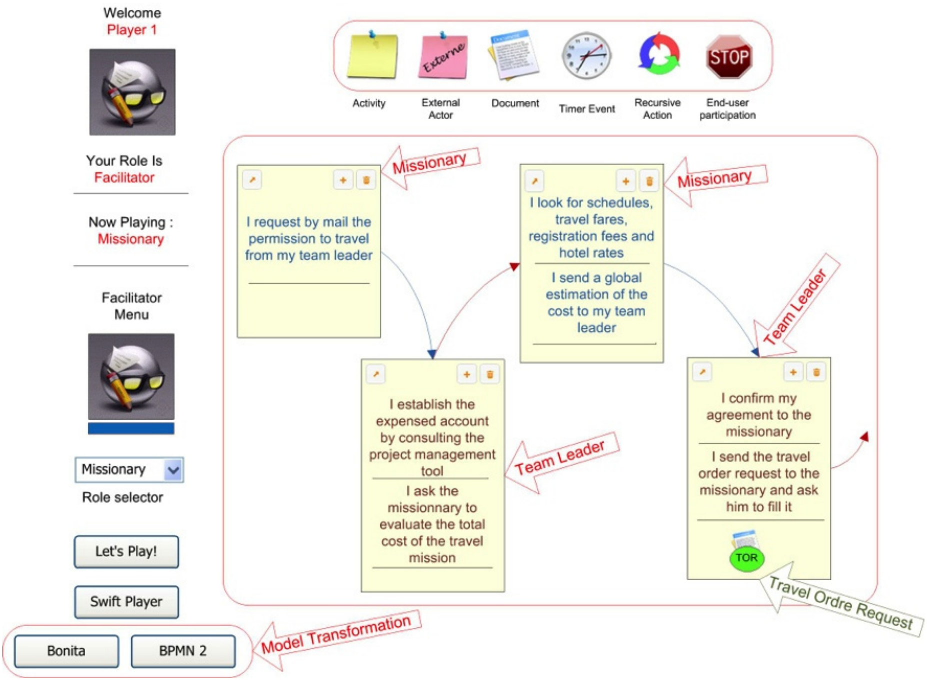


Fig. 2. Organizational perspective: a business process representation in ISEAsy

Figure 3 shows all the participants playing with the role-playing simulation game and using the tool support ISEAsy. In this game, participants assume a role and act out a real-life situation in order to get in a participative way, a description of the daily activities during a specific process. Each participant plays the same role as he has in real life. He uses the set of graphic elements with which he represents the actions performed during real life. A participant places a virtual post-it on the workspace to represent an activity he accomplishes during the process and draws one or more arrows handing over the turn to the next participants. Participants take their turn, one after the other, depending on the situation, as would occur in real life. As an example, in a travel management process, the game begins with the missionary, who needs to establish a mission request. The Document creation is a specific action. If a participant needs a document previously created, he drags the respective color coding label into his post-it. If the intervention of an external actor is necessary, the facilitator plays this role by dragging a pink post-it. No action is noted on this post-it, only documents may move on it. Figure 3 shows the result of the role-playing game in the tool ISEAsy support of the method ISEA. The result is very similar to a BPMN basic process. Just like the models described in BPMN, the organizational perspective includes behavioral (dependences between activities) and informational (documents) dimensions.

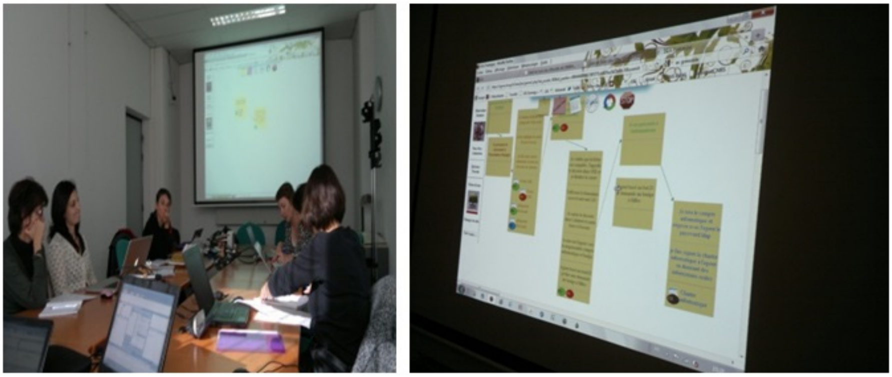


Fig. 3. The role playing game using the tool ISEAsy

3.2 Elicitation of the Informational Perspective

We illustrate here the strategy starting from "Organizational model elicitation" to "Informational model elicitation". The goal here is to get, once again using a participative approach, a simple domain model that could be transformed and completed by an analyst in a more formal domain model using modeling standards such as UML or Entity/Association. The starting point is the files attached in the documents elicited in the organizational perspective. Three main activities are proposed:

- Individual activity "Cutting ": in a first phase, the facilitator distributes 2 or 3 documents to each actor who is individually asked to cut in each document the different fragments that seem pertinent to be grouped together. For example, a participant may cut fragments on a document corresponding to information on a missionary, and in the same document, information on the travel (departure date, arrival date, etc.) (see Figure 4). This activity is individual and must not exceed ten minutes; otherwise the participants may be bored.
- Collective activity "Model elaboration": participants are collectively asked to place the document fragments on a tree symbolizing a tree of concepts. One after the other, the actors place the different fragments either in a new branch of the tree symbolizing a new concept (for example, the new concept Travel), either in an existing branch symbolizing new elements to describe an existing concept (for example, information added to the concept Missionary) (see Figure 5). When all fragments are placed on the tree, the individual activity "Cutting" iterates until the whole documents are cut.

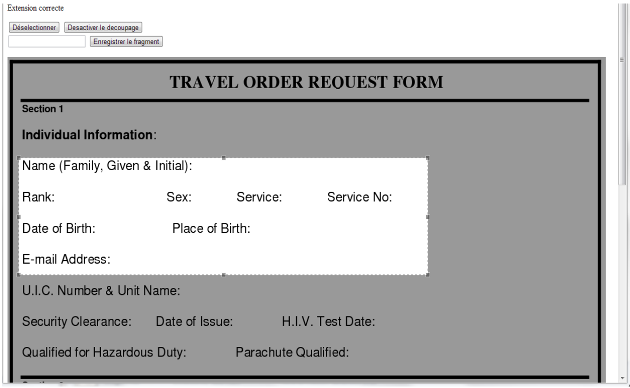


Fig. 4. The "cutting" activity: a fragment in a document

- Collective activity "Conflict resolution": several fragments of different documents may represent the same information. For example, name and date of birth of a missionary may exist on different documents. In this case, actors place the information on the same branch, and the facilitator will take a time to resolve these conflicts. The final result is a tree of concepts representing a simple domain model, each branch corresponding to a concept. The tree of concepts can easily be transformed, thanks to automatic transformation rules, into a more formalized domain model (in UML for example) in which the main concepts (branches) and sub-concepts (branches of branches, see figure 5) are identified. An analyst should then work on

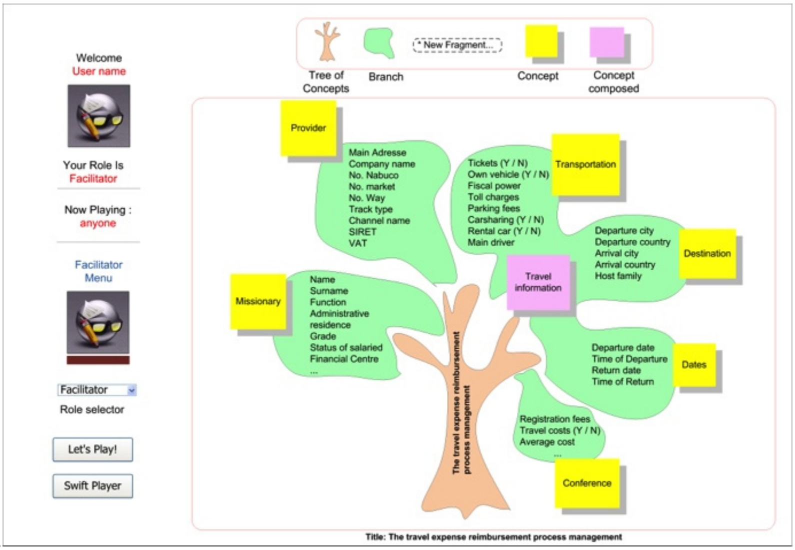


Fig. 5. Informational perspective: the tree of concepts in ISEAsy

the domain model in order to add relationships and multiplicities, and make the domain model more precise (for example, adding specializations or aggregations).

3.3 Elicitation of the Interactional Perspective

We illustrate here the strategy starting from "Informational model elicitation" to "interactional elicitation". The goal here is to create customized interface sketches to test the user interface navigability and its coherence within the process and perhaps to propose process improvements. The proposed sketches could then be transformed and completed by a user interface specialist in executable and standardized UI models. As for the other perspectives, different activities are proposed:

- Individual activity "Sketching": thanks to the business process model resulting from the elicitation of the organizational perspective and to the tree of concepts resulting from the elicitation of the informational perspective, each participant is asked to imagine the user interface he would like in order to realize its activities in the most efficient way, and perhaps to resolve the potential difficulties identified during the organizational model evaluation (see Figure 1). For example, in this last phase (not illustrated here for space reasons), missionary and team-leader were bored with too much message exchanges at the beginning of the process (different message exchanges between them in Figure 3). To resolve this difficulty, the missionary may first imagine to be proposed a menu where he could estimate the price of the mission, look for an estimation of the price of the hotel and transport on adequate web sites, look for the conference rates on the conference website, and only then contact his team leader to get an approval (see the screen of the missionary in figure 6). In the same way, the team leader could imagine the sketching of the user interface allowing him to receive emails when he has to validate or refuse a request (see the screen of the team leader in figure 6). To construct their interfaces, missionary and team leader can use existing UI sketching tools such as Balsamiq[2].
- Collective activity "Navigation validation": the facilitator draws the navigation between the proposed sketches using tools such as Gambit [27]. Participants are then asked to validate the navigation or to correct their interfaces in order to be satisfied (see Figure 6).

The result of this perspective is a representation of the ideal interfaces and screens navigations between actors. This ideal representation may have an impact on the process model evolution (process "to be"). In this case, the process model should be corrected in the organizational perspective.

[2] Balsamiq, `http://www.balsamiq.com`

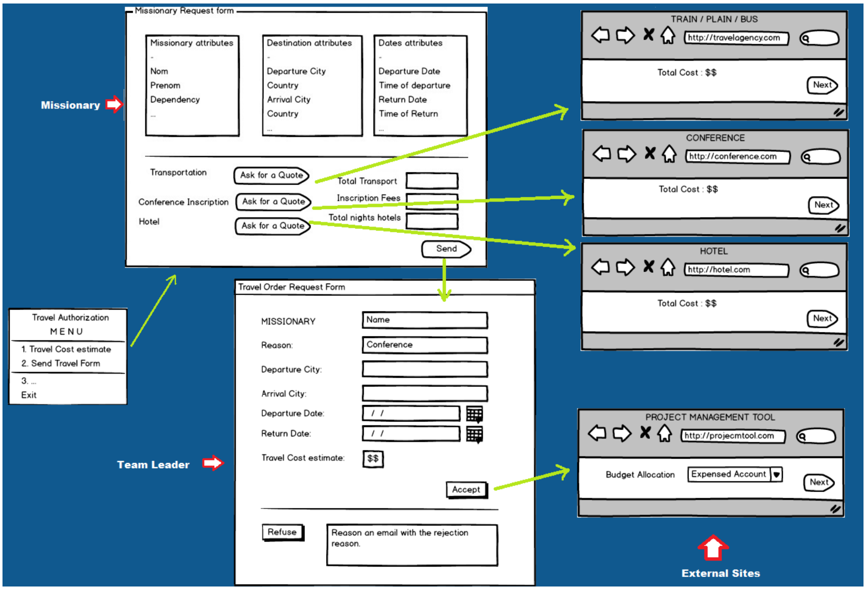


Fig. 6. Interactional perspective: navigation between sketches

4 Validation of the ISEA Method

The modeling languages and elicitation/evaluation/transformation activities were designed and validated adopting a modeling language development cycle that we proposed as previous research results [16]. This development cycle for modeling languages (see figure 7) is based on the integration of user centered experimental practices and is based on three stages:

- Analysis stage aims to validate the language dictionary and to sketch the language notation (concrete syntax).
- Design stage allows validating the notation and the abstract syntax (meta-model).
- Operationalization stage is dedicated to validate language support tools.

During the analysis stage, the languages and elicitation/evaluation/transformation activities are validated by a restrict circle of users, on average two processes, each process involving from 6 to 8 actors. This restrict circle is spread during the design stage (from 4 to 6 processes). The tools are generally validated by the same users as in design so as to measure the acceptability degree of the method in a mediatized mode. During each stage (analysis, design, operationalization), experiments are led in a purpose of validation, but also exploration and co-construction with the future users. For the co-construction of the end-user modeling language, we generally start with a standard language (BPMN, EA, etc.) that we prune until it is easy to understand and to use in a participative mode. The notation is adapted to be pleasant and talking related to the ergonomic criteria defined in [2, 17]. The concrete syntax once integrated in the tool is again considered as a component to be estimated. Indeed, the concrete syntax in its operational shape is frequently slightly different from that produced in the design time: in particular, icons may be different in a tool or with a pencil/paper.

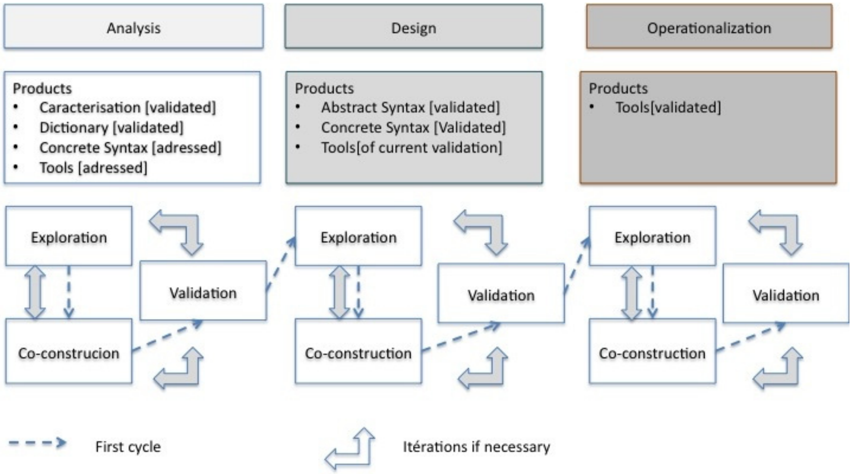


Fig. 7. User-centered validation cycle integrated in a language development process

Following this development cycle, the different perspectives of the ISEA method are in different maturity levels:

Organizational Perspective

The organizational perspective language (dictionary, notation and abstract syntax) and the elicitation/evaluation/transformation activities dedicated to this perspective are completely validated with end-users. In particular, the tool ISEAsy was validated with around twenty business processes. The multiple user-centered experimentations lead us to different evolutions of the language and the activities. In particular during analysis and design stages, the language was considerably simplified in order to be comprehensible by the end-users. We suppressed a lot of elements that we firstly thought useful for the end-users to model the processes: conditions, repetitive actions, actions composing an activity, etc. At the end, the DSL proposed in the organizational perspective contains very few simple elements: activity, external actor, document, timer event, recursive action, end of participation and change of actor (see Figure 3).

Informational Perspective

The informational perspective language (dictionary, notation and abstract syntax) and the elicitation activities dedicated to this perspective are completely validated. The tool supporting the language and the evaluation/transformation activities are on current validation. The user-centered experimentations we made lead us to different evolutions. In particular during the analysis and the design stages:

- other types of model domain notations were experimented, for example houses composed of different levels (a house being a concept and a level being an element of this concept), towns composed of different houses (a house being a concept and roads between houses being relationships between concepts), ... These representations were not approved by the end-users which finally proposed the idea of the tree (see Figure 8) that we adopted in our last version of the method (see Figure 5).

- during the first experiments, we proposed notations allowing to express entities but also associations. The associations expressed were not exploitable to build an analysis model and we removed this notion, letting the analyst complete the model with associations if needed.

Fig. 8. Co-construction with the end-users of the informational perspective representation

Interactional Perspective

As for the two other perspectives, we lead several user-centered experimentations in order to co-construct with the end-users the interactional perspective language and elicitation activities. Figure 9 shows examples of hand-made sketches using the domain model to help the user to design the desired sketches.

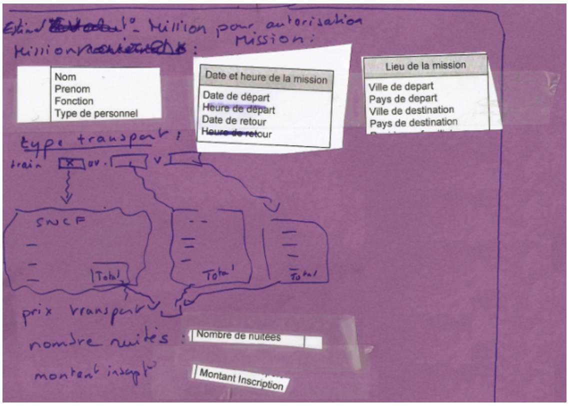


Fig. 9. Co-construction of the interactional perspective: an "hand-made" screen

The interactional perspective is yet in co-construction. If the dictionary and the concrete syntax are validated, the abstract syntax is under definition. The tool supporting the language and the elicitation activities is only addressed: it is a prototype built with existing tools such as Balsamiq and Gambit, allowing us to lead experiments.

5 Related Works

The ISEA method is based on one hand on the multi-perspective business processes modeling domain and on the other hand on participative approaches for business processes improvement.

Business processes modeling usually combines multiple perspectives. In [30], Sheer presents an Architecture of Integrated Information Systems (ARIS) based on four business process perspectives: organizational, data, control and functional. In [29] five perspectives have been proposed: functional (what has to be executed during a process), process (conditions to execute a process and the activities that have to be performed), organization (organizational structure and actors), information (which data have to be processed by business processes) and operation (elementary operations performed by resources and applications perspectives). In [7], authors extend these perspectives with the intentional perspective that represents goals and strategies that the enterprise implements in its processes. Furthermore, several works have been proposed in order to bridge the gap between the different perspectives, in particular between intentional and organizational perspectives [4, 6, 12, 13, 19, 22]. For example, [8] describes a business process-driven requirements engineering approach to derive requirements from organizational models that express business strategies and from business processes in BPMN. In [4], authors propose a method for eliciting non-functional goals from business processes. Moreover, [6] introduces the notion of intentional fragments to bridge the gap between process models and goal models. These multiple perspectives business processes modeling approaches are formal and allow achieve a global vision of the different characteristics involved in a business process. They can be used by analysts in order to help them to bridge the different perspectives of a business process. Different BP modeling perspectives can be explored and extended according to the organization and different needs or situations that enterprises can encounter. However these approaches are not participative, prioritize the results and do not provide the resources that stimulate collaboration between the different stakeholders of the process.

In the other side, participative approaches, mainly based on quality tools, involve the stakeholders of a process in the proposition of ideas for process improvement, use techniques to stimulate and motivate people, help to solve problems and propose creative solutions. Process improvement concerns the set of actions realized to identify, analyze and improve existing business processes to better match the organizational users' needs. There are several proposals, methods, tools and techniques in the field of processes improvement, from individual problem solving, rapid team problem solving, and quality tools to improve processes. Thus, Ishikawa [11], McConnell [14] and McQuater [15] propose a list of tools and techniques for quality improvement. Based on these quality tools, approaches such as [3, 5, 20, 28] are participative approaches, using for example brainstorming tools to generate new ideas for process improvement. The DMAIC methodology [23, 28] also uses quality-management tools to improve existing business processes. A participative problem-structuring methodology is presented in [1]. According to the authors, the proposed framework stimulates the interaction and makes participants more accountable to improve business processes in a holistic manner.

In general, such participative approaches don't propose multi-perspectives modelling, they are not based on modeling languages and they are not integrated in a traditional business process development cycle. The goal of ISEA was to propose a participative end-user modeling method for business processes modeling. Such an approach was also proposed in [9] where BPMN diagrams are validated by end-users and analyzed by systems analysts in order to reach an agreement on the effect that the information system will have on the organization, but this approach doesn't propose multi-perspective modeling.

6 Conclusion and Further Works

ISEA is a participative end-user modeling method for business process which proposes multi-perspective business processes modeling and improvement. The modeling process is defined in a map where the goals are to elicit and evaluate end-users models and to construct analysis model. The strategies between the goals are realized by participative and playful activities.

For the moment, ISEA allows modeling three perspectives: organizational, informational and interactional. All the proposed strategies have not the same maturity degree. The strategies for the elicitation/evaluation/transformation of the organizational perspective were the results of several evaluations. ISEAsy, the support tool of ISEA method, is used for the elicitation and improvement of the Grenoble University business processes. Discussions are in progress for its use within the RELIER network (Quality network for higher education and scientific research). The tool integrates a basic transformation into BPMN, the resulting models are accessible with the BPM tool Bonita[3]. Proposals for the informational perspective are in the operationalization stage: the tool integrated in ISEAsy was the object of demonstrations, which allowed improving the elicitation of the informational perspective. However, the validation experiments remain to made. The proposals for the interactional perspective were only evaluated by a restricted set of users (only one process with 6 participants).

The first purpose is to complete the evaluation of models and strategies of the informational and interactional perspectives, and to complete the map with other perspectives, for example, intentional or decisional perspectives that are particularly useful for elicitation of innovative processes. The second purpose is to take into account new types of processes: right now, ISEA is suited to existing administrative processes: we have started a study to evaluate the usability of ISEA on co-design processes in industrial organizations. Long-term perspectives are to apply ISEA to

[3] `http://www.bonita.com`

other domains than business process management, more particular for knowledge acquisition in the context of innovative collaborative projects.

We are convinced that the two purposes "new business processes perspectives" and "new types of processes" are linked. For example, intentional and decisional perspectives will be essential to elicit innovative processes. During the previous experiments on the university processes, we first proposed an intentional perspective aimed to identify process goals. Nevertheless, in the context of our experiments (existing processes which need to be improved), this step didn't seem to be useful to the participants who wanted to focus on their daily activities and on the encountered problems.

Acknowledgments. The authors thank the Government of Ecuador (SENACYT - EPN) for funding this research and the MARVELIG platform for supporting the experiments.

References

1. Adamides, E.D., Karacapilidis, N.: A knowledge centred framework for collaborative business process modelling. Business Process Management Journal 12(5), 557–575 (2006)
2. Bastien, J.M.C., Scapin, D.: Ergonomic Criteria for the Evaluation of Human-Computer interfaces. INRIA. Technical report No 156. Rocquencourt (1993)
3. Brandenburg, H., Wojtyna, J.P.: L'approche processus, Mode d'emploi, 2nd edn. Editions d'organisations, Eyrolles Group, Paris (2008)
4. Cardoso, E.C.S., Almeida, J.P.A., Guizzardi, G., Guizzardi, R.S.S.: Eliciting Goals for Business Process Models with non-Functional Requirements Catalogues. 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. 33–45. Springer, Heidelberg (2009)
5. Cattan, M., Idrissi, N., Knockaert, P., Vignon, D.: Maitriser les processus de l'entreprise: Guide opérationnel. Eyrolles Group (2006)
6. Cortes-Cornax, M., Matei, A., Letier, E., Dupuy-Chessa, S., Rieu, D.: Intentional Fragments: Bridging the Gap Between Organizational and Intentional Levels in Business Processes. In: Meersman, R., et al. (eds.) OTM 2012, Part I. LNCS, vol. 7565, pp. 110–127. Springer, Heidelberg (2012)
7. Daoudi, F., Nurcan, S.: A benchmarking framework for methods to design flexible business processes. Special Issue of the Software Process: Improvement and Practice Journal on Business Process Management (2007)
8. De la Vara González, J.L., Díaz, J.S.: Business process driven requirements engineering: A goal based approach. In: 8th Workshop, Business Process Modeling, Development, and Support (BPMDS 2007), Norway (2007)
9. De la Vara, J.L., Sánchez, J.: Improving requirements analysis through business process modelling: A participative approach. In: Abramowicz, W., Fensel, D. (eds.) BIS 2008. LNBIP, vol. 7, pp. 165–176. Springer, Heidelberg (2008)
10. Deneckère, R., Kornyshova, E., Rolland, C.: Enhancing the Guidance of the Intentional Model MAP: Graph Theory Application. RCIS (2009)
11. Ishikawa, K.: What is total quality control the Japanese way. Prentice-Hall, London (1985)
12. Koliadis, G., Ghose, A.K.: Relating business process models to goal-oriented requirements models in KAOS. In: Hoffmann, A., Kang, B.-H., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 25–39. Springer, Heidelberg (2006)

13. 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)
14. McConnell, J., Berkowitz, R.E.: The seven tools of TQC. Delaware Group (1989)
15. McQuater, R.E., Scurr, C.H., Dale, B.G., Hillman, P.G.: Using quality tools and techniques successfully. The TQM Magazine 7(6), 37–42 (1995)
16. Mandran, N., Dupuy-Chessa, S., Front, A., Rieu, D.: Démarche centrée utilisateur pour une ingénierie de langages de modélisation de qualité. Revue RSTI-ISI 18(3) (2013)
17. Moody, D.L.: The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE 35(6), 756–779 (2009)
18. Morrison, E., Ghose, A., Dam, H., Hinge, K., Hoesch-Klohe, K.: Strategic alignment of business processes (2011)
19. Neiger, D., Churilov, L.: Goal-Oriented Business Process Modeling with EPCs and Value-Focused Thinking. Business Process Management (2004)
20. Noyé, D.: L'amélioration participative des processus. Mouvement français pour la qualité. INSEP Consulting Éditions, 3rd edn. (2002) ISBN: 2-914006-05-5
21. Nurcan, S.: A survey on the flexibility requirements related to business processes and modeling artifacts. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 378. IEEE (2008)
22. Pavlovski, C., Zou, J.: Non-functional requirements in business process modeling. In: Proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling, vol. 79. Australian Computer Society (2008)
23. Pimsakul, S., Somsuk, N., Junboon, W., Laosirihongthong, T.: Production Process Improvement Using the Six Sigma DMAIC Methodology: A Case Study of a Laser Computer Mouse Production Process. In: The 19th International Conference on Industrial Engineering and Engineering Management, pp. 133–146 (2013)
24. Rolland, C., Prakash, N., Benjamen, A.: A Multi-Model View of Process Modelling. Requirements Engineering Journal, 169–187 (1999)
25. Santorum, M.: A Serious Game based Method for Business Process Management. In: 5th IEEE Int. Conference on Research Challenges in Information Science, RCIS 2011 (2011)
26. Santorum, M., Front, A., Rieu, D.: ISEAsy: A social business process management platform. In: 6th Workshop on Business Process Management and Social Software, BPMS2 2013 in Conjunction with BPM 2013, Beijing-China, pp. 26–30 (2013)
27. Sangiorgi, U., Vanderdonckt, J.: GAMBIT: Addressing multi-platform collaborative sketching with html5. In: Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 257–262. ACM (2012)
28. Thomas, P., Keller, P.: The Six Sigma Handbook, 3rd edn. McGraw-Hill, New York (2009) ISBN 0-07-162338-8
29. Van der Aalst, W.M.P., Weske, M., Wirtz, G.: Advanced Topics in Workflow Management: Issues, Requirements, and Solutions. Journal of Integrated Design and Process Science (2003)
30. Scheer, A.-W., Nüttgens, M.: ARIS architecture and reference models for business process management. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 376–389. Springer, Heidelberg (2000)
31. Becker, J., Pfeiffer, D., Räckers, M.: Domain Specific Process Modelling in Public Administrations – The PICTURE-Approach. In: Wimmer, M.A., Scholl, J., Grönlund, Å. (eds.) EGOV. LNCS, vol. 4656, pp. 68–79. Springer, Heidelberg (2007)

Modeling the Resource Perspective of Business Process Compliance Rules with the Extended Compliance Rule Graph*

Franziska Semmelrodt, David Knuplesch, and Manfred Reichert

Institute of Databases and Information Systems,
Ulm University, Germany
firstname.familyname@uni-ulm.de

Abstract. Process-aware information systems must ensure compliance of the business processes they implement with global compliance rules related to security constraints, domain-specific guidelines, standards, and laws. Usually, respective compliance rules cover multiple process perspectives; i.e., they not only deal with the control flow perspective that restricts the sequence in which the process activities shall be executed, but also refer to other process perspectives like data, time, and resource. Although there are various approaches for specifying compliance rules (e.g., based on temporal logic and narrative patterns), only few languages allow for the visual modeling of compliance rules. In turn, existing visual languages focus on the control flow perspective, but treat the other process perspectives as second class citizens. To remedy this drawback, this paper presents an approach for the visual modeling of business process compliance rules, including the resource perspective. The suitability of this approach is evaluated in a case study that was performed by business analysts in the healthcare domain.

1 Introduction

During the last decades many frameworks were proposed that aim to ensure the correctness of business process models. While early works focused on structural and behavioral model correctness (e.g., absence of deadlocks and livelocks) [1, 2], the semantic correctness of process models with imposed compliance rules (i.e., business process compliance) has been subject to recent work [3, 4, 29]. Compliance rules formally capture security constraints, domain-specific guidelines, corporate standards, and laws in a machine-readable manner. Besides control flow (i.e. sequence of activities), the resource perspective on business processes constitutes another fundamental aspect of business process compliance and respective rules (e.g. separation and binding of duties) [5, 6, 7].

For example, consider the compliance rules from Table 1. These refer to a woman's hospital [8, 9, 10, 11]. In particular, they highlight the need for covering

* This work was done within the research project C^{3}Pro funded by the German Research Foundation (DFG) under project number RE 1402/2-1.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 48–63, 2014.

Table 1. Healthcare compliance rules

C1	An X-ray examination for an inpatient must be ordered by a *ward physician*. In this context, the *same physician* must fill in an order form [10].
C2	An X-ray checkup in the *radiology department* must be performed by a *radiologist*. Prior to this, the informed consent of the patient must be checked by a *medical technical assistant (MTA)* of the *radiology department* [10].
C3	Diagnoses must be made by *ward physicians* after receiving the X-ray diagnosis and the X-ray images from the *secretary* of the *radiology department* [10].
C4	The central patient *admission* should admit a patient at the latest one week after she was referred to the hospital by a *gynecologist* [9].
C5	At least one day before a surgery takes place, blood bottles must be ordered by a *ward physician* of the *surgery ward* [9].
C6	Before a *physician* requests an informed consent (IC), the *same physician* must inform the patient about risks [8, 9, 10, 11].

the resource perspective in the context of business process compliance rules. On one hand, compliance rule C1 considers the resource perspective by requiring a performer with role *physician* assigned to the respective ward. On the other, C1 requires that both tasks (i.e., *order X-ray* and *fill order form*) are performed by the same person (i.e., binding of duties). C6 constitutes another example of such a binding of duties rule. In turn, the resource perspective related to compliance rule C2 requires performers having different roles, but being assigned to the same organizational unit. By contrast, C3 and C4 relate tasks to performers with different roles and organizational units. Altogether, the rules from Table 1 emphasize the high relevance of the resource perspective in business process compliance rules.

While there exist pattern-based approaches [12, 13] for modeling compliance rules that also cover the resource perspective, the latter has been neglected in the design of visual languages for modeling compliance rules so far. To remedy this drawback, this paper provides an approach for the visual modeling of compliance rules that covers the resource perspective as well. In particular, we will show how the resource perspective can be captured with the extended Compliance Rule Graph (eCRG) language. Further, we evaluate the applicability and expressiveness of the eCRG language in respect to the resource perspective in the context of a case study. In the latter we analyze various processes and related compliance rules from a woman's hospital.

Note that we have already introduced the fundamentals of the eCRG language in previous work [14]. However, [14] only briefly deals with the resource perspective of the eCRG as one out of multiple perspectives. By contrast, this paper provides the first detailed presentation of those eCRG elements covering the resource perspective. The remainder of this paper is structured as follows: Section 2 introduces fundamentals required for understanding this work. Section 3 discusses the eCRG based modeling of the resource perspective of business process compliance rules along examples. In particular, we first introduce a scenario referring to the organizational model of a woman's hospital. Second, we present the specific elements of the eCRG language for covering the resource perspective. Third, these eCRG elements are applied to model the rules from Table 1. Finally, results from a case study (i.e. evaluation) we conducted in the healthcare domain are discussed. Related work is presented in Section 4, while Section 5 concludes the paper.

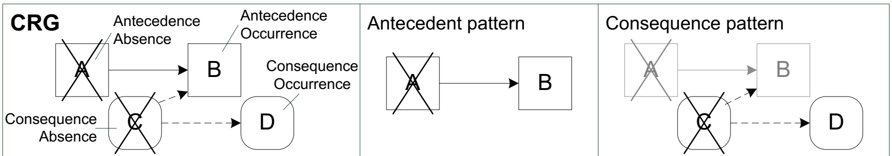
2 Backgrounds

This paper introduces the resource perspective of the extended Compliance Rule Graph (eCRG) modeling language. Since the eCRG language is based on the Compliance Rule Graph (CRG) language, we first introduce CRG and then present the fundamentals of the eCRG language.

2.1 Compliance Rule Graph

The Compliance Rule Graph (CRG) language allows for the visual modeling of compliance rules focusing on the control flow perspective (i.e. sequence flow) of business processes [15, 16, 17]. More precisely, a CRG constitutes an acyclic graph that consists of an *antecedence pattern* and one or several related *consequence patterns*. Both patterns are modeled using *occurrence* and *absence nodes*, which either express the occurrence or absence of events (e.g. related to the execution of a particular task). Edges between such nodes indicate control flow dependencies.

As illustrated in Fig. 1, a trace is considered as compliant with a CRG iff for each match of the antecedence pattern there is at least one corresponding match of every consequence pattern. Furthermore, a trace is considered as trivially compliant iff there is no match of the antecedence pattern. For example, the CRG from Fig. 2 expresses that for each B not preceded by an A, a D must occur, which is not preceded by any C that, in turn, precedes the respective B.



Example traces			
< E, **D**, F, G, **B** >	compliant	only match < E, D, F, G, **B** >	< E, **D**, F, G, B >
< **D**, F, **C**, E, **B** >	compliant	only match < D, F, C, E, **B** >	< **D**, F, C, E, B > (C is after D!)
< **A, B**, **C**, E, **D** >	trivially compliant	no match (A is before B!)	-
< **C**, F, **B**, G, E >	violation	only match < C, F, **B**, G, E >	no match (missing D)
< **B**, **C**, **D**, E, **B** >	violation	1st match < **B**, C, D, E, B >	< B, C, **D**, E, B >
		2nd match< B, C, D, E, **B** >	no match (C is before D!)

Fig. 1. CRG example and semantics [14]

2.2 Extended Compliance Rule Graph

The CRG language focuses on the **control flow perspective** of compliance rules, but factors out other perspectives. In [14], we introduced the extended Compliance Rule Graph (eCRG) as a visual language for modeling compliance rules that not only covers the control flow perspective, but provides integraed support for the resource, data, and time perspectives as well.

To enable such a support of multiple perspectives, the eCRG language allows for *attachments* in addition to nodes and connectors (i.e. edges). Respective *attachments* represent constraints of the nodes or edges they are linked to. Furthermore, an eCRG may contain instance nodes referring to particular objects, which exist independently from the respective rule (e.g. Mr. Smith, postnatal ward, physician). Note that instance nodes are neither part of the antecedence nor the consequence pattern. Fig. 2 provides an overview of eCRG elements.

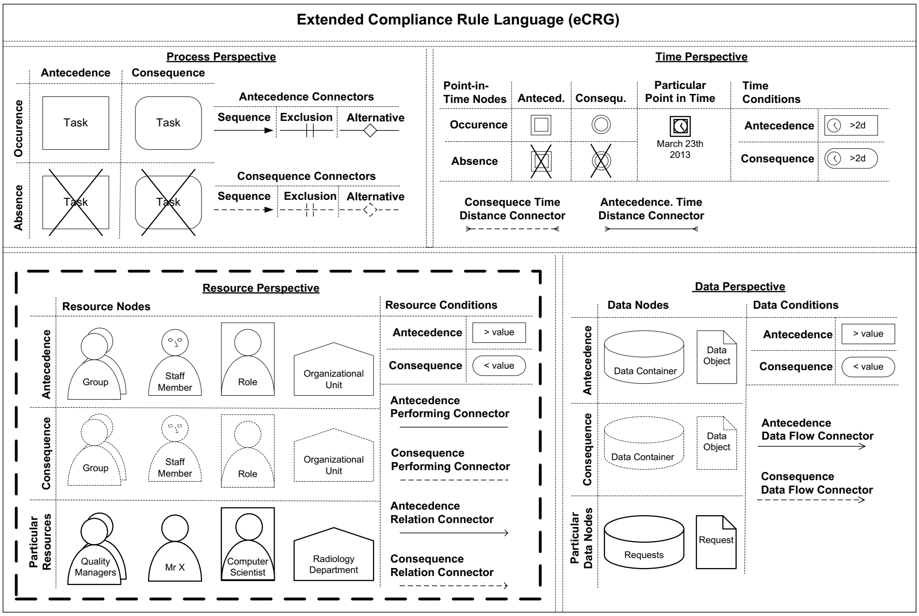


Fig. 2. Elements of the eCRG language

Control Flow Perspective. Modeling the control flow perspective of compliance rules is supported through four kinds of *task nodes*, i.e., antecedence occurrence, antecedence absence, consequence occurrence, and consequence absence task nodes. Based on these nodes it can be expressed whether or not particular tasks shall be executed. In addition, two kinds of *sequence flow connectors* are provided that allow constraining the execution sequence of tasks. Note that the absence of a sequence flow indicates parallel flow. Furthermore, *exclusive connectors* express mutual exclusion of the tasks they refer to. Finally, *alternative connectors* express that at least one of the connected tasks must occur [14].

Time Perspective. The eCRG language offers the following elements for modeling the time perspective: *Point-in-time nodes*, *time condition attachments*, and *time distance connectors* (cf. Fig. 2). Like task nodes, *point-in-time nodes* can be either antecedence occurrence, antecedence absence, consequence occurrence, or consequence absence nodes. Furthermore, a particular date or point in time (e.g. 26th October 2014) can be expressed using instance nodes. *Time conditions* may be attached to task nodes and sequence flow connectors to constrain the duration of a task or the time distance between *task nodes* and *point-in-time nodes.* Finally, *time distance connectors* allow constraining the time distance without implying a particular sequence.

Data Perspective. *Data container nodes* and *data object nodes* support the modeling of the data perspective in eCRGs. Furthermore, *data flow connectors* and *data conditions* are provided. *Data container nodes* refer to process data elements or global data stores. By contrast, *data object nodes* refer to particular data values and data object instances. Both kinds of data nodes may be part of the antecedence or consequence pattern, or represent a particular data container and data object respectively. *Data flow connectors* define which process tasks read or write which data objects or data containers. To constrain data containers, data objects and data flow, *data conditions* may be attached. Finally, *data relation connectors* may either be used to compare different data objects or to constrain the value of data containers at particular points in time.

Resource Perspective. For modeling the resource perspective of compliance rules *resource nodes* are provided, i.e., staff member, role, group, and organizational unit nodes. Similar to task nodes, *resource nodes* may be part of the antecedence or consequence pattern. Alternatively, they may represent a particular resource instance (e.g. Mr. Smith, postnatal ward, physician). To specify dependencies among resources, *resource relation connectors* are provided. In turn, *resource conditions* constrain a particular resource node. Finally, the *performing relation* indicates the performer of a task node. This paper focuses on the resource perspective of process compliance rules. Respective elements are therefore described in more detail in Section 3.

3 The Resource Perspective of Compliance Rules

After having introduced the fundamentals of the eCRG and CRG languages, we discuss how the resource perspective of business process compliance rules can be modeled when using eCRG. For this purpose, we first provide an examplary application scenario from a woman's hospital. This scenario is then used to illustrate the resource perspective of the eCRG language.

3.1 Scenario

This section illustrates the resource perspective along a healthcare scenario, which refers to clinical processes from the woman's hospital. Fig. 3 illustrates our resource meta-model. It comprises the entity types *organizational unit*, *group*, *staff member*, and *role* as well as the relation types between them. However, our approach is not restricted to the entity and relation types from Fig. 3. For example, our scenario refers to additional relation types and properties (e.g. relation type *supervisor*, property '*is surgery ward*') as well.

Fig. 4 shows the organizational units relevant in the context of our scenario. On one hand, these units are subordinated ones of the *university hospital* including the *woman's hospital* with its wards (e.g. *postnatal ward 1/2*) and other units (e.g. *admission*) as well as the *radiology department*. On the other hand, Fig. 4 further refers to external medical practices of a *gynecologist* and a *general practitioner*.

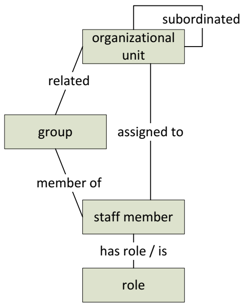


Fig. 3. Meta-model

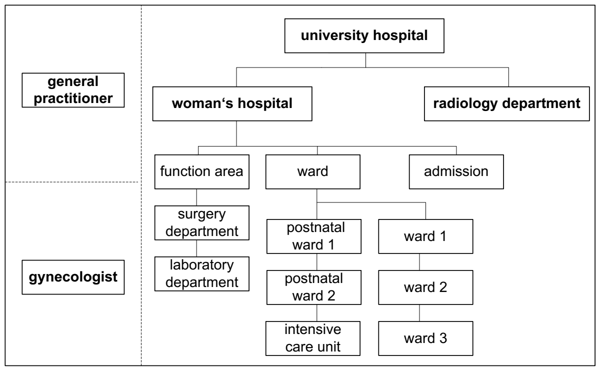


Fig. 4. Organizational units

Fig. 5 provides the assignment relation (cf. Fig. 3) of an anonymized extract of the staff database relevant in our scenario. The roles of the respective actors are shown in Fig. 6. For example, *Mrs. A*, *Mr. B*, and *Mr. C* are assigned to the *radiology department*, while *Mrs. E* is assigned to *wards 1* and *2* (cf. Fig. 5). In turn, *Mr. B* and *Mrs. E* are both physicians, while *Mrs. A* has role *MTA* (i.e., medical technical assistant).

To complement our scenario, Fig. 7 specifies the relation *supervisor*. For instance, *Mr. B* is supervisor of *Mrs. A*. In turn, Fig. 8 provides two attributes of the aforementioned wards; i.e., capacities and information on whether or not the ward is a *surgery ward*.

Finally, Fig. 9 shows a possible execution log of a healthcare process from our scenario [10].

assigned	gynecologist	general practitioner	university hospital									
			woman's hospital									
				function area		ward						radiology department
			admission	surgery department	laboratory department	postnatal ward 1	postnatal ward 2	intensive care unit	ward 1	ward 2	ward 3	
⋮												
Mrs. A												X
Mr. B												X
Mr. C												X
⋮												
Mr. D			X									
⋮												
Mrs. E									X	X		
Mr. F						X	X					
Mrs. G									X	X		
Mr. H									X		X	
⋮												
Mr. I		X										
⋮												

Fig. 5. Staff members and relation assignment

3.2 Resource Perspective in Detail

As outlined in Sect. 2, the resource perspective of the eCRG language provides elements referring to *organizational units*, *groups*, *roles*, and *staff members*. In turn, these may either be part of the antecedence pattern (solid) or consequence pattern (dashed), or be a particular instance (bold) (cf. Fig 2). The *performing relation connector* allows using these elements in order to specify the performers of both antecedence and consequence task nodes in detail. Accordingly, the performing relation connector can either be antecedence (solid) or consequence (dashed). Fig. 10 illustrates the application of the performing relation connector and its semantics in detail. In Fig. 10a, the antecedence performing relation is used to connect antecedence tasks with an antecedence staff member. In turn, Fig. 10b shows a consequence performing relation connecting an antecedence task with an antecedence staff member. In Fig. 10c, two consequence performing relations are used to connect both antecedence tasks with a consequence staff member. Note that the eCRGs from Figs. 10b and 10c have the same meaning. Fig. 10d shows how a consequence task can be connected to an antecedence task by using a consequence performing relation, while Fig. 10e shows how the consequence performing relation connects two consequence tasks with the same consequence staff member. Note that antecedence performing relation connectors must not be connected to any element of the consequence pattern.

has role / is	...	physician	nurse	secretary	MTA	...
⋮						
Mrs. A					X	
Mrs. B		X				
Mr. C				X		
⋮						
Mr. D		X				
⋮						
Mrs. E		X				
Mr. F		X				
Mr. G			X			
Mr. H			X			
⋮						
Mr. I		X				
⋮						

Fig. 6. Staff members and their roles

is supervisor of	...	Mrs. A	Mr. B	Mr. C	...
⋮					
Mrs. A					
Mr. B		X	X	X	
Mr. C					
⋮					

is supervisor of	...	Mrs. E	Mr. F	Mrs. G	Mr. H	...
⋮						
Mrs. E				X	X	
Mr. F						
Mrs. G						
Mr. H						
⋮						

Fig. 7. Relation supervisor

As indicated by the examples from Table 1, compliance rules refer to relations between different elements of the resource perspective. The *resource relation connector* can specify relations between resources in the antecedence as well as the consequence pattern. Accordingly, each resource relation connector is either part of the antecedence pattern (solid) or the consequence pattern (dashed). The corresponding resource relation can be expressed by attaching rectangles in case of antecedence relation connectors and ovals in case of consequence relation connectors. Fig. 11 shows the use of the resource relation connector and its semantics in more detail. Fig. 11a uses antecedence resource relations to connect antecedence staff members with an antecedence organizational unit. In turn, Fig. 11b illustrates an antecedence and a consequence resource relation both connecting an antecedence staff member with an antecedence organizational unit. Fig. 11c comprises a consequence resource relation that connects antecedence and

attributes	ward					
	postnatal ward 1	postnatal ward 2	intensive care unit	ward 1	ward 2	ward 3
⋮						
is surgery ward			X	X		
capcity	20	20	15	30	30	20
⋮						

Fig. 8. Ward attributes

step	date	time	activity	performer	data/documents
1	05.02.2009	09:20	examine patient	Mrs. E	
2	05.02.2009	09:40	order X-ray	Mrs. E	
3	05.02.2009	09:45	fill request form	Mrs. E	request form
4	05.02.2009	09:50	inform patient	Mrs. E	
5	05.02.2009	09:55	answer questions	Mrs. E	
6	05.02.2009	09:58	request IC	Mrs. E	signed IC
7	05.02.2009	10:10	transfer patient	Mrs. G	
8	05.02.2009	10:12	transmit IC	Mr. H	signed IC
9	05.02.2009	10:45	check IC	Mrs. A	signed IC
10	05.02.2009	10:50	prepare patient	Mrs. A	
11	05.02.2009	11:05	perfom X-ray	Mr. B	X-ray image
12	05.02.2009	11:20	transfer patient	Mrs. G	
13	05.02.2009	11:35	document result	Mr. C	X-ray diagnosis
14	05.02.2009	11:45	transmit X-ray image & X-ray diagnosis	Mr. C	X-ray image & diagnosis
15	05.02.2009	14:10	make diagnosis	Mrs. E	X-ray image & diagnosis
16	05.02.2009	14:15	prescribe therapy	Mrs. E	
17	05.02.2009	14:40	document diagnosis and therapy	Mr H	

Fig. 9. Execution log of radiology process

consequence staff members, while an antecedence relation connects the same antecedence staff member with resource *physician*. In turn, Fig. 11d applys a consequence relation connector to refer from the staff member to resource *physician*. Finally, Figs. 11e and 11f show how the performing relation can implicitly incorporate the assignment relation and the role relation of our meta-model in some special cases (cf. Fig. 3). Note that antecedence resource relation connectors can only connect elements of the antecedence pattern, but must not be connected to any consequence resource.

Finally, resource conditions may be attached to the elements of the resource perspective. Resource conditions may either be part of the antecedence (rectangle) or consequence pattern (oval). Their semantics is illustrated in Fig. 12. In particular, Fig. 12a shows the use of an antecedence condition constraining an antecedence organizational unit. In turn, Fig. 12b applies a consequence condition to the same antecedence organizational unit, while in Fig. 12c a consequence organizational unit is used. Despite this difference, Figs. 12b and 12c have the same meaning. The meaning of Fig. 12d changes, when turning the antecedence organizational unit into a consequence organizational unit. Note that

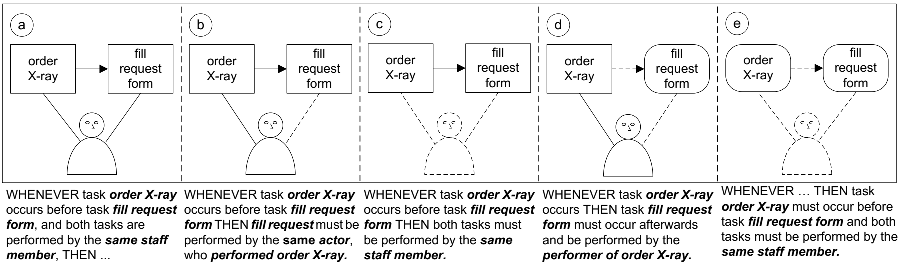


Fig. 10. Performing relation

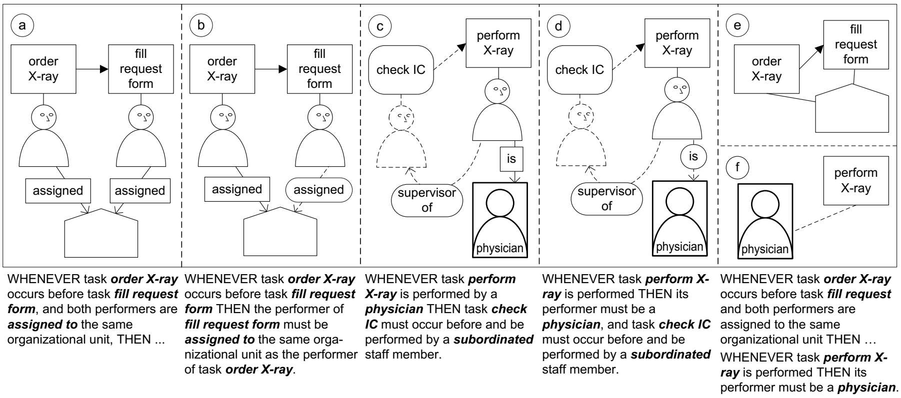


Fig. 11. Resource relations

antecedence resource conditions may only be attached to antecedence resource nodes, but not to elements of the consequence pattern.

A simple formal specification of the eCRG language, including the resource perspective, is provided in [18].

In Fig. 13, the six compliance rules from Table 1 are visualized using the eCRG language, including its elements for capturing the resource perspective. Note that the execution log from Fig. 9 complies with each of these eCRGs. Trivially, the log complies with rules C4 and C5, since it does not contain any of the tasks *refer patient* and *surgery*; i.e., there is no match of the antecedence patterns of rules C4 and C5. In turn, there exist matches for the antecedence patterns of rules C1, C2, C3, and C6 as well as the corresponding consequence patterns. Step 2 (i.e. *order X-ray*) matches with the antecedence pattern of C1, while the following Step 3 matches with the consequence pattern of C1 since it refers to task *fill request form* and is performed by the same staff member *Mrs. E.* As required by the consequence pattern, *Mrs. E* has role *physician* and is assigned to organizational unit *ward*. The antecedence pattern of C2 matches with Step 11 (i.e. *perform X-ray*). As specified in the consequence pattern of C2, the performer of Step 11 (i.e. *Mr. B*) is assigned to the *radiology department*. Further, this

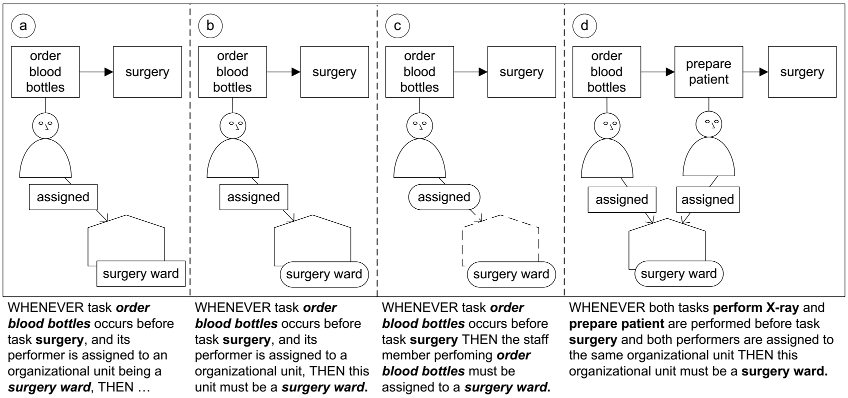


Fig. 12. Resource conditions

performer has role *physician*. The consequence pattern is completed by Step 9 (i.e. *check IC*), which is performed by *Mrs. A* with role *medical technical assistant (MTA)*. Furthermore, *Mrs. A* is assigned to unit *radiology department*. Step 15 (i.e. *make diagnosis*) triggers C3; i.e., it matches with the antecedence pattern of C3. As required by the consequence pattern of C3, the performer (i.e. *Mrs. E*) of Step 15 has role *physician* and is assigned to the unit *ward*. Furthermore, Step 15 is preceded by Step 14 (i.e. *transmit X-ray image & X-ray diagnosis*), which is performed by *Mr. C*. The latter is a secretary of the *radiology department*. Finally, the antecedence pattern of C6 matches with Step 6 (i.e. *request IC*). Step 4 (i.e. *inform patient*) satisfies the corresponding consequence pattern. Hence it is performed by the same staff member (i.e. *Mrs. E*), who also possesses role *physician*.

3.3 Evaluation

To evaluate the suitability of the eCRG language with respect to the modeling of the resource perspective, we conducted a case study in the healthcare domain. In particular, business analysts (i.e., non-IT-specialists) analyzed six process model collections stemming from the woman's hospital [8, 9, 10, 11]. Altogether, they identified 30 compliance rules and modeled them using the eCRG language. Out of these 30 compliance rules, 17 rules refer to the resource perspective. For these 17 compliance rules, the business analysts were able to capture the resource perspective with eCRG; i.e., the eCRG language allowed them to capture all relevant aspects of the resource perspective. Besides this, they revealed drawbacks regarding the modeling of the control flow and time perspectives. In particular, the business analysts emphasized the missing support for periodic time events and the missing ability to refine tasks. Table 2 summarizes study results.

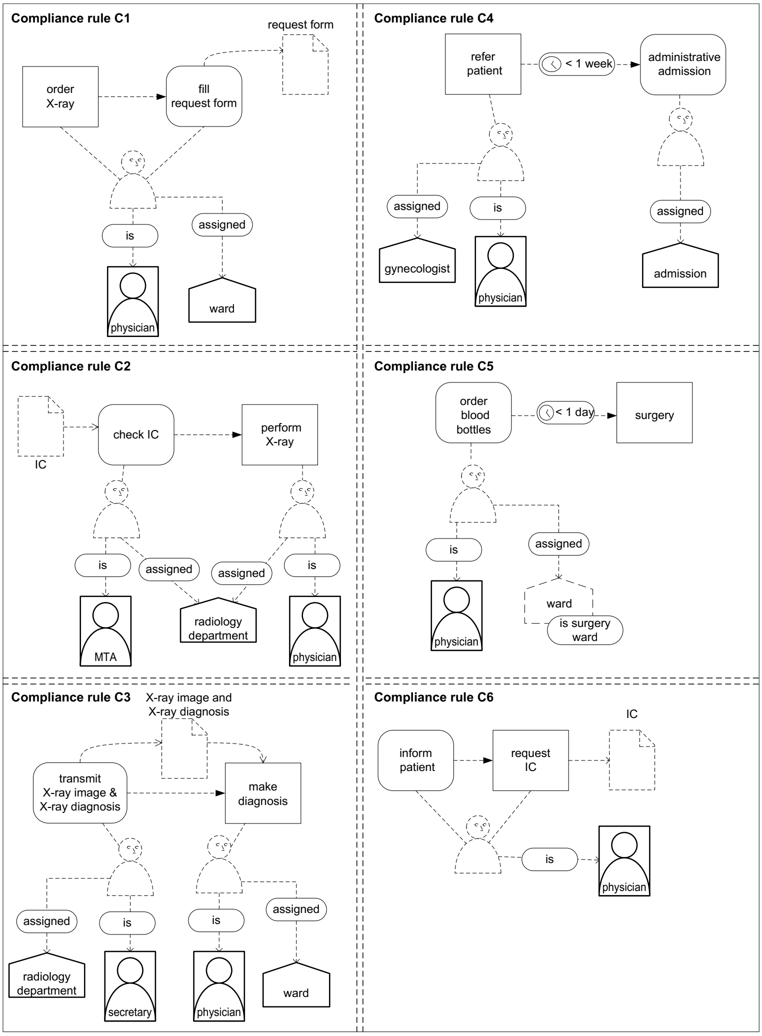


Fig. 13. Healthcare compliance rules

4 Related Work

Modeling issues related to the resource perspective of business processes are addressed in [19]. In turn, [20, 21, 22, 23, 24, 25] discuss the interaction, time, and data perspectives of business processes.

The integration of business process compliance throughout the entire process lifecycle is investigated in [6, 17, 26]; [27] examines compliance issues in the context of cross-organizational processes developing a logic-based formalism for describing both the semantics of normative specifications and compliance check-

Table 2. Evaluation of the eCRG language

Perspective	Status
Control flow perspective	black box character of Tasks
Data perspective	✓
Resource perspective	✓
Time perspective	periodical points in time

ing procedures. In turn, [28] introduces a semantic layer that interprets process instances according to an independently designed set of internal controls.

To verify whether compliance rules are fulfilled by process models at design time, many approaches apply *model checking* [4, 29, 30, 31, 32]; some of them address the data and time perspectives as well. [12] uses *alignments* to detect compliance violations in process logs. Other approaches for verifying compliance apply the notion of *semantic congruence* [33], use *petri-nets* [34], or rely on *mixed-integer programming* [35]. In turn, [36, 37, 38] deal with the compliance of interaction models and cross-organizational process collaborations. Finally, there exist visual approaches for compliance rule modeling [4, 15, 31, 39, 40]. As opposed to eCRG, they focus on the control flow and - partly - the data perspective, but factor out the resource perspective.

5 Summary and Outlook

While compliance rule modeling has been addressed by a plethora of approaches, the visual modeling of the data, time, and resource perspectives has not been sufficiently addressed yet [5, 12, 13]. To remedy this drawback, this paper introduces an extension of the compliance rule graph (CRG) language [15, 16, 17] in order to cover the resource perspective in visual compliance rules as well. Each language element has been presented in detail and illustrated along an example. In turn, all examples were gathered in a healthcare case study that was performed by business analysts. This case study further contributes to evaluate our approach, proving its suitability for modeling the resource perspective of business process compliance rules.

To enable tool support for both the modeling and verification of compliance rules, the semantics of the introduced visual compliance rule language has been formalized in a technical report [18].

In a next step, we will consider the feedback we gathered in the case study in order to enhance the visual compliance rule language. Furthermore, we are developing techniques for verifying the compliance of business processes with imposed multi-perspective compliance rules during runtime. However, our overall aim is to ensure multi-perspective compliance for all phases of the process life cycle, including *a priori* compliance checking at design time as well as *a posteriori* compliance checking after process execution. Finally, we will consider compliance checking in the context of process changes.

References

1. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
2. Reichert, M., Dadam, P.: ADEPT$_{flex}$ – supporting dynamic changes of workflows without losing control. Intelligent Inf. Sys. 10(2), 93–129 (1998)
3. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: EDOC 2006, pp. 221–232 (2006)
4. Awad, A., Weidlich, M., Weske, M.: Specification, verification and explanation of violation for data aware compliance rules. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 500–515. Springer, Heidelberg (2009)
5. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Hints on how to face business process compliance. In: JISBD 2010 (2010)
6. Ramezani, E., Fahland, D., van der Werf, J.M., Mattheis, P.: Separating compliance management and business process management. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBIP, vol. 100, pp. 459–464. Springer, Heidelberg (2012)
7. Schaad, A., Spadone, P., Weichsel, H.: A case study of separation of duty properties in the context of the austrian "elaw" process. In: SAC 2005, pp. 1328–1332. ACM (2005)
8. Konyen, I., Schultheiß, B., Reichert, M.: Prozessentwurf eines Ablaufs im Labor. Technical Report DBIS-16, University of Ulm (1996)
9. Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T., Reichert, M., Dadam, P., Kreienberg, R.: Prozessentwurf am Beispiel eines Ablaufs aus dem OP-Bereich - Ergebnisse einer Analyse an der Universitätsfrauenklinik Ulm. Technical Report DBIS-6, University of Ulm (1996)
10. Konyen, I., Schultheiß, B., Reichert, M.: Prozessentwurf für den Ablauf einer radiologischen Untersuchung. Technical Report DBIS-15, University of Ulm (1996)
11. Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T., Reichert, M., Dadam, P., Kreienberg, R.: Prozessentwurf für den Ablauf einer ambulanten Chemotherapie. Technical Report DBIS-7, University of Ulm (1996)
12. Ramezani, E., Fahland, D., van der Aalst, W.M.P.: Where did I misbehave? Diagnostic information in compliance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 262–278. Springer, Heidelberg (2012)
13. Turetken, O., Elgammal, A., van den Heuvel, W.J., Papazoglou, M.: Capturing compliance requirements: A pattern-based approach. IEEE Soft., 29–36 (2012)
14. Knuplesch, D., Reichert, M., Ly, L.T., Kumar, A., Rinderle-Ma, S.: Visual modeling of business process compliance rules with the support of multiple perspectives. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 106–120. Springer, Heidelberg (2013)
15. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 9–23. Springer, Heidelberg (2010)
16. Ly, L.T., Rinderle-Ma, S., Knuplesch, D., Dadam, P.: Monitoring business process compliance using compliance rule graphs. In: CoopIS 2011, pp. 82–99 (2011)
17. Knuplesch, D., Reichert, M.: Ensuring business process compliance along the process life cycle. Technical Report 2011-06, Ulm University (2011)
18. Knuplesch, D., Reichert, M., Ly, L.T., Kumar, A., Rinderle-Ma, S.: On the formal semantics of the extended compliance rule graph. Technical Report 2013-05, Ulm University (2013)

19. 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)
20. Kumar, A., Wang, J.: A framework for document-driven workflow systems. In: Int'l Handbook on Business Process Management, pp. 419–440. Springer (2010)
21. Eder, J., Tahamtan, A.: Temporal conformance of federated choreographies. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 668–675. Springer, Heidelberg (2008)
22. Lanz, A., Weber, B., Reichert, M.: Time patterns for process-aware information systems. In: Requirements Engineering (2012)
23. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf. Sys. 35(8) (2010)
24. Barros, A., Dumas, M., ter Hofstede, A.H.M.: Service interaction patterns. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 302–318. Springer, Heidelberg (2005)
25. Knuplesch, D., Pryss, R., Reichert, M.: Data-aware interaction in distributed and collaborative workflows: Modeling, semantics, correctness. In: CollaborateCom 2012, pp. 223–232. IEEE (2012)
26. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems. Inf. Sys. Front. 14(2), 195–219 (2012)
27. Governatori, G., Sadiq, S.: The journey to business process compliance. In: Handbook of Research on BPM, pp. 426–454. IGI Global (2009)
28. Namiri, K., Stojanovic, N.: Pattern-Based design and validation of business process compliance. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 59–76. Springer, Heidelberg (2007)
29. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 332–346. Springer, Heidelberg (2010)
30. Ghose, A.K., Koliadis, G.: Auditing business process compliance. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 169–180. Springer, Heidelberg (2007)
31. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal 46(2), 335–361 (2007)
32. Kokash, N., Krause, C., de Vink, E.: Time and data aware analysis of graphical service models. In: SEFM 2010 (2010)
33. Höhn, S.: Model-based reasoning on the achievement of business goals. In: SAC 2009, pp. 1589–1593. ACM, New York (2009)
34. Accorsi, R., Lowis, L., Sato, Y.: Automated certification for compliant cloud-based business processes. Business & Inf. Sys. Engineering 3(3), 145–154 (2011)
35. Kumar, A., Yao, W., Chu, C.: Flexible process compliance with semantic constraints using mixed-integer programming. INFORMS J. on Comp. (2012)
36. Knuplesch, D., Reichert, M., Mangler, J., Rinderle-Ma, S., Fdhila, W.: Towards compliance of cross-organizational processes and their changes. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 649–661. Springer, Heidelberg (2013)
37. Knuplesch, D., Reichert, M., Fdhila, W., Rinderle-Ma, S.: On enabling compliance of cross-organizational business processes. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 146–154. Springer, Heidelberg (2013)

38. Knuplesch, D., Reichert, M., Pryss, R., Fdhila, W., Rinderle-Ma, S.: Ensuring compliance of distributed and collaborative workflows. In: CollaborateCom 2013, pp. 133–142. IEEE (2013)
39. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. Vis. Lang. Comp. 22(1), 30–55 (2011)
40. Feja, S., Speck, A., Witt, S., Schulz, M.: Checkable graphical business process representation. In: Catania, B., Ivanović, M., Thalheim, B. (eds.) ADBIS 2010. LNCS, vol. 6295, pp. 176–189. Springer, Heidelberg (2010)

Addressing the Paradigmatic Limitation of Conventional Business Process Management Concepts by Proposing New Definitions

Matin Mavaddat, Stewart Green, and Jin Sa

University of the West of England

Abstract. Considering the history of the formation of the business process management discipline and its concept definitions, and by looking at organisations as social systems, it can be demonstrated that conventional business process management practices can be associated with the functionalist social paradigm and therefore are only applicable in unitary problem contexts. Participants in unitary problem contexts have similar values, beliefs and interests, share common goals and objectives and are all involved in decision-making about how to achieve the common goals and objectives. It can be argued that this problem context covers only a very small percentage of the problems that an organisation is concerned with and that this inherent paradigmatic limitation in the current definitions of business process management concepts causes the outcomes of the BPM practices based on them to be unrealistic, incomplete and even at points misleading. To address this paradigmatic limitation this paper proposes new definitions for BPM's main concepts to reduce its tight coupling with the unitary problem context and make it more applicable in pluralist and coercive problem contexts and therefore closer in its outcomes to the reality of the organisation.

Keywords: Business process management, Organisation as a social system, Business process realisation, Dependency cycle, Conversation log.

1 Introduction

Understanding the business processes that are being followed in big organizations is very hard if not impossible. Van der Aalst [1] in his paper "Challenges in Business Process Analysis" mentions: "reality is often very different from what is modelled or what people think" and also "as long as managers and system designers take a power point reality as a starting point, information systems will remain to have serious alignment problems".

A great percentage of Business Process Re-engineering or Business Process Management projects fail and many Business Process Support Systems have great alignment problems with the ways people actually carry out their work [1], [2], [18]. The researcher's experience shows that this misalignment is to such an extent that people usually stop using such support systems and start applying workarounds [18].

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 64–78, 2014.

By analysing the history of the formation of BPM and by considering its key concepts and definitions, this paper tries to demonstrate that the conventional BPM concepts and definitions are bound to the functionalist social paradigm whose only objective is the survival of the organization through ensuring its efficiency and adaptability like a machine and hence should only be used in unitary problem contexts. This means conventional BPM concepts assume that members of the organization have similar beliefs and interests, they share common goals and objectives and they have all been involved in the decision-making [14] (this will be further discussed in the next sections).

This paradigmatic limitation in the definitions of some of BPM's key concepts has caused its tools and techniques to be less effective in organizations, which are not bound to the functionalist social paradigm. These organizations should be analysed from different perspectives, using different social paradigms and metaphors and their problems should be considered to span across different problem contexts such as pluralist and coercive as well as unitary problem contexts[13], [18].

Applying BPM techniques that suffer from this paradigmatic limitation in organizations whose business processes belong not just to the unitary problem context may result in incomplete and at points misleading outcomes and ineffective and unusable process support systems [18].

In the literature, many scholars have identified the problems of incompleteness and ineffectiveness [1, 2, 3, 4], [9], [29, 30] and have suggested different approaches, methodologies and techniques to address them, but none of them has investigated the business process management discipline's shortcomings by focusing on its paradigmatic limitations.

This paper is structured as follows. Section 2 reviews the literature related to the conventional business process management key concepts (2.1) and the history of the formation of the BPM discipline (2.2) and introduces different sociological paradigms and their associated problem contexts (2.3). Section 2 lays the foundations for section 3 in which the reasons why the conventional BPM concepts are bound to the unitary problem context are discussed using its concept definitions and the history of formation. Section 4 redefines the key BPM concepts to also support pluralist and coercive problem contexts; and in section 5 these proposed definitions are illustrated by using a synthesised example. Finally, section 6 concludes the paper by summarising the problem and the proposed solution.

2 Literature Review

In this section, the literature related to a number of key concepts in business process management and organizational systems analysis and the history of the formation of the BPM discipline is briefly reviewed. Using these concepts and definitions and the history of the BPM discipline formation, in section 3 it will be demonstrated why the researcher believes that current BPM tools and techniques can only be applied in unitary problem contexts.

2.1 Business Process Management Concepts and Definitions

Weske [25] defines business process management as "concepts, methods and techniques to support the design, administration, configuration, enactment and analysis of the business processes". He defines a business process as: "A set of activities that are performed in an organisational and technical environment to jointly realise a business goal". He outlines the most important goal of BPM is achieving a "better understanding of the operations that a company performs and its relationships". He also believes the main BPM goals are the following:

- Increasing the flexibility of the organisation to respond to change
- Creating a repository of business processes as a valuable asset
- Continuous process improvement, and
- Narrowing the gap between business processes and their realisation using software systems.

Dumas, La Rosa, Mendling, et al., [11] define BPM as: "the art and science of overseeing how work is performed in an organisation to ensure consistent outcomes and to take advantage of improvement opportunities". They state that business processes are "what companies do whenever they deliver a service or a product to customers". They believe business processes are built of activities, events, decision points, actors and one or several outcomes. Using these ingredients they define business processes as: "a collection of interrelated events, activities and decision points that involve a number of actors and objects, and that collectively lead to an outcome that is of value to at least one customer".

Using the above business process definitions Dumas, La Rosa, Mendling, et al. [11], define BPM as: "a body of methods, techniques and tools to discover, analyse, redesign, execute and monitor business processes". They also mention that one of the main characteristics of BPM is having business processes and business process models at its core and its emphasis on using business process models in all the different stages of the BPM lifecycle.

2.2 The History of Business Process Management Discipline Formation

Tom Davenport in the foreword of the book "Business Process Management, practical guidelines to successful implementations" [15] dates back the start of process thinking to Frederick Taylor and his colleagues whom at the turn of the last century developed modern industrial engineering and process improvement. This technique was limited to manual labour and production processes. Tom Davenport believes the next great improvement to BPM has been the addition of statistical process control to Taylorist approach by Shewart, Deming, Juran and others. Their version of process management involved:

- Measuring and limiting process variation
- Continuous improvement
- Giving the workers the power to improve their own work processes.

Companies, such as TOYOTA, created "Continuous process improvement" and "Total Quality Management" based on statistical principles. Lean techniques that many American firms have recently adopted are based on a less strict Toyota Production System (TPS).

Process re-engineering, which was the next major variation on business process management, was introduced in the 1990s when the Western economy was experiencing a recession. BPR added the following ideas to the process management concepts:

- Radical redesign and improvement of processes
- Consideration of cross-functional business processes
- Use of IT as an enabler for realisation of some parts of the business processes

BPR was also the first BPM movement to focus on non-production processes.

Dumas, La Rosa, Mendling, et al., [11] and Harmon [12] also introduce the origins and history of BPM and show that BPM concepts have emerged through the rise of functional organisations that had the specialist workers, who had their focus on a single part of the product or process, at their core. They demonstrate that the shortcomings of functional optimisation in functional organisations leads to the birth of process thinking and then later on to business process re-engineering. It can be inferred that they also believe BPM has been developed from the lessons learned from BPR project failures and also its relevant technological advancement in IT.

2.3 Sociological Paradigms and their Associated System Metaphors and Their Applicability in Different Problem Contexts

In this section different problem contexts for system analysis and their associated sociological paradigms will be briefly introduced.

Jackson & Keys [13, 14] introduced the "ideal-type" grid of problem contexts. This grid has been established in two dimensions: "systems" and "participants" as can be seen in Figure 1.

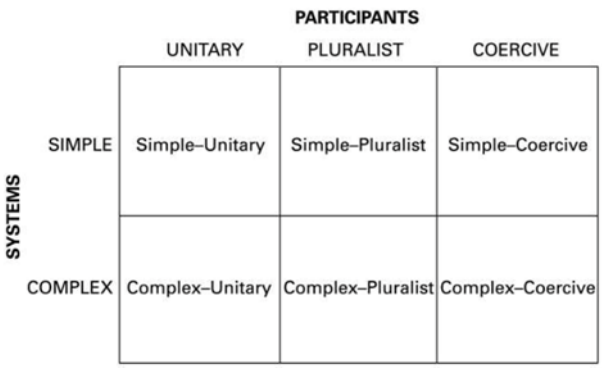


Fig. 1. "Ideal-type" grid of problem contexts cited in Jack-son (2003)

Systems have been divided into two categories - simple and complex; and Participants into three categories - unitary, pluralist and coercive.

Simple systems can be characterised as follows:

- They have a few subsystems.
- They have highly structured interactions.
- They do not change much over time.
- They are comparatively unaffected by the actions of their surrounding environment or their parts.

Complex systems can be characterised as follows:

- They have a large number of subsystems.
- They are involved in many loosely structured interactions.
- They are adaptive and evolutionary.
- They have purposeful parts and a chaotic environment that affect the system.

The horizontal axis categorises the relationship of the system participants. Participants whose relationships have been categorised as unitary:

- Have similar values, beliefs and interests
- Share common goals and objectives
- Are all involved in decision-making about how to achieve the common goals and objectives.

For participants whose relationships have been categorised as pluralist:

- Their basic interests are compatible.
- They don't share the same values and beliefs.
- If space has been made available within which they can have debates, arguments, conflict and disagreement and they feel that they have been involved in the decision-making, then accommodation and compromises can be found and they are happy to agree on productive ways forward towards the temporarily agreed goals and objectives.

Participants whose relationships have been categorised as coercive:

- Have few common interests
- Are unable to reach compromises
- Have no agreed common goals
- Take decisions based on the power structure and distribution.

Jackson [13] using the work of Burrell & Morgan [7] and Alvesson & Deetz [5], introduces the following four sociological paradigms and classifies the systems thinking approaches by those paradigms.

1. The functionalist paradigm: This paradigm gets its name from the fact that it wants to ensure that the system is efficient, adaptable, does what it needs to do to achieve its objectives, and as a consequence survives. Within this paradigm, the constituent elements of a system, their relations and the relation of the system with its environment get studied.
2. The interpretive paradigm: This paradigm has been built on the assumption that social systems are created from different interpretations of people about the situations they are in. These interpretations are causes to act and interact, and also pursue the purposes that have been originated from these interpretations.
3. The emancipatory paradigm: This paradigm's main concern is discrimination in all its shapes and forms, and emancipation of oppressed individuals and groups.
4. The postmodern paradigm: This paradigm opposes the rationality that is being sought by the other three paradigms in the organisation. It believes organisations are far too complex to be understood by any other paradigms. It emphasises on having fun, bringing the conflicts to the surface, and encouraging variety and diversity.

By using the system metaphors introduced by Burrell & Morgan [7] - organisations as machines, living organisms, brains, flux and transformation, cultures, political systems, psychic prisons, instruments of domination and organisations as carnivals - Jackson [13] shows that concepts and methodologies that are associated with the functionalist paradigm (and as a result associated with machines, living organisms, brains, flux and transformation system metaphors) are only applicable in the unitary problem contexts that has the characteristics that were introduced previously.

In the next section, using the history of BPM and by referring to the current BPM concepts and definition, it will be shown that conventional BPM practices can be associated with the functionalist social paradigm and therefore can only be applied to unitary problem contexts. It will also be discussed that phenomena in most contemporary organizations are not bound to unitary problem context and as a result this limitation causes the outcomes of the current BPM practices to be incomplete.

3 Business Process Management and the Functionalist Paradigm

Having in mind the history and origin of the BPM and looking at its main goals and objectives and also characteristics and concept definitions, it can be established that the current BPM concept definitions, tools and techniques look at an organisation from a functionalist sociological paradigm and that is considered here a paradigmatic limitation. Reviewing the recent literature such as [11], [10], [12], [25] from the key academics in the field of BPM better illustrates this paradigmatic limitation.

Looking at the history of the constitution of BPM concepts and techniques it seems that they have emerged through the rise of functional organisations that had the specialist workers, who had their focus on a single part of the product or process, at their core. The functionalist sociological paradigm is mainly concerned with a system's efficiency and adaptability [13], which can be considered the main goals of the BPM as well [25], [11], [12]. The conventional BPM goals and objectives were discussed in the previous section (2.2).

The quotations, statements and discussions below better illustrate the viewpoint of the current BPM theorists and better demonstrate that they are still working in the unitary problem context.

As was mentioned before, concepts that can be associated with the functionalist social paradigm can only be applied in unitary problem contexts, as a result, the characteristics of relationships in a unitary problem context are prevalent in the current BPM from description and definition of key concepts to ultimate goals and objectives. All these definitions assume the process participants have a shared understanding of the organisational goals and they try to realise these goals and objectives by implementing business processes that can be changed and improved [10], [11], [12], [22], [25].

Ould [22] states: "In an organisation, people do things not because they are themselves but because they have a responsibility in the organisation; they are perhaps paid to carry out that responsibility: they have a role in that organisation." This definition can be associated with the functionalist social paradigm. It ignores the individuality of the people in the organisation and tries to unify people based on their roles and responsibilities. This definition also ignores all personal goals and objectives and drives and agendas and reduces them to the money they are being paid to do the job.

Harmon [12] at the beginning of his book "Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals" talks about organisations as systems. He states "In essence, the systems perspective emphasizes that everything is connected to everything else and that it's often worthwhile to model businesses and processes in terms of flows and feedback loops." According to System Dynamics methodology to get an appropriate understanding of a complex system it is essential to form and understanding of the following four phenomena [28]:

- System boundary
- Feedback loop networks
- Level or stock and rate or flow variables
- System leverage/ intervention points

So it is quite clear that in the above statement, Harmon [12] is specifically referring to System Dynamics methodology for analysing organisations as systems. This systems analysis methodology, Systems Dynamics, has been categorised as a methodology only applicable in unitary complex problem contexts by Jackson & Keys [14].

Dumas, Aalst & Hofstede [10], where they talk about Person-to-Person Processes, interestingly, start the discussion in the larger problem context. They mention that people processes are very complex, semi-structured, variable and dynamic and they talk about organisational structure, power distribution, context and cultural settings' impacts on these processes, but as they continue through the chapter they try to reduce the problem under investigation to a problem in the unitary problem context. They start talking about "certain emergent regularities and patterns of group behaviour" and structures that the process participants reproduce repetitively as a result of shared belief and value system. They conclude that seemingly unstructured interactions are to a large extent, dictated by linguistic, cultural and social norms. Eventually they talk about "Formalized interactions" and the fact that as the interactions matures, they develop into routines. It is important to mention that they don't consider this formalisation to mechanise the interactions but to define the "structural backdrop, against which complex and diverse interactions unravel".

It is clear that in reality not all relationships in organisations are bound to the unitary problem context [14], [19], [20], [13] so limiting our definitions and techniques to this problem context causes the outcome of our analysis efforts to be far from the reality of the organisation, incomplete and even misleading.

4 Redefining Key BPM Concepts for Pluralist and Coercive Problem Contexts

In this section new definitions are provided for key BPM concepts that are intended to be relevant to pluralist and coercive problem contexts in addition to the unitary problem context.

It can be concluded from the literature [14], [19], [20], [13] related to organisations as social systems that a large percentage of the current organisational problems can be classified to be in pluralist and coercive problem contexts.

As was mentioned before, problems in the pluralist problem context have the following characteristics [13, 14]: the participants' basic interests are compatible but they don't share the same values and beliefs, therefore it is important that a space be made available for them to have debates, arguments, conflict and disagreement so that they feel that they have been involved in the decision-making and to find accommodation and compromises and agree on productive ways forward towards the temporarily agreed goals and objectives.

Problems in coercive problem context have these characteristics [13, 14]: few common interests exist between the participants, compromise does not seem possible, they have no agreed common goals and decisions are made based on the power structure and distribution.

So as it appears for BPM to be applicable in pluralist and coercive problem contexts, one of its important outcomes should be suggesting and providing the means to facilitate and guide these *conversations, debates, arguments* towards a productive way to achieve temporarily agreed goals and objectives. It seems that the contemporary BPM can be successful if it takes power structure and distribution into consideration and also facilitates its discovery.

Taking into account the above considerations, the following definitions are suggested for the following fundamental BPM concepts [18]:

Activity (AKA business task): Using the following ideas:

1. Briol's [6] definition of activity for choreography diagrams [21]: "An activity is an interaction representing a set of one or more message exchanges between two or more participants";

2. Conversation for Action diagram [24], [26], [27] and
3. The i* framework's strategic dependency concept [29, 30],

a new concept called a dependency cycle [16, 17, 18] can be used for defining a business task or activity. The dependency cycles show how different organisational role instances (people working in the organisation) depend on each other to fulfil an objective. The dependency cycles represent collaboration pieces as they show how people converse, and with conversing, how they depend on each other to fulfil their goals and objectives. A dependency cycle starts with a request and ends with either a withdrawal, i.e. a rejection of the result, or a declaring of fulfilment of the objective of the dependency cycle, or in other words the dependum of the dependency cycle. These dependency cycles and their chronological orders are the main building blocks of business processes.

Business Process: "Business processes are sequences of collaborations – the new activity or business task definition (dependency cycles) – for achieving personal and business goals relevant to the subject matter based on personal intentions and motivations and distribution of power. The ultimate business process forms when all instances of collaboration around the subject matter have materialised and at each moment of time before then, only a business process fragment [23] is at hand."

Business Process Model: Based on the new definition of a business process, business process models are defined as: "Compilations of instances of sequences of collaborations and interaction (activities) for achieving personal and business goals relevant to the subject matter under investigation based on personal intentions and motivations and distribution of power." [18] One important point about this definition is that the instances cannot be reduced to a unified model as, due to the dynamic nature of the process definition, each instance will be different and they don't follow a pre-defined pattern. So the model is a set of distinct instances. Another important point about this definition is that at each point of time we will only have an incomplete model based on the extracted instances and the complete model only forms when all instances have been extracted. This incomplete model is a fragment [23] of the ultimate business process. Therefore at each moment of time, the created business process is a business process fragment. Although these fragments are not complete and consistent enough to be executed, they are of great value for understanding the current situation of the organisation.

There are two interesting points that are worth mentioning here about these new definitions:

1. These definitions are applicable in unitary problem contexts: If we ignore the power distribution and assume that a unified vision with common goals and objectives exists, then the above definitions turn into conventional BPM concepts and definitions.
2. An interesting self-similarity [8] gets created in different parts of the business process model. In other words, in each level of abstraction we have different dependency cycles that in turn create more dependency cycles to fulfil their

objectives. This means that dependency can be both the most granular part and the coarsest grained part of the business process. This fractal like [8] characteristic makes the model and its complexity closer to the reality of interactions within an organisation with a fractal dimensionality than the n-dimension models in conventional BPM. This can more easily be seen in the next section where we will illustrate the use of these new concepts in an example.

Mavaddat et al. [18] have validated the suggested definitions in a proposed framework using two case studies. Using these two case studies it was demonstrated that the extracted business processes provide a good representation of the reality and suffer to a lesser extent from conventional business process management definitions' and techniques' paradigmatic limitation.

5 A Synthesised Example

This example illustrates the application of the proposed new definitions for BPM key concepts.

Consider the following simple scenario taken from [18] – an instance of a business process fragment [23]:

Mr. X, who is the manager of Mr. Y, has asked Mr Y. to write a report on subject S. Mr. Y sets up a meeting with Mrs. P and Mr. Q, who have a better understanding of the subject S and consults with them to learn more about the subject. Mr. Y then writes up the report and asks his assistant Mr. T to type it and then Mr Y. ultimately sends it to his manager Mr. X. There are some mathematical calculations in the report, therefore Mr. T, in order to be able to type the report, needs a special word-processing module that he buys from the Supplier1.

The first dependency cycle starts when Mr. X asks Mr. Y for a report on subject S and it finishes when Mr. Y provides Mr. X with the report. It can be tagged a resource dependency cycle, and, using the choreography notation [18], the model in Figure 2 can be created.

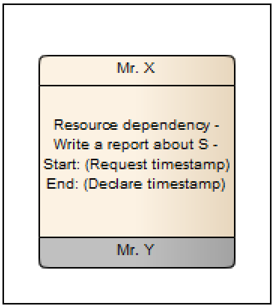


Fig. 2. Process instance from Mr. X's point of view

This model is complete by itself from Mr. X's point of view. He has asked for something and he has received it. The model can delve deeper though, it can now show the process from Mr. Y's point of view, that is:

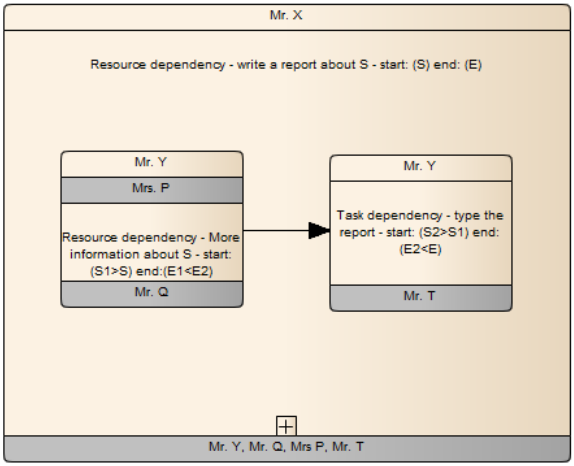


Fig. 3. Process instance from Mr. Y's point of view

Figure 3 illustrates that from Mr. Y's point of view, because he has broader information about the process. Mr. X has a dependency on Mr. Y, Mr. Q, Mrs. P and Mr. T.

Mr. X's dependency on Mr. Y is a direct dependency as he has asked Mr. Y directly to do something but he has an indirect dependency on others through Mr. Y. Mr. X was not probably even aware of these indirect dependencies.

Figure 3 also shows that Mr. Y, to fulfil the task requested from him by Mr. X, has created three other dependencies: two dependencies on Mrs. P and Mr. Q for consultation and one dependency on Mr. T for typing the final report. It can be seen in Figure 3 that Mr. Y is not aware of Mr. T's dependency on Supplier1.

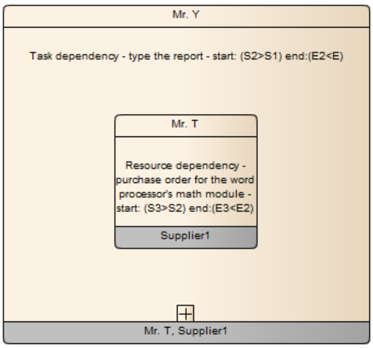


Fig. 4. Process instance from Mr. T's point of view

We can now create the model from Mr. T's point of view. As there are some mathematical calculations in the draft of the report, Mr. T needs to buy a module for his word processor in order to type it, he sees the process as shown in figure 4.

Figure 4 shows that the process instance starts for Mr. T when he is asked to type the report and finishes when he delivers the typed report to Mr. Y. It also shows that in order to fulfil the dependency, Mr. T has to create a new dependency on Supplier1.

Comparing the process from Mr. Y's perspective and Mr. T's perspective, it can be seen that Mr. Y might not even be aware of the dependency he has on Supplier 1 through the dependency that Mr. T has created with them.

So by combining different viewpoints the model in Figure 5 can be created:

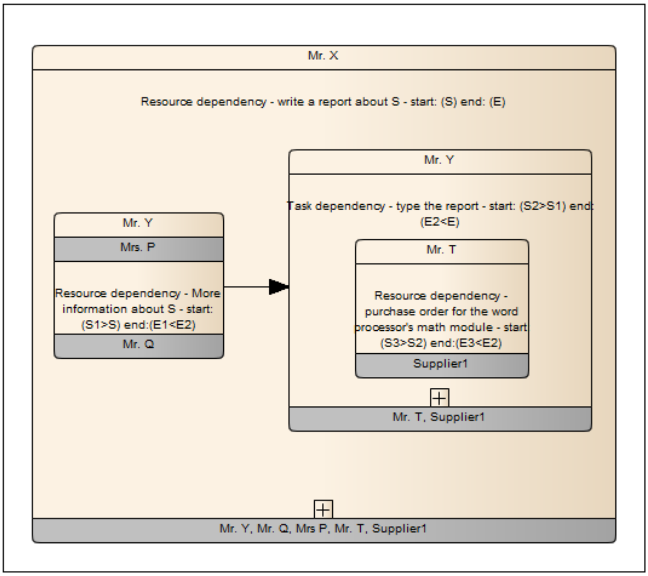


Fig. 5. Combined viewpoints – business process instance

Figure 5 shows the complete process instance. It shows that a dependency cycle has been created by Mr. X with Mr. Y directly and with Mr. Q, Mrs. P, Mr. T (indirectly through Mr. Y) and Supplier 1(indirectly through Mr. Y and Mr. T). The dependum of the dependency cycle is the report on subject S and the type of the dependency cycle is a resource dependency. In order to deliver the dependum of the main dependency cycle, the report on subject S, Mr. Y has created a dependency cycle with Mrs. P and Mr. Q to consult about the report's subject S. The type of the dependency cycle is a resource dependency cycle and the dependum of it is "information" about the subject of the report. Mr. Y creates another dependency cycle with Mr. T (directly) and with supplier1 (indirectly through Mr. T) to type the report. The dependency cycle type is task dependency and the dependum of this dependency cycle is "typed report". Finally Mr. T, to deliver the dependum of the dependency cycle that is "typed report" creates a dependency with Supplier 1. The type of the dependency cycle is resource dependency and its dependum is a word-processor module.

The characteristics that were discussed in the previous section can be seen from the above models:

1. The main focus is people's interactions and collaborations for achieving the goals and objectives
2. It looks at the business process from different participants' viewpoints so it is easier to model individual objectives as well as the ultimate business process goals
3. It manifests self-similarity in every level. The final business process fragment is structurally similar to the constituent business tasks.

Based on the new BPM concept definitions a framework has been developed for extracting business processes from conversation logs and it has successfully been applied to two email corpora as the source of data in a case study. The new concept definitions, the framework and the case studies are presented in detail in the author's PhD thesis [18].

6 Conclusion

In this paper, using the history of the formation of BPM concepts and by referring to the definitions of concepts provided by the main scholars of the field, it was demonstrated that current BPM practices look at an organisation as a social system from the functionalist social paradigm and as a result make them applicable only in unitary problem contexts. This means conventional BPM practices are only applicable in situations where the participants

- Have similar values, beliefs and interests
- Share common goals and objectives
- Are all involved in decision-making about how to achieve the common goals and objectives.

These practices are not applicable in pluralist and coercive problem contexts. But, only a very small percentage of current organisational problems are grounded in unitary problem contexts and this paradigmatic limitation of BPM causes its results to be unrealistic and at points misleading. Therefore, contemporary BPM practices should try not to be bound just to the unitary problem context so it should provide a means to analyse organisations from different perspectives and using different social paradigms.

In order for BPM to be able to be applicable in different problem contexts, its concepts and definitions should be restated and revised. The world that the conventional BPM practices represent is the dry world of tasks, task sequences, roles, predefined decisions and predefined goals and objectives, or, in a nutshell, a rigid, deterministic world whose main purpose is efficiency and effectiveness. On the other hand the real world that contemporary BPM practices should try to represent seems to be the world of people and their collaboration and conversation, the world of debate, argument and conflict, the world of compromises and accommodations and compensation, the world of power struggle and temporary goals and objectives, or, in a nutshell, an uncertain, stochastic and at points chaotic world.

To lessen the coupling between BPM and the unitary problem context, new definitions for fundamental BPM concepts were proposed based on pluralist and coercive problem context characteristics and the conversation for action diagram and i* strategic dependency concepts. Their application was illustrated in a synthesised example and it was argued why these new concepts and definitions are better representations of the reality of business processes in an organisation [16, 17, 18], [24, 26, 27], [29, 30]. The new definitions are not intended to contradict or eliminate the previous business process definitions; in fact they complement the previous work, as has been shown. However, they should enable the weak points in an existing process, or system that support the process to be identified [17] and also enable the provision of improved business process support systems.

Future work will include field studies both to confirm the underlying assumptions about the nature of organisations and to validate the efficacy of the approach taken to address these assumptions.

References

1. van der Aalst, W.M.P.: Challenges in business process management: Verification of business processes using Petri nets. Bull. EATCS 80, 174–198 (2003)
2. van der Aalst, W.M.P.: Challenges in business process analysis. Enterp. Inf. Syst. (2009)
3. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: A research agenda. Comput. Ind. 53, 231–244 (2004)
4. Aalst, W.M.P., Nikolov, A.: Mining e-mail messages: Uncovering interaction patterns and processes using e-mail logs. IJIIT 4(3), 27–45 (2008)
5. Alvesson, M., Deetz, S.: Critical Theory and Postmodernism Approaches to Organizational Studies. In: Clegg, S.R., Hardy, C., Nord, W.R. (eds.) Handbook of Organization Studies, pp. 191–217. Sage Publications (1996)
6. Briol, P.: BPMN 2.0 Distilled. lulu.com (2010)
7. Burrell, G., Morgan, G.: Sociological Paradigms and Organizational Analysis. Heinemann (1979)
8. Brown, C., Liebovitch, L.: Fractal Analysis, Series: Quantitative Applications in the Social Sciences. Sage Publications Inc. (2010)
9. Dietz, J.L.G., Habing, N.: The notion of business process revisited. In: Meersman, R. (ed.) OTM 2004. LNCS, vol. 3290, pp. 85–100. Springer, Heidelberg (2004)
10. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software Through Process Technology. Wiley-Blackwell (2005)
11. Dumas, M., La Rosa, M., Mendling, J., Reijers, A.: H.: Fundamentals of Business Process Management. Springer (2013)
12. Harmon, P.: Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals, 2nd edn. Morgan Kaufmann (2007)
13. Jackson, M.C., Keys, P.: Towards a System of Systems Methodologies. J. Oper. Res. Soc. 35, 473–486 (1984)

14. Jackson, M.C.: Systems Thinking: Creative Holism for Managers. John Wiley & Sons (2003)
15. Jeston, J., Nelis, J.: Business Process Management: Practical Guidelines to Successful Implementations, 2nd edn. Routledge (2008)
16. Mavaddat, M., Beeson, I., Green, S., Sa, J.: Facilitating Business Process Discovery using Email Analysis. In: The First International Conference on Business Intelligence and Technology (2011)
17. Mavaddat, M., Green, S., Sa, J.: Improving Organisational Structure, Culture, Communication and Business Processes through Dependency Cycle Extraction and Analysis. In: IADIS International Conference Information Systems 2013 (2013)
18. Mavaddat, M.: Business Process Discovery through Conversation Log Analysis in Pluralist and Coercive Problem Contexts. Ph.D thesis, University of the West of England (2013)
19. Morgan, G.: Images of Organization. Sage, London (1997)
20. Morgan, G.: Images of Organisation. Sage, London (1986)
21. Omg.org: Business Process Model and Notation (2011), `http://www.omg.org/spec/BPMN/2.0/`
22. Ould, M.A.: Business Process Management - A rigorous approach. BCS. The Chartered Institute for IT (2005)
23. Schumm, D., Karastoyanova, D., Kopp, O., Leymann, F., Sonntag, M., Strauch, S.: Process Fragment Libraries for Easier and Faster Development of Process-based Applications. J. Syst. Integr., 39–55 (2011)
24. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press (1969)
25. Weske, M.: Business Process Management Concepts, Languages, Architectures. Springer (2007)
26. Winograd, T.: A Language/Action Perspective on the Design of Cooperative Work. ACM SIGCHI Bull. 20, 79 (1988)
27. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. Addison Wesley (1987)
28. Wright, D., Meadows, D.H.: Thinking in Systems: A Primer. Routledge (2009)
29. Yu, E.: Modelling strategic relationships for Process Reengineering. In: Social Modeling for Requirements Engineering, pp. 11–152. MIT Press (2011)
30. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering (Cooperative Information Systems Series). MIT Press (2011)

Towards Process-Aware Cross-Organizational Human Resource Management*

Cristina Cabanillas[1], Alex Norta[2], Manuel Resinas[3], Jan Mendling[1], and Antonio Ruiz-Cortés[3]

[1] Vienna University of Economics and Business, Austria
{cristina.cabanillas,jan.mendling}@wu.ac.at
[2] Tallinn University of Technology, Estonia
alex.norta@gmail.com
[3] University of Seville, Spain
{resinas,aruiz}@us.es

Abstract. Finding human resources with the required set of skills, experience, and availability to execute an activity at a specific moment, is a socio-technical challenge for enterprises that use business-process aware systems. On an intra-organizational level, there exists an increasing body of knowledge for automated human-resource management. However, the recent pervasiveness of service-oriented cloud computing combined with mobile devices and big data, has resulted in the emergence of cross-organizational ecosystems in which workforce is distributed. Consequently, human-resource management has to consider more requirements compared to a purely intra-organizational setting. This position paper addresses the gap and describes a set of challenges in the management of human resources in service outsourcing scenarios based on process views and automatic process-view matching. The contribution is a specification of research directions that must be pursued so that resource management successfully adopts the special requirements for scaling to a cross-organizational level.

Keywords: human resource management, process matching, process view, resource allocation, resource assignment, service outsourcing.

1 Introduction

For running human resource management in Virtual Enterprises (VEs) [1], new enabling concepts and technologies are available in the form of Service-Oriented Cloud Computing (SOCC) [2]. This is facilitated via the use of Business Process as a Service (BPaaS) [3] that helps to choreograph service interactions in the VEs. Organizations that engage in a VE configuration for the

* This work was partially supported by the European Union's Seventh Framework Programme (FP7/2007-2013), the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants 318275 (GET Service), P12-TIC-1867 (COPAS), TIN2012-32273 (TAPAS), TIC-5906 (THEOS)).

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 79–93, 2014.

purpose of carrying out business transactions have an interest in disclosing only what is necessary. Otherwise, details of Business Processes (BPs) on an in-house level must remain opaque for protecting business secrets that represent competitive advantages. A disclosure of the latter usually results in an organization loosing its base for revenues and financial solvency. The means for hiding crucial business internals is to employ *process views* that are subsets of the full in-house conceptual-level processes in the domain of a collaborating party. A process view shields business secrets or irrelevant details and allows an organization to reveal only publicly relevant parts of its private BPs to partner organizations.

Service outsourcing is a business paradigm in which a service-providing organization performs or coordinates parts of a Business Process (BP) of a service-consuming organization. These parts were typically performed or coordinated by the service-consuming organization itself. As depicted in Figure 1, the so-called eSourcing-framework [4,5] supports service outsourcing by enabling the flexible construction and structural *matching* of public, external process views that are extractions of private, internal BPs so that the same activity can occur in different models. A so-called BPaaS-HUB [6] facilitates this matching process during setup time by serving as a repository for the process views and offering automation support during matching, e.g., for background checking potential collaborating counterparties on the fly through mashups.

However, process views and matching relations have been defined considering only the behavior of the process, i.e., the control flow. The human resource perspective involved in the collaboration has hardly been considered in this context, despite its importance for the actual execution of the outsourced activities. Indeed, one of the main reasons of activity outsourcing is the lack of resources with the required skills or the required software within the organization, and the impossibility to acquire them [7].

This position paper works towards that gap by identifying six challenges concerning the research question "how can human resources be managed cross-organizationally assuming a business-process aware collaboration paradigm in VEs?", and two sub-questions deduced from it: (RQ1) How can process views be extended with human resources information? (RQ2) How can the matching of process views with human resources information be automated?

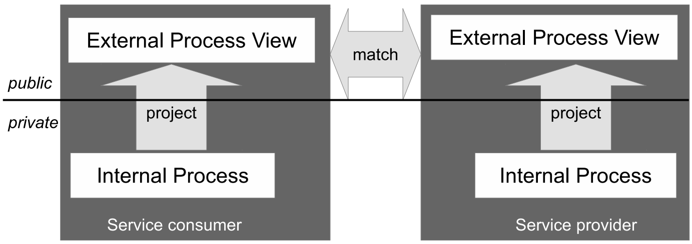


Fig. 1. A specification framework for service outsourcing [4]

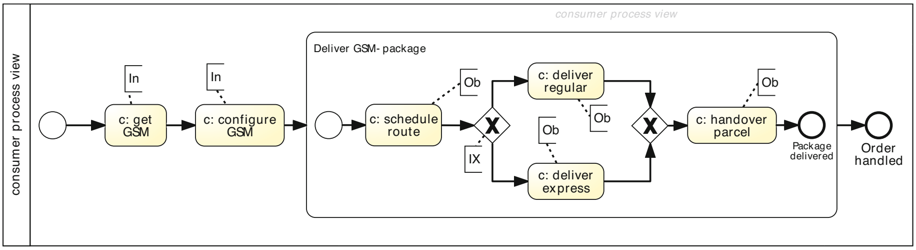


Fig. 2. BPMN process view of consumer. Used annotations: In=invocable, Ob=observable, EX=EXOR, IX=IXOR. [4].

The structure of this paper is as follows. Section 2 describes a scenario that reflects real needs for resource management in eSourcing. Section 3 defines the concepts related to process views and describes the challenges that give answer to RQ1. Section 4 follows the same procedure for resource-aware process matching (i.e., RQ2). Finally, Section 5 outlines conclusions and future work.

2 Motivating Scenario

In this section, we use concepts related to service outsourcing with process views to describe a motivating scenario that has been adapted from [4]. We also explain the involvement of resources in the process activities and how they are managed. Those details on process views and process-view-based matching that are not relevant for the understanding of the scenario are provided in later sections of the paper.

2.1 Service Consumer Side

As part of a larger in-house process located on the internal process layer of a service consumer, Figure 2 shows an example of a telecom sales process, which is a view projection as a service request to the external (public) layer (cf. Figure 1). This projected process view starts with picking a Global System for Mobile (GSM) and configuring it according to customer[1] demand. In order to speed up the process, the consumer wants that this configuration (activity *c: configure GSM*) be performed by a person with at least one year of experience configuring GSM devices, and if possible, the same person must be in charge of this activity for all the process executions, i.e., a Binding of Duties (BoD) at process level [8]. Thus, the *condition* related to experience becomes a necessary (maybe not sufficient) requirement to *assign* the activity to people at the provider side.

Thereafter, a subprocess is executed to deliver the GSM package, containing the following activities. First, a route is scheduled, and then, either an

[1] The customer is the final beneficiary of the process, hence, in this case the purchaser of the GSM device. The consumer gives service to the customer.

express or a regular delivery is performed. In order to avoid choosing transportation routes in own favor for reasons external to the enterprise, the person in charge of scheduling the route cannot be same person delivering the package in the same process execution, i.e., the consumer decides to implement a Segregation of Duties (SoD) policy [8]. The final activity is to hand the GSM package over to the customer, who signs the receipt.

The model representing this process view in Figure 2 uses a specific notation for outsourcing based on the de-facto standard Business Process Model and Notation (BPMN) [9]. We distinguish equally labelled activities that are defined in different domains by prefixing activity labels with a namespace indicator, which is either *c:* for a consumer, or *px:* for a provider. Note that labels may be different but semantically judged as equivalent. If the service provider initiates an activity, it is called *observable* [10,11] (labeled *Ob* in Figure 2) and if the service consumer initiates an activity, it is called *invocable* [10,11] (labeled *In*). We annotate a choice construct to specify whether it is external or internal to a service provider. The annotation IX in Figure 2 denotes that the provider decides on the branch of an exclusive-choice split during enactment, and an EX-annotation means the service consumer decides. Notice that information on resources (e.g., assignment conditions, preferences or allocation mechanism) is missing in the model, as resource management has not been considered in process views yet.

All the activities in the process view must be executed by a service provider, from which the set of resources meeting the conditions established are assigned to the activities. From the *potential performers* assigned to the activities, only some will be selected as *actual performers* when the process is under execution, considering preferences or further allocation criteria.

However, before starting the collaboration, decisions on who chooses the actual performer for the activities, whether the consumer is notified about it, and who must deal with allocation exceptions, must be negotiated between consumer and provider. In the case at hand, the provider can decide on the final person to be allocated to outsourced activities, as long as she guarantees that all the assignment conditions and the preferences established by the consumer are considered. Nonetheless, the consumer wants to be able to assess performance in the outsourced process considering the people involved so that she can make future decisions. Therefore, the provider is obliged to inform her about the actual performers of the tasks. In case of allocation problems (e.g., illness or holiday season causing a lack of potential performers), the provider is responsible for solving the problem.

2.2 Service Provider Side

The model in Figure 3 shows a private process owned by a service provider for matching to the consumer process in Figure 2. The two subprocesses in the BP share some activities and ordering constraints with the consumer process, for example *px: get GSM* and *px: configure GSM*. Furthermore, activity labels *c: schedule route* and *px: determine route* can be considered synonyms and, thus,

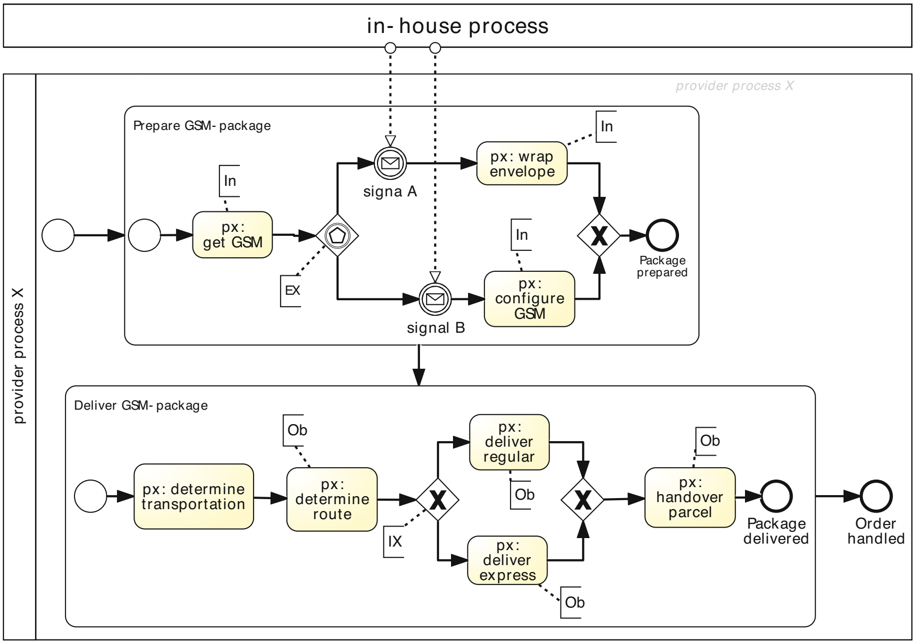


Fig. 3. BPMN process of service provider [4]

they match. However, the provider may hide internal resource-related information if it is sensitive and should not be seen by the consumer, e.g., specific information related to the organizational structure, which in this case is stored in a database and implements a Role-Based Access Control (RBAC) model [12], i.e., a hierarchy of roles. Nevertheless, that information can be internally used to extend the assignment conditions established by the consumer, so that it is opaque to the consumer. For instance, for activity *c: configure GSM*, the provider may decide that the performer must be someone with role Technician. Therefore, taking into account both consumer's and provider's assignment conditions, all the technicians of the provider with at least one year of experience in GSM configuration are *potential performers* to execute the activity. Then, the *preference* specified by the consumer in terms of a BoD (see above) serves for *ranking* them, so that the person with highest priority will be *allocated* to the activity and become its *actual performer.*

Still, the provider BP contains unobservable and uninvocable activities that are not part of the consumer's process view. In addition, the private BP of the provider in Figure 3 differs from the consumer view in Figure 2 by an event-based gateway. The two message flows in Figure 3 from the service consumer's in-house domain depict that the service consumer decides the gateway path. The provider process contains additional activities *px: wrap envelope* and *px: determine transportation* that are opaque for the service consumer during enactment.

Table 1. Projection relations between internal processes and process views [4]

Projection Relation	Omitting	Hiding	Aggregation	Relation between process view and internal process	Monitoring
Black box		x	x	process view is single observable activity, internal process has no invocable activities	very limited
Glass box		x	x	process view has only multiple observable activities, internal process has no invocable activities	limited
Gray box	x	x		process view has multiple invocable and observable activities, internal process has invocable and observable activities	limited, partial, full
Open box	x	x	x	process view has multiple invocable and observable activities, internal process has invocable and observable activities	limited, partial, full
White box				process view and internal process are identical	limited, partial, full

3 Service Outsourcing with Process Views

As depicted in Figure 1, the so-called eSourcing framework [5,4] supports service outsourcing by enabling the flexible construction and structural matching of public, external process views that are extractions of private, internal processes so that the same activity can occur in different models. Concerning the control-flow perspective, these views can be defined in terms of three *projection relations.*

– Hiding: a set of nodes executed at the internal level are not shown in the process view at the external level.
– Omitting: a set of nodes that do not need to be executed at the internal level are not shown in the process view at the external level, e.g., activities of an $EXOR$-branch.
– Aggregation: a set of nodes executed at the internal level is shown as a single node in the process view at the external level.

While matching of service offers with service requests requires the process views to be isomorphic, there are projection options termed black-, glass-, gray-, open- and white box [4] between process views and in-house BPs. In Table 1 the extreme projection relations are listed. Black box, glass box, and open box stem from a Web-service outsourcing example by [13] while [14] identifies gray-box and white-box projection in a Petri-net study. All other possible projection relations are hybrid forms of these extreme relations.

Black-box projection occurs if the process view contains only a single observable activity that aggregates or hides a set of nodes from the internal process. The internal process does not contain invocable activities as it is not possible to hide or aggregate them. Also $EXOR$-nodes are no abstraction option as the service consumer cannot monitor or control the internal provider process.

Glass-box projection occurs if the process view only contains observable activities that hide or aggregate activities from a internal-level process while the process view has no invocable activities. The service consumer merely monitors the internal-process progress.

Gray-box projection occurs if the process view comes into existence through hiding and omitting from the internal proccess. The process view optionally contains both observable and invocable activities while there is no aggregation. The service consumer can monitor the internal process using observable activities in the process view. The service consumer can control the progress of the internal process at th provider side through invocable activities in the process view.

Open-box projection occurs by establishing the process view through hiding, omitting, and aggregation from the internal proccess. The process view contains both observable and invocable activities, allowing a service consumer to monitor and control the progress of the internal process.

Finally, with a *white-box projection* the process view is identical to the internal process. There is no application of abstraction rules and the service consumer has a direct view on the internal process of the provider including full monitoring and control of the provider-process.

Including human resources in process views such as those mentioned above raises the following challenges:

Challenge 1: Projection Relations for Human Resource Assignments

Description: There might be sensitive information that an enterprise has to hide or omit. For instance, as described in the motivating scenario, the consumer may not be able to show all the organizational structure in order to not violate the Data Protection Act or similar, or she may not want to reveal all the skills of the employees to not take certain risks, e.g., get employees recruited by the provider, or get threatened to post private information. The situation aggravates when a second outsourcing to another supplier is initiated by the provider. Indeed, several situations have occurred in the past related to the publication of sensitive information in outsourcing scenarios. For instance, in 2003 a medical transcriber in Pakistan threatened to post patients' private records online of the California San Francisco Medical Center (UCSM) if certain wages were not paid by an intermediate company that had outsourced to them activities which had been in turn outsourced by the UCSM [15].

Therefore, ensuring privacy control in all the parties involved in an eSourcing collaboration, at the same time as the sharing of information required from each other, is fundamental. Projection rules and relations are defined to guarantee the required degree of privacy for every party involved.

State of the Art: In the context at hand, three projection rules and five projection relations are defined to guarantee the required degree of privacy for every party involved [4]. However, only control-flow aspects are considered so far. Studying how to manage resources in the projection rules (paying special attention to aggregation) and in the projection relations, is necessary. New rules and relations may also be identified. As a result, all the parties involved in an eSource scenario must be able to hide or omit resource-related information in the projections.

Challenge 2: Expressiveness of the Resource Assignment Language

Description: A language to define the conditions that the members of an organization must meet to be allowed to participate in an activity, is required in order to specify resource assignments in BPs. Different languages offer different expressiveness regarding the types of conditions that can be defined. The most basic ones are related to organizational entities such as roles, positions, or organizational units; and capabilities associated to people such as their skills or their education. However, advanced features are desirable when dealing with security issues and people experience. In particular, in the domain at hand, the most interesting conditions that should be covered by the assignment language can be seen as five different sub-challenges related to expressiveness, namely:

Challenge 2.1: Capability-Based Resource Assignment. The ability to define assignment conditions based on the capabilities associated to resources, e.g., their skills or their education.

Challenge 2.2: Security-Aware Resource Assignment. The ability to define access-control constraints to configure security in the execution of outsourced process activities, e.g., SoD and BoD [8].

Challenge 2.3: History-Aware Resource Assignment. The ability to define the assignment of resources based on historical information on past process executions. This is especially relevant in long-term outsourcing scenarios where the consumer may want to assign resources depending on their past performance in similar activities.

Challenge 2.4: Preferences in Resource Assignment. The ability to define preferences on the people that can execute an outsourced activity to give priority to the most suitable person, so that the provider must try to allocate the work to the first person in the priority ranking generated according to the organizational model of the company and the characteristics of its members. Preferences have implications in order challenges (see Challenge 3).

Challenge 2.5: Task Duties in Resource Assignment. The ability to define different degrees of responsibility associated to an activity, e.g., a person responsible for its execution, a person accountable for it, or a person providing external information required for its completion.

Note that the assignment specified by the consumer defines the minimum set of conditions that must be fulfilled. Further conditions can be added by the provided internally, of which the consumer might not be aware (cf. Section 2).

State of the Art: The languages for the definition of resource assignments typically support only a subset of the features mentioned above. For instance, BPMN [9] and WS-Human Task [16]/BPEL4People [17] do not allow defining access-control constraints natively. In the case of BPMN, this could be done by using another language with the notation, as proposed in [18]. WS-Human Task and BPEL4People allow defining different task duties for the BP activities, which constitute a subset of the task duties defined in so-called RASCI matrices [19]. Yet Another Workflow Language (YAWL) [20] and Architecture of Integrated Information Systems (ARIS) [21] do not take into account information about past executions or the existence of different task duties associated to an activity. Finally, some approaches [8,22] deal with security and history-based aspects but they do not support the definition of conditions about resource capabilities because they are based on the RBAC model [12], which relies only on organizational roles. Resource Assignment Language (RAL) [18,23] could be a candidate to achieve this challenge, as it provides support for features 2.1, 2.2, 2.3 and 2.5, as well as conditions based on organizational entities.

Regarding preferences, different types of preferences and different formalisms to define them have been used in different domains [24]. For example, quantitative preferences have been discussed in economics, operations and web systems [25,26,27,28], while qualitative preferences have been the focus of artificial intelligence and database research [29,30,31]. Some approaches for the discovery and ranking of semantic web services cover both types of preferences. Specifically, Semantic Ontology of User Preferences (SOUP) [32] has been used to define preferences in Business Process Management (BPM) together with RAL [33], thus bridging and existing gap in intra-organizational process-aware resource management.

Challenge 3: Control over the Allocation of Outsourced Activities

Description: The consumer is typically in charge of defining the minimum set of resource assignment conditions for the outsourced activities, but an agreement on who decides on the allocation and the information publicly available for the consumer is also necessary. There are at least three options: (i) the consumer does not know who in the provider performs the work; (ii) the consumer cannot make a decision on the worker for the outsourced activities, but he gets informed about who performed the task after completion; (iii) the consumer decides who must execute the activity from the set of potential performers of the provider, thus being totally aware of the allocation procedure.

This is specially relevant when security policies must be implemented in the process (e.g., the Dynamic SoD (DSoD) and Static BoD (SBoD) constraints applied in the motivating scenario), as it is necessary to take into account who has participated in previous executions of the process activities. The degree of transparency

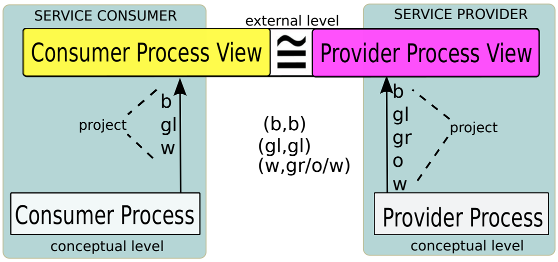


Fig. 4. Valid collaboration configurations [4]

in resource allocation must, thus, be agreed upon between consumer and provider, as privacy concerns also play a role in this decision (see Challenge 1).

Also related to this challenge is the question of who must handle exceptions related to resource allocation in case of a lack of potential performers available. Under those circumstances, the consumer might want to modify the resource assignments of the outsourced activities that are affected, or the resolution of the conflict might fall on the provider, which can, e.g., implement some substitution policy always guaranteeing that the assignment conditions remain fulfilled. In the worst case, a second outsourcing might be required.

State of the Art: In [4], the concepts of invocable and observable activities in eSourcing scenarios were introduced related to the party involved in the collaboration that must initiate the work in the outsourced activities. However, there was no information about who selects the specific person that works in the activity and the related information that is made visible to the other party. Similarly, exception handling is disregarded and becomes especially relevant when resources are taken into account. Therefore, the extension of these concepts to consider resource allocation and reaction to potential exceptions is an open issue.

4 Matching between Process Views

The service consumer on the left and service provider on the right of Figure 4 associate their process views with one another using the matching relations of Section 3 relating their process views to respective underlying internal processes. The figure shows only certain combinations of projection (cf. Section 3) and matching relations that are possible for process views at the external level. The internal levels show the useful projection relations near the projection arrow for the service consumer and provider. In the centre, the tuples with projection combinations are permitted combinations for matching process views on the external level. If the service consumer performs a black-box projection, the provider must also use black-box projection. All other projections do not yield a single aggregation externally.

If the service consumer uses a glass-box projection, it imposes a limitation on the service provider that he can only use observable activities. Thus, the latter must also respond with a glass-box projection. However, when a service consumer uses open-, or white-box projection, the service provider has the options of a gray-box, open-box, or white-box projection. The contained invocable activities eliminate black-box and glass-box projections for the service provider.

A service consumer cannot use a gray-box or open-box projection as the process view may omit an invocable node from the internal process. However, that creates a problem during enactment time when the consumer invokes a node from her internal process level without the event being able to bridge into the domain of the service provider because the required externally projected equivalent invoke node is missing in the process view.

The previous matching relations have been defined disregarding the existence of resource assignments and, thus, the need of managing human resources. With resource-aware process views, the matching mechanisms between the process views of consumer and provider should be extended in several ways. In particular, the following challenges have been identified:

Challenge 4: Consideration of the Resource Perspective in the Matching Relations of the Process Views

Description: As consumer and provider develop process views independently, these are not necessarily in line. However, they must be compatible and must meet the degree of matching needed for the communication, which varies in different scenarios, e.g., if an enterprise outsources part of its BP to a supplier, synchronization might not be required, but the monitoring of the progress at the provider side might be desired [4]. Therefore, the resource perspective must be taken into account also when defining matching relations, which may change with respect to those defined considering only the control flow of the process. For instance, the different degrees of matching that can be found between the resource assignments defined by consumer and provider has to be studied.
State of the Art: As described above, Eshuis et al. identified three matching relations related to the BP control flow [4], but resource assignments were disregarded. These matching relations should be re-studied and adapted considering resource-related information. New matching relations stemming from the consideration of resource assignments in the process activities might be identified.

Challenge 5: Different Vocabulary in the Parties Involved

Description: Natural language is very rich and the terminology used to refer to concepts vary depending on the country, the region and even, the person. Searching for term matching in different languages may also be challenging. These issues affect organizational metamodels too, in which the same concept is sometimes referred to in two different ways (synonyms) and two terms are usually used to mean the same concept (homonyms). Capabilities and/or qualifications or professional regulations tend to receive different names in different contexts or domains as well. All this might incur in a lack of understanding between the parties involved in a collaboration. Therefore, ensuring that consumer

and provider use the same vocabulary (or at least a vocabulary that both can interpret in the same way) is crucial for eSourcing success.
State of the Art: A domain-specific dictionary or thesaurus upon which all the parties must agree, can be used to provide the terms with a specific meaning. Several approaches can be used to build it, e.g. Levenshtein distance [34] or Lin metric [35], which provide meaningful semantics of the concepts and their equivalence. There are dictionaries such as WordNet [36] and BabelNet [37], on which we can rely. Indeed, these tools have been used in previous approaches to automatically generate BP models from textual descriptions [38], and vice versa [39], producing good results. For the purpose at hand, Natural Language Processing (NLP) techniques must be put in place.

Challenge 6: Automation in the Matching of Process Views

Description: Algorithms to automate the matching check between the process views of the parties involved in a collaboration are required. This would allow ensuring not only that process views are compatible with each other before initiating a collaboration, but also that the checking can be repeated during process execution if something changes in one of the parties, at a lower time than if having to perform it manually. Therefore, this challenge contributes to save time and the subsequent human cost, at the same time as it helps to improve the quality of the service and provides flexibility to perform changes whose consequences can be straightforwardly checked. Please, notice that Challenge 5 must also be considered to guarantee that process matching meets the requirements.
State of the Art: General concepts from ontology matching [40] have recently been adopted for the automatic matching of processes [41]. The challenge at hand is in this context that activities can be labeled with verb phrases in heterogeneous ways [42], which goes beyond what ontology matching can readily handle. The control flow perspective provides the opportunity to constrain the search space for matches [43]. Improvement directions have been investigated in [44,45,6]; however, without taking the resource perspective explicitly into account.

5 Conclusions

In this paper, we have explored how human resources fit in a process-aware cross-organizational setting in order to answer the question "how to manage human resources cross-organizationally assuming a business-process aware collaboration paradigm in VEs?". The starting point is a description of service outsourcing based on the definition of process views and different matching relations to check process compatibility of the parties involved, as introduced in the eSourcing framework [5,4].

In particular, we have described a scenario based on a real case in which the resource perspective is reflected. Then, we have identified a set of challenges related to the two main elements involved in eSourcing scenarios, specifically three challenges related to the integration of human resources information in process views, and three challenges associated to the automatic matching between process views with resources information. The state of the art of related work has been summarized for every challenge.

Altogether, the problem is complex because many issues need to be addressed (e.g., a resource assignment language, a dictionary and a method for the disambiguation of natural language), and sensitive aspects such as privacy and security must be dealt with. However, we have found approaches for intra-organizational human resource management that provide (partial) support for (some of) the desired features. For instance, RAL [18] could potentially be used to specify expressive resource assignments, and matching algorithms developed to compare BP models similarity [41] could be adapted to the eSourcing domain.

Nonetheless, in this paper we have assumed that a process activity is always executed by a single person, i.e., individual allocation. In reality, many activities require collaborative work, especially in domains such as software development. Therefore, team work should also be studied in this context.

References

1. Mehandjiev, N., Grefen, P. (eds.): Dynamic Business Process Formation for Instant Virtual Enterprises. Springer (2010)
2. Pimentel, F.D.P.E., Zavattaro, G.: Service-oriented and cloud computing (2012)
3. Chou, S.-W., Chiang, C.-H.: Understanding the formation of software-as-a-service (saas) satisfaction from the perspective of service quality. Decision Support Systems (2013)
4. Eshuis, R., Norta, A., Kopp, O., Pitkänen, E.: Service Outsourcing with Process Views. IEEE Transactions on Services Computing 99 (2013)
5. Norta, A., Eshuis, R.: Specification and verification of harmonized business-process collaborations. Information Systems Frontiers 12, 457–479 (2010)
6. Norta, A., Kutvonen, L.: A Cloud HUB for Brokering Business Processes as a Service: A Rendezvous Platform that Supports Semi-automated Background Checked Partner Discovery for Cross-Enterprise Collaboration. In: 2012 Annual SRII Global Conference (SRII), pp. 293–302 (2012)
7. Straub, D., Weill, P., Schwaig, K.S.: Strategic Dependence on the IT Resource and Outsourcing: A Test of the Strategic Control Model. Information Systems Frontiers 10(2), 195–210 (2008)
8. Strembeck, M., Mendling, J.: Modeling process-related RBAC models with extended UML activity models. Inf. Softw. Technol. 53, 456–483 (2011)
9. OMG, BPMN 2.0, recommendation, OMG (2011)
10. Ebert, J., Engels, G.: Observable or invocable behaviour - you have to choose! Technical report 94-38, Leiden University (1994)
11. Preuner, G., Schrefl, M.: Requester-centered composition of business processes from internal and external services. Data Knowl. Eng. 52(1), 121–155 (2005)
12. American National Standards Institute, Inc., Role-Based Access Control. ANSI INCITS 359-2004 (February 2004), http://csrc.nist.gov/rbac
13. Grefen, P., Ludwig, H., Dan, A., Angelov, S.: An analysis of web services support for dynamic business process outsourcing. Information and Software Technology 48(11), 1115–1134 (2006)

14. Norta, A.: Exploring Dynamic Inter-Organizational Business Process Collaboration. PhD thesis, Technology University Eindhoven, Department of Information Systems (2007)
15. Brown, M.: Privacy and Outsourcing: Evolving Concerns. Newsletter (May 2007)
16. Web Services-Human Task (WS-HumanTask) v1.1, tech. rep., OASIS (2010)
17. WS-BPEL Extension for People (BPEL4People), tech. rep., OASIS (2009)
18. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 50–61. Springer, Heidelberg (2012)
19. Website, The RASCI matrix (last accessed in January 2014), `http://www.ha-ring.nl/en/doc_en/rasci-matrix`
20. van der Aalst, W.M.P., ter Hofsted, A.H.M.: YAWL: Yet Another Workflow Language. Inf. Syst. 30(4), 245–275 (2005)
21. ARIS, RACI, ARIS Community Website (2012), `http://www.ariscommunity.com/raci`
22. Casati, F., Castano, S., Fugini, M.: Managing Workflow Authorization Constraints through Active Database Technology. Information Systems Frontiers 3, 319–338 (2001)
23. Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: Designing Business Processes with History-Aware Resource Assignments. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 101–112. Springer, Heidelberg (2013)
24. Domshlak, C., Hüllermeier, E., Kaci, S., Prade, H.: Preferences in AI: An overview. Artif. Intell. 175(7-8), 1037–1052 (2011)
25. Fishburn, P.C.: Utility theory for decision making. Wiley (1970)
26. Keeney, R.L., Raiffa, H.: Decisions with multiple objectives: Preferences and value tradeoffs. Cambridge Univ. Press (1993)
27. Agrawal, R., Wimmers, E.L.: A Framework for Expressing and Combining Preferences. In: SIGMOD Conference, pp. 297–306. ACM (2000)
28. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. IEEE Trans. Software Eng. 30(5), 311–327 (2004)
29. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. J. Artif. Intell. Res. (JAIR) 21, 135–191 (2004)
30. Kießling, W.: Foundations of Preferences in Database Systems. In: VLDB, pp. 311–322. VLDB Endowment (2002)
31. Chomicki, J.: Preference formulas in relational queries. ACM Trans. Database Syst. 28(4), 427–466 (2003)
32. García, J.M., Ruiz, D., Ruiz-Cortés, A.: A Model of User Preferences for Semantic Services Discovery and Ranking. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 1–14. Springer, Heidelberg (2010)
33. Cabanillas, C., García, J.M., Resinas, M., Ruiz, D., Mendling, J., Ruiz-Cortés, A.: Priority-Based Human Resource Allocation in Business Processes. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 374–388. Springer, Heidelberg (2013)
34. Sahinalp, S., Tasan, M., Macker, J., Ozsoyoglu, Z.: Distance based indexing for string proximity search. In: Proceedings of the 19th International Conference on Data Engineering, pp. 125–136 (March 2003)

35. Lin, D.: An Information-Theoretic Definition of Similarity. In: International Conference on Machine Learning, ICML 1998, pp. 296–304. Morgan Kaufmann Publishers Inc. (1998)
36. Fellbaum, C.: WordNet. Blackwell Publishing Ltd. (2012)
37. Navigli, R., Ponzetto, S.P.: BabelNet: Building a Very Large Multilingual Semantic Network. In: Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 216–225 (2010)
38. Friedrich, F., Mendling, J., Puhlmann, F.: Process Model Generation from Natural Language Text. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 482–496. Springer, Heidelberg (2011)
39. Leopold, H., Mendling, J., Reijers, H.A., Rosa, M.L.: Simplifying process model abstraction: Techniques for generating model names. Inf. Syst. 39, 134–151 (2014)
40. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
41. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP Framework: Identification of Correspondences Between Process Models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)
42. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Inf. Syst. 37(5), 443–459 (2012)
43. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R., Stuckenschmidt, H.: Probabilistic Optimization of Semantic Process Model Matching. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 319–334. Springer, Heidelberg (2012)
44. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 211–218. Springer, Heidelberg (2013)
45. Weidlich, M., Sagi, T., Leopold, H., Gal, A., Mendling, J.: Predicting the Quality of Process Model Matching. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 203–210. Springer, Heidelberg (2013)

Extending the Social Network Interaction Model to Facilitate Collaboration through Service Provision

Ourania Hatzi, Giannis Meletakis, Panagiotis Katsivelis, Andreas Kapouranis, Mara Nikolaidou, and Dimosthenis Anagnostopoulos

Department of Informatics and Telematics, Harokopio University of Athens,
9, Omirou str., 17778, Athens, Greece
{raniah,meletakis,pkatsiv,it20811,mara,dimosthe}@hua.gr

Abstract. Social network technology has been established as a prominent way of communication between members of an organization or enterprise. This paper presents an approach extending the typical social network interaction model to promote participant collaboration through service provision within an organization, towards the Enterprise 2.0 vision. The proposed interaction model between enterprise network participants incorporates their actual roles in the organization and enables the definition of custom relation types implementing custom policies and rules. It supports a complex mechanism for refined content propagation according to participant relations and/or roles. Moreover, the collaboration of participants to provide services and complete specific business tasks through Social Business Process Management is facilitated by enabling the execution of specific activities in each participant profile according to his/her actual role. To explore the potential of the proposed interaction model towards Enterprise 2.0, two prototype social networks, developed to serve different communities and needs, are discussed as case studies.

Keywords: Social BPM, Organization 2.0, Enterprise 2.0, Collaborative Communities, Roles and Relations, Social Network Technology, Service Provision and Task Coordination.

1 Introduction

Social networks have emerged as a new model for communication and interaction between individuals, as well as among members of communities or organizations (Acquisti & Gross, 2006). Currently, there are numerous social network platforms, both general-purpose, such as Facebook, and targeted to specific communities, such as MySpace. Social network platforms enable user communication in everyday social life, while they compete with each other in terms of popularity, by continuously offering enhanced functionality, advanced features, external service integration and connection with other social networks (Kossinets & Watts, 2006; Kumar, Novak & Tomkins, 2006; Liu, Maes & Davenport, 2006; Boyd & Ellison, 2007).

The utilization of Web 2.0 technologies, within organizations or enterprises, to promote collaboration between organization members, consists the Organization 2.0 or Enterprise 2.0 vision (Johannesson, Andersson & Wohed, 2008), aiming to explore

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 94–108, 2014.

how social networks may enhance intra-organization interaction. Corporations and organizations have incorporated social network technology either by using popular public social networks (Thompson & Doherty, 2006) or more often by utilizing private social networks (Geyer et al, 2008), aiming at more effective knowledge dissemination, intra-organization communication and efficient collaboration and service provision between their members (Grasso & Convertino, 2012).

Towards Enterprise 2.0, the potential of collaboration using private social networks has been explored for specific enterprises (DiMicco et al, 2008, Geyer et al, 2008, Motahari-Nezhad et al, 2012) and even for specific communities, such as healthcare/medicine (Boulos & Wheeler, 2007), learning/pedagogical (Hiltz, 1998; McLoughlin & Lee, 2007), and academic (Bermejo et al, 2012). Results are encouraging, as they indicate that novel technological concepts, such as the ones offered through social networks, tend to attract users and facilitate interaction also within the limits of a specific enterprise or community.

Companies encourage their employees to use their private social networks so they can strengthen weak ties with other employees through social interaction. They help organization members interact and contribute to work related issues (DiMicco et al, 2008), while leading to explore new forms of business interaction. At the same time, private social networks tackle emerging security and privacy issues. One of the most well known examples of such a private social network, is the SocialBlue (former Beehive) project (Geyer et al, 2008), created by IBM.

Collaboration within an organization, even utilizing private social networks, currently remains mostly at the informational or communicational level; that is, the social network infrastructure is used only for exchanging information, performing trivial tasks, such as arranging a meeting, or even share and collaboratively edit documents. There are certain efforts that attempt to provide enhanced functionality to assist collaboration, such as file sharing (Shami, Muller & Millen, 2011), targeting the collaborative production of content. Other works, such as (Bruno, 2012), (Hoegg et al, 2006) and (Ploderer, Howard & Thomas, 2010), explore how services offered by existing social networks can be utilized to promote collaboration between their participants. Moreover, the application of business models through social networks is also examined (Richter & Riemer, 2009).

Many current enterprise social network implementations are provided as SaaS platforms, providing services for information sharing among employees, such as activity streams, instant-messaging, file sharing, group creation, real-time document editing etc, and charge on a per-participant basis (Yammer, Zyncro, SocialCast, Jive).

Current trends indicate that enterprise social networks, in order to substantially improve the way enterprise members actually work, should not only facilitate information sharing but also help participants cooperate to complete specific business tasks. To elevate the impact of enterprise social networks, participants expect some sort of collaborative process execution, leading to Social BPM (Bruno et al, 2011).

Following BPM concepts, there are examples of social network platforms supporting participant roles. Tibbr enterprise social network, for example, offers discrete participant roles; however, they refer to social network administration privileges, not business process task assignment and execution privileges (Tibbr). SoCaM framework, implemented over HP enterprise social network, targets collaborative process execution, by supporting Case Management (Motahari-Nezhad et al, 2012). SoCaM represents processes and tasks as first class entities in the social network and assigns participant roles to tasks; however, these roles do not emerge from the actual participant roles present in the organizational structure of the enterprise at hand. Instead, SoCaM offers three specific roles which are the same for each task and depict the obligations of certain participants involved in this task.

What the aforementioned efforts have in common is that they attempt to adapt enterprise collaboration requirements to the existing social network interaction model and infrastructure. In contrast to the popular generic social network interaction model, whose success was based on its simplicity, we argue that in order to accommodate Organization 2.0, network participants should be able to interact and collaborate based on predefined roles, emerging from actual roles in the organization, enterprise or a specific-purpose community, where each participant is expected to contribute accordingly and complete certain tasks assigned to them.

Thus, a requirement emerges for the adaptation of a new collaboration model and the development of social network platforms supporting Organization 2.0, featuring complex interaction/collaboration models, multiple member roles and relations, and collaborative task execution based on discrete, predefined roles (Lewis, 2006; Oreilly, 2007; Vossen & Hagemann, 2007; Bruno et. al, 2011; Grasso & Convertino, 2012).

In order to support a way to execute business process steps using a SN platform, the existence of an extended interaction model is a prerequisite to enable the enforcement of restrictions according to business rules to the human-driven workflow that can be supported in a Social Network environment. In this paper we propose to extend the typical social network interaction model to explore the aforementioned requirements imposed on social network technology in order to promote Organization 2.0. Besides information sharing and collaborative editing, participants should be engaged to perform specific activities according to their role in the organization and current circumstances and cooperate with others based on enterprise policies and rules. This is achieved through collaborative application management and execution, leading to service provision. A social networking platform could support such functionality by ensuring the execution of applications on the participants' profiles, taking into account the participant role in the enterprise. Thus, role management should be integrated within the supported interaction model. Furthermore, relations, specializing the generic relations between participants of a social network, should be supported, in order to reflect the position and responsibilities of each member of the enterprise and facilitate role-based task assignment.

Based on the proposed extended interaction model, a social networking framework was developed for both enterprises and closed communities, facilitating the implementation of social networks that serve collaboration based on participant roles. To demonstrate the potential of the proposed concepts, two different social networks developed are discussed as case studies: a) Unity, an academic social network, aiming at promoting collaboration between the members of an academic institution, currently tested by members of the Department of Informatics and Telematics of Harokopio University of Athens and b) MedWeight SN, aiming at supporting a closed community of volunteers for weight maintenance using professional dietitian advice.

The rest of the paper is organized as follows: Section 2 presents the proposed interaction model to serve Organization 2.0. Section 3 discusses the application of the proposed model in both case studies. Conclusions and future directions reside in Section 4.

2 Extending Social Network Interaction Model

To effectively serve Organization 2.0 & Enterprise 2.0 a corresponding social network platform should accommodate several requirements. These requirements emerge by identifying elements regarding collaboration through service provision within an organization, which cannot be directly accommodated by existing social network interaction model features. These requirements are include:

- representation of discrete organization member roles
- incorporation of the organization co-operation model based on predefined relations
- information sharing and promotion of collaboration between organization members in a familiar yet intriguing way
- provision of services by specific organization members to others, based on their actual roles and relations, which in practice determine their privileges responsibilities in this specific environment
- coordination of collaborative tasks performed by cooperating organization members
- integration of services offered by external systems through a unified environment

Such a social network could be developed within the limits of a single organization, or it could also be expanded to include multiple organizations on a regional, national or international level, without affecting the underlying collaboration model.

Existing social networking platforms do not discriminate between participants or relations between them. They are based on a simple interaction model: participants interact with others with no restrictions and they may establish between them a single generic kind of relation with specific semantics, for example *friend*. Current generic public social network models feature only the Participant, Profile and Group entities of the UML model presented in Fig. 1. Requirements as the ones discussed previously cannot be covered by existing public social networks, since the generic interaction model offered by them cannot be adapted to reflect participant organization, while there are also security and privacy concerns. To fully accommodate the goals of an organizational / enterprise social network existing social networking technology should provide an interaction model with enriched semantics, as explained in the following and summarized in Fig. 1.

2.1 Basic Interaction Entities

2.1.1 Participant Roles and Relations

The interaction model serving Enterprise 2.0 features discrete roles for participants, corresponding to their actual position and responsibilities in the organization or enterprise. Roles can determine possible relations between participants. The decision about how specific the roles should be is based on whether further specialization affects the emerging relations. Roles also determine additional data stored in the profile for each participant. Roles can be used to either assign tasks to participants bearing a specific role or indicate the role a participant should have in order to be able to execute specific tasks; combined with relations they direct the flow of information.

Both organization and social relations are supported. Organization relations can be either unidirectional, indicating that an organization member receives services from another member, or bidirectional, indicating that the members cooperate to achieve certain tasks. When a relation exists, the object of the relation receives updates, posts and material published to the corresponding stream of the subject member profile, and benefits from specific services provided by them.

On order not to limit the free interaction and flow of information, as it emerges in a social network context, the social relation fellow is also defined. It is a bidirectional relation denoting that two participants are socially connected and it corresponds to the generic relation offered by existing social networks. This relation may exist between any two participants, regardless of their roles; if the relation exists, each participant receives posts, updates and content published to the social stream of the other. Social relations do not affect workflows and task executions; however, they play an important role in the social network model as they are expected to strengthen ties between participants and encourage cooperation, thus leading to improved interaction.

2.1.2 Streams

The most common operation that a participant performs in a social network is publishing content, which can be of a variety of types, such as links, texts, files, multimedia etc. Published information is propagated in the form of a stream to all participants related to the publishing entity, who receive notifications and updates about the publication, urging them to review it and possibly contribute to it, as dictated by the notion of collaborative content in Web 2.0 (Anderson, 2007).

In an organization, specific streams should be defined based on participant roles and relations. Apart from the intra-organization member relations, the social aspect of the community should not be dismissed; therefore, each member may develop a social relation with any other member of the community, regardless of their roles in it. At the same time, a clear separation between them should be maintained, thus a more complex propagation mechanism is introduced incorporating discrete streams.

Along with streams, the proposed model also defines propagation rules indicating which participants receive the publications directed to each stream. While the publisher maintains a unified stream on the corresponding profile, the propagation of published information does not take place for all publisher's contacts indiscriminately, but is based on the type of their relation with the publisher, determining the stream they receive. The combination of discrete participant roles, multiple streams, extended relations and rules governing the propagation of content successfully achieves the separation between the organizational and social information shared within the organization.

2.1.3 Groups

The combination of roles, relations and streams does not fully facilitate fine-grained content propagation; therefore, a more elaborate mechanism for content delivery is proposed, through groups. Groups are arbitrary sets of contacts that any social network member can create and modify dynamically. Each group has a specific name, and the member who creates it, as its owner, has control over membership of other participants, which may join or leave the group. All members and only the members of a group can publish content in the group, while the owner maintains control over all posts. Each publication to a certain group belongs to a corresponding custom, ad-hoc group stream and is propagated to all members of this group. Groups are generic enough to serve multiple purposes in a complex organization.

2.2 Collaboration and Task Coordination

Collaboration in a typical social network is performed through exchange of information and notifications in a distributed fashion (Gross & Koch, 2006). In addition to sharing content and notifications through discrete streams and groups, the proposed social network model supports the provision of specific services and enables its participants to complete specific business tasks in collaboration with other participants (Dengler et al, 2010).

Services may be provided by cooperating applications executed in a specific participant profile. Typical social networks enable applications to be executed on the user profile. These applications usually read data from the user profile and may invoke external applications through a web service interface. They also have access to store data in the user profile. In order to ask for services rather than information from another participant, a more sophisticated communication mechanism is required, facilitating information exchange between applications executed on different profiles.

We propose treating all services, either simple or more complex ones, as tasks consisting of specific steps (e.g. activities) which may be performed by participants of a specific role – a policy that emerges from the actual enterprise organization. Each activity corresponding to a specific task step is handled as an application, which may only be executed in the profile of a participant having the proper role, and may involve the invocation of external services to be completed (Hatzi et al, 2012). Each application, as any other program, needs specific input data to start execution and, when executed, produces output data. The coordination of tasks, e.g. the conditions under which specific activities may be executed, is performed based on the available input data of applications implementing the specific activities. An application implementing a specific activity cannot start its execution until all its input data are available. This data may be part of the user data stored in the profile the application is executed on, or produced as output data of other applications, which may be executed on the same profile, e.g. by the same user, or more frequently on external profiles corresponding to users having the proper role to invoke those applications.

Evidently, in order for collaborative tasks to be supported, inter-application communication executed on different profiles must be enabled. Based on available social network technology, applications may access and store data in a stream specific for this purpose, the Activity Stream, which is private to applications and not visible to participants. While the task is progressing, proper notifications are issued to collaborating participants, urging them to be involved for their part in the task. Obviously, the participants collaborating for a specific task must be properly associated with corresponding relations.

The proposed extended interaction model is presented in Fig. 1 using UML notation. The basic generic social network interaction model features only the participant, profile and groups entities. The proposed extensions concern the assignment of roles to participants, which attach an additional properties container to their profiles, as well as the specialization of the generic relations, to indicate more refined interaction structure. The specialized relations are unidirectional or bidirectional and defined between specific roles. The rest of the entities, i.e. streams, applications and notifications, take into account participant roles and relationships in order to implement organization policies regarding rights and restrictions. These extensions have as a consequence that applications are allowed to be executed only by participants belonging to a specific role – this enables the representation of enterprise tasks assigned to collaborating participants.

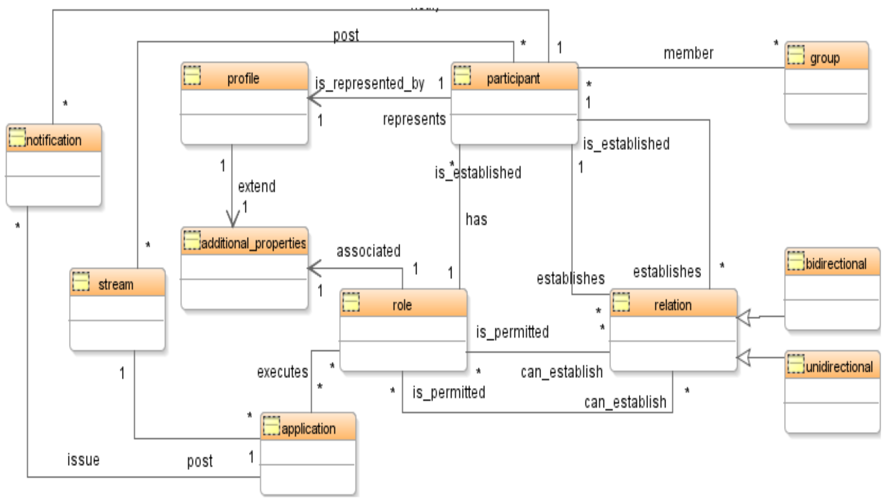


Fig. 1. Enterprise 2.0 extended model

3 Case Studies

3.1 Supporting Organization 2.0 in an Academic Institution

Towards Oranization 2.0, the Unity SN was developed to enable collaboration between the members of an academic community. It was based on Google OpenSocial framework and is currently deployed in the Department of Informatics and Telematics of Harokopio University. The case study is briefly presented focusing on task coordination features to demonstrate the impact of the definition of discrete roles and relations, which govern participant responsibilities, to task execution and service provision. Without them task coordination could not be effectively supported.

3.1.1 Interaction Model

Each member of an academic organization has specific responsibilities, may represent specific University services, such as the University Library or the Student Admission Office, and may perform specific tasks to serve other community members.

Following the proposed model, the following roles can be identified:

- Student: undergraduate students, postgraduate students and PhD candidates
- Teaching staff: faculty members and additional teaching staff
- Administrative staff: University employees that could potentially provide services to community members, i.e. Admission Office employees, Library employees, Erasmus office employees, Computer Center employees, etc.

Based on these roles, the following organizational relations are defined:

- Tutor: a unidirectional relation declaring that a student is being taught / supervised by a member of the teaching staff.
- Facilitator: a unidirectional relation declaring that a Student or Teaching staff member is served by a member of the Administrative Staff.

The social relation fellow is also defined, between any two participants.

In the academic environment, groups may be formed for courses, or any other special interest group, such as the Open Source Community.

The combination of discrete participant roles, multiple streams, extended relations and rules governing the propagation of content successfully achieves the separation between the academic and social aspects of the academic community. For example, a professor may announce his office hours only to participants who are his students and not to all his contacts, while at the same time he may publish information about an upcoming film festival to all his social contacts.

Fig. 2 depicts an example of a participant profile in Unity. The profile shows information concerning the participant, his role in the social network and his contacts, which can also be viewed by relation category. It also contains recent activity, posts and notifications received by the participant. Finally, it features the "App Drawer", i.e. the application deployment space.

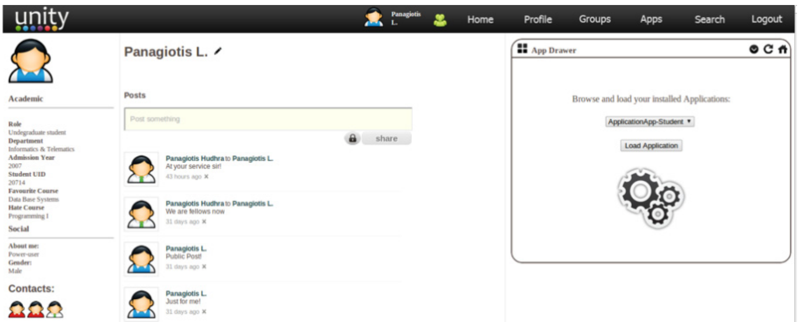


Fig. 2. An example of a Unity Participant profile

The specific participant contacts viewed by relation category are depicted in Fig. 3 (left part), while posts and notifications received by the participant are presented in Fig. 3 (right part).

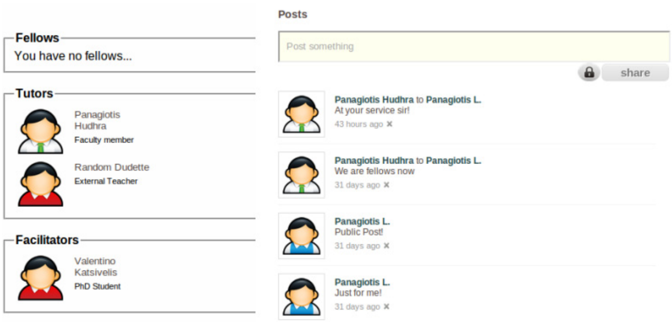


Fig. 3. Participant contacts (left) and notifications (right)

3.1.2 Collaboration and Task Coordination

As a task coordination example we consider the graduation process. In order to be eligible for graduation, a university student must fulfill the following requirements:

- All necessary courses have been successfully completed.
- The degree thesis has been successfully examined and submitted to the University Library, as indicated by the corresponding certificate.
- All books borrowed from the University Library must have been returned.
- The student ID and transportation card have been submitted to the Department Secretariat.

The student can subsequently fill out a graduation application form and submit it to the Department Secretariat, who confirms that all requirements are valid and notifies the student of the graduation ceremony date.

Such a process could be modeled using a BPMN diagram focusing on the discrete activities performed to accomplish this task, as presented in Fig. 4.

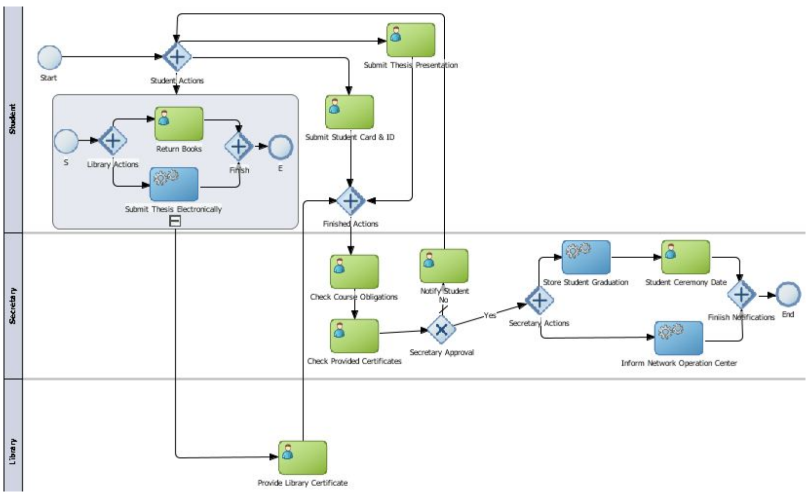


Fig. 4. Graduation Process described in BPMN

However, in the context of the social network, students may view graduation process as a set of certificates they have to gather in their profile before graduating, adopting the Case Management approach as discussed in (Motahari et al, 2012). In such an approach, the student as a Unity participant is not guided to perform specific steps; they are only notified of which data (certificates) are missing from their profile to be eligible for graduation. Some of these certificates cannot be issued by students, thus they need to notify Library and Secretary personnel having the proper authority (and consequently, role) to issue the certificates for them (e.g. are authorized and eligible to serve them since they are their facilitators). Certificates are issued running the corresponding application in the participant profile having the authority to execute the application according to their role in the academic community.

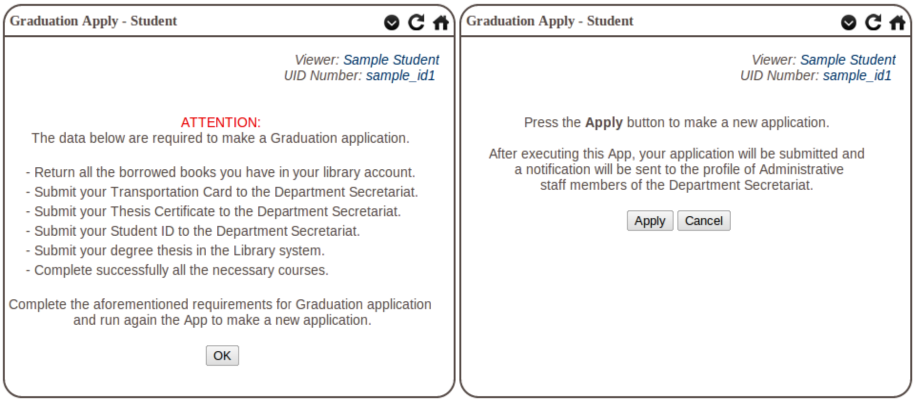


Fig. 5. Graduation Application execution

The student starts the graduation process by installing and executing the Graduation Application. The application checks if all corresponding certificates are available, as shown in Fig. 5. If not, the student should collect them, otherwise the graduation application is submitted to the department secretariat.

An administrative staff member of the Department Secretariat may execute the GraduationCeremony Application, to notify all applicants of the graduation ceremony date, as depicted in Fig. 6. Note that this application is available only to participants with an Administrative Staff role.

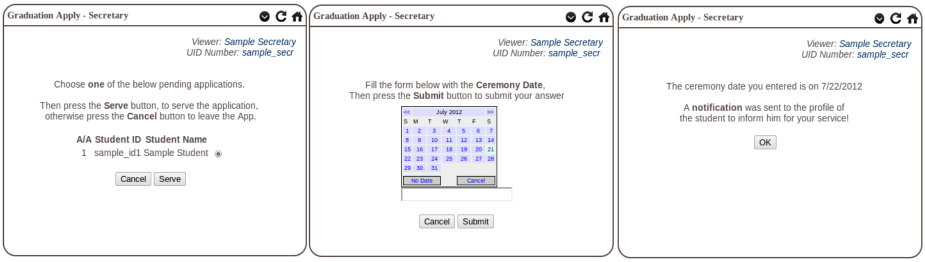


Fig. 6. GraduationCeremony Application execution

What happens when the student may not submit the application? In this case, students should collect all necessary certificates as indicated by the Graduation Application.

For example, in order to confirm that the student has returned all borrowed books to the University Library, the LibraryBookAccountStudent Application must be executed, as depicted in Fig. 7. The application requests the student's Library Identification Number and issues a notification to the administrative staff of the University Library. This application registers data such as the LibraryIdentificationNumber and the NumberOfBooks to the Activity Stream, in order to be able to communicate with other applications.

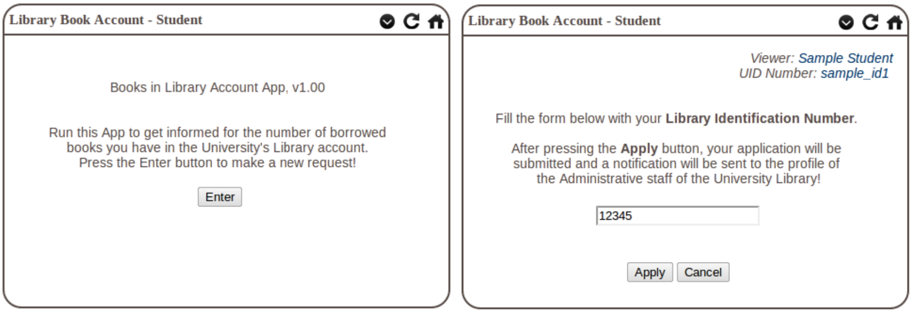


Fig. 7. LibraryBookAccountStudent Application execution

A member of the Library administrative staff, which is connected by the facilitator relation with this specific student, must then execute the collaborating BookAccountLibrary Application, which shows all pending requests. This application requests a username/password for the Library Information System by the member of the Library administrative staff, invokes the appropriate API to obtain the number of books that the student has borrowed, updates the ActivityStream and issues a notification to the student. The process is depicted in Fig. 8.

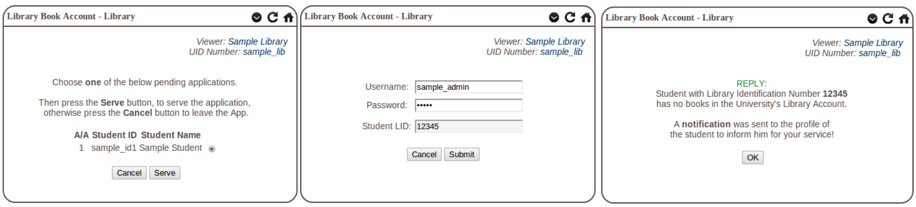


Fig. 8. BookAccountLibrary Application execution

The definition of discrete roles in the interaction model, based on actual organization member roles, enables application execution based on the role and responsibilities of each academic community member, while relations define the circumstances under which a specific member of the academic community may serve another and help them collaborate to complete a task.

3.2 Organization 2.0 Features Useful in a Collaborative Community

MedWeight Social Network aims at supporting volunteers to maintain their weight and eat healthy for a period of three years. The network aims to build ties between volunteers participating the network, to help each other maintain their weight and exchange healthy eating habits and recipes. Furthermore, advice and guidance from dietitians may be provided without treating the volunteers as "clients". It is a research project from the Department of Nutrition and Dietetics Science of Harokopio University of Athens, which is involved in the study. It is currently deployed in its prototype phase using Python and Django technology. The user interface is currently in Greek. Although this is a private social network targeting a closed community, it still features the basic characteristics of content dissemination and service provision based on predefined, discrete participant roles and relations.

More specifically, the following roles were identified:

- Volunteer: a person who takes part in the study and wants to benefit but has no expert knowledge concerning diets and nutrition
- Dietitian: an expert scientist that provides services and feedback to users of the role Volunteer

Based on these roles, the following relations were defined:

- Instructor, which is a unidirectional relationship from a volunteer to a dietitian
- Fellow, which is a bidirectional social relationship and can be defined between any two members of the community

A screenshot of a participant profile is depicted in Fig. 9

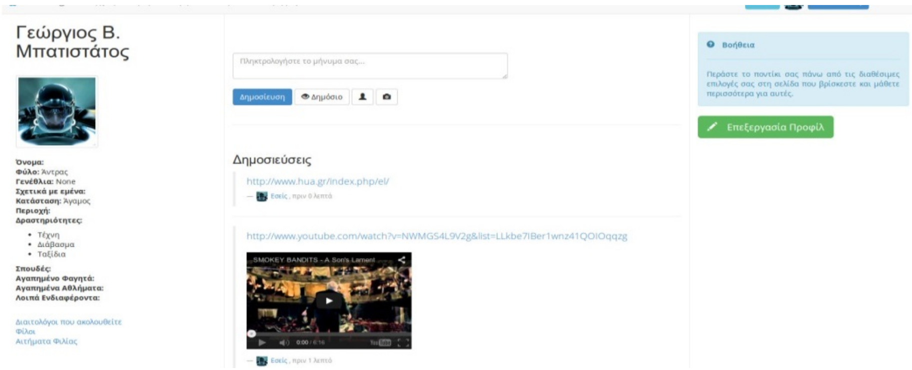


Fig. 9. MedWeight social network participant profile

As before, roles and relations are used for content propagation, as well as for application execution, leading to task completion. In such closed communities, role-based content propagation is important, as certain participants belong to roles indicating "expertise" or "authority", enabling other participants to establish trust to the integrity of the content they post and act accordingly.

As a task example, the weight maintenance application is briefly presented. Volunteers may daily register measurements of their weight, running such an application in their profile. With each measurement, the application calculates certain dietetic factors, such as Body Mass Indicator. If any of these factors have exceeded a certain limit, a notification is issued to dietitians chosen by the volunteers as their instructors. Consequently, the dietitian can provide personalized feedback and expert advice to the volunteer, properly directing the proper content to him/her. A screenshot of this application is depicted in Fig. 10.

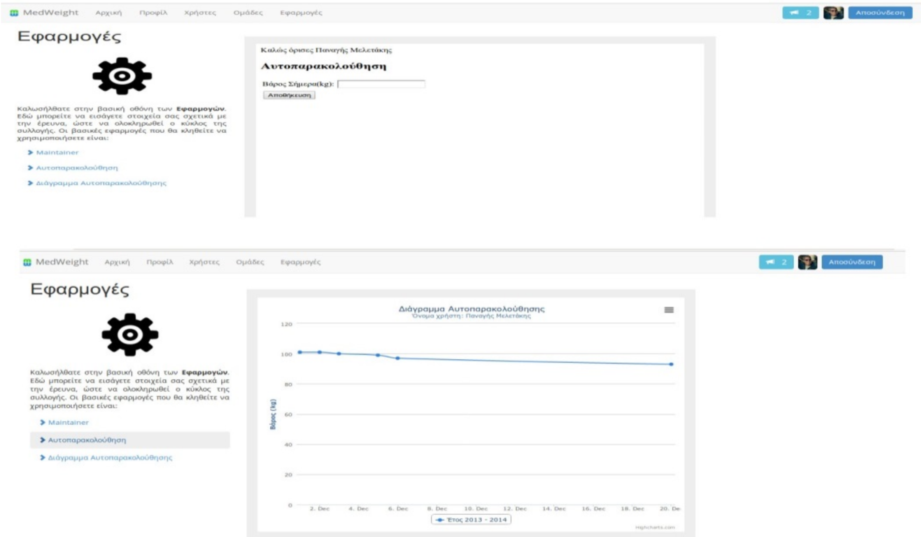


Fig. 10. Weight maintenance application – weight insertion & report

The extension of the original interaction model with roles, in the case of communities, enables to impose restrictions on application execution. For example, in this particular case, a volunteer in the social network will receive feedback on their weight maintenance only from expert dietitians and not other volunteers, as non-expert advice might be anywhere between misleading and dangerous.

4 Conclusions

Current social network technology and corresponding interaction mechanisms cannot effectively serve the Organization 2.0 & Enterprise 2.0 vision, since business task coordination based on predefined organization roles is not a supported feature. To this end, the typical social network interaction model was extended and a corresponding social network platform supporting it was developed.

The support of a single, simple relation offered by typical social networks is not adequate to model restrictions on the interaction between enterprise members collaborating to perform a task. The definition of discrete roles and relations enhances the description of workflows corresponding to specific business tasks, which are completed by collaborating participants. Taking into account different roles and relations, multiple content streams may be defined, facilitating improved control over the propagation of content to participants.

Future work concerns a more elaborate mechanism for defining applications in the proposed social network model that will be able to handle semantics through ontologies or folksonomies. Such an extension would encourage the development and integration of applications by third parties, permitting the proposed model to be used effectively for e-administration or e-government, involving multiple organizations, as well as for inter-enterprise collaboration. Application and experimentation with the proposed collaboration model in other collaborative communities and enterprises featuring discrete roles and relations, following the concept of Enterprise 2.0, will also be explored.

References

1. Acquisti, A., Gross, R.: Imagined communities: Awareness, information sharing, and privacy on the Facebook. In: Golle, P., Danezis, G. (eds.) Proceedings of 6th Workshop on Privacy Enhancing Technologies, pp. 36–58. Robinson College, Cambridge (2006)
2. Anderson, P.: What is Web 2.0? Ideas, technologies and implications for education. JISC Technology and Standards Watch (2007)
3. Bermejo, J.A.A., Bravo, C.B., Mateos, M.J.R., Piera, J.R.: Social Networks in the Higher Education Framework - Understanding the University as an Organization: Inlumine, Our Study Case. In: Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions, pp. 805–824. IGI (2012)
4. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication 13(1), article 11 (2007)
5. 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(4), 297–326 (2011)
6. Bruno, G.: An Approach to Defining Social Processes Based on Social Networks. In: Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions, pp. 272–286. IGI (2012)
7. Boulos, M.K., Wheeler, S.: The emerging Web 2.0 social software: An enabling suite of sociable technologies in health and health care education. Health Information & Libraries Journal 24(1), 2–23 (2007)
8. Dengler, F., Koschmider, A., Oberweis, A., Zhang, H.: Social Software for Coordination of Collaborative Process Activities. In: Muehlen, M.z., Su, J. (eds.) BPM 2010 Workshops. LNBIP, vol. 66, pp. 396–407. Springer, Heidelberg (2011)
9. DiMicco, J., Millen, D.R., Geyer, W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for Social Networking at Work. In: Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work, CSCW 2008, pp. 711–720. ACM, New York (2008)
10. Elgg, Open Source Social Networking Engine, `http://elgg.org/` (accessed September 2, 2012)
11. Facebook, `https://www.facebook.com/` (accessed August 24, 2011)
12. Geyer, W., Dugan, C., DiMicco, J., Millen, D.R., Brownholtz, B., Muller, M.: Use and Reuse of Shared Lists as a Social Content Type. In: CHI 2008 Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, pp. 1545–1554. ACM, New York (2008)
13. Grasso, A., Convertino, G.: Collective Intelligence in Organizations: Tools and Studies. Computer Supported Cooperative Work (2012) 21, 357–369 (2012)
14. Gross, T., Koch, M.: Computer-Supported Cooperative Work - Concepts and Trends. In: Proc. 11th Conf. of the Association Information and Management (AIM), Bonner Koellen Verlag. LNI P-92, pp. 165–172 (2006)

15. Hatzi, O., Nikolaidou, M., Katsivelis-Perakis, P., Hudhra, V., Anagnostopoulos, D.: Using social network technology to provide e-administration services as collaborative tasks. In: Kő, A., Leitner, C., Leitold, H., Prosser, A. (eds.) EDEM/EGOVIS 2012. LNCS, vol. 7452, pp. 216–230. Springer, Heidelberg (2012)
16. Hiltz, S.R.: Collaborative Learning in Asynchronous Learning Networks: Building Learning Communities. In: 3rd WebNet World Conference of the WWW, Internet, and Intranet Proceedings, pp. 1–8. New Jersey Institute of Technology (1998)
17. Hoegg, R., Martignoni, R., Meckel, M., Stanoevska-Slabeva, K.: Overview of business models for Web 2.0 communities. In: Proc. Workshop Gemeinschaften in Neuen Medien GeNeMe Dresden, pp. 33–49. TUDPress, Dresden (2006)
18. Jive, `http://www.jivesoftware.com/`
19. Johannesson, P., Andersson, B., Wohed, P.: Business Process Management with Social Software Systems - A New Paradigm for Work Organisation. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008 Workshops. LNBIP, vol. 17, pp. 659–665. Springer, Heidelberg (2009)
20. Kossinets, G., Watts, D.J.: Empirical Analysis of an Evolving Social Network. Science 6 311(5757), 88–90 (2006)
21. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: Proceedings of 12th International Conference on Knowledge Discovery in Data Mining, pp. 611–617. ACM Press, New York (2006)
22. Lewis, D.: What is web 2.0? ACM Crossroads 13(1), 3 (2006)
23. Liu, H., Maes, P., Davenport, G.: Unraveling the taste fabric of social networks. International Journal on Semantic Web and Information Systems 2(1), 42–71 (2006)
24. McLoughlin, C., Lee, M.J.: Social software and participatory learning: Pedagogical choices with technology affordances in the Web 2.0 era. In: Proceedings ASCILITE Singapore 2007, pp. 664–675 (2007)
25. Motahari-Nezhad, H.R., Bartolini, C., Graupner, S., Spence, S.: Adaptive Case Management in the Social Enterprise. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) Service Oriented Computing. LNCS, vol. 7636, pp. 550–557. Springer, Heidelberg (2012)
26. Myspace, `http://www.myspace.com/` (accessed August 24, 2011)
27. OpenSocial, `http://code.google.com/apis/opensocial/` (accessed August 24, 2011)
28. Oreilly, T.: What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. International Journal of Digital Economics (65), 17–37 (2007)
29. Ploderer, B., Howard, S., Thomas, P.: Collaboration on Social Network Sites: Amateurs, Professionals and Celebrities. Computer Supported Cooperative Work (CSCW) 19(5), 419–455 (2010)
30. Richter, A., Riemer, K.: Corporate Social Networking Sites – Modes of Use and Appropriation through Co-Evolution. In: ACIS 2009 Proceedings, pp. 722–732 (2009)
31. Shami, N.S., Muller, M.J., Millen, D.R.: Browse and discover: Social file sharing in the enterprise. In: CSCW 2011 Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, pp. 295–304. ACM, New York (2011)
32. SocialCast, `http://www.socialcast.com/`
33. Thompson, J., Doherty, B.: The diverse world of social enterprise: A collection of social enterprise stories. International Journal of Social Economics 33(5/6), 361–375 (2006)
34. Vossen, G., Hagemann, S.: Unleashing Web 2.0: From concepts to creativity. Morgan Kaufmann (2007)
35. Yammer, `https://www.yammer.com/`
36. Zyncro, `http://www.zyncro.com/`

A Framework for Synchronizing Human Behavior, Processes and Support Systems Using a Socio-technical Approach

Ilia Bider[1,2] and Stewart Kowalski[1]

[1] DSV Stockholm University, Stockholm, Sweden
[2] IbisSoft AB, Stockholm, Sweden
{ilia,stewart}@dsv.su.se

Abstract. The paper suggests a framework for achieving alignment between a process and its external and internal environment. The framework consists of two components. The first component concerns alignment between the process and its external environment - business environment in which the process functions or is to function. The second component concerns alignment between the process and its internal environment the most important part of which are people participating in the process. The second component, which is in the focus of the paper, is based on the socio-technical view on information systems. The framework is aimed to move the focus of business process reengineering/improvement from local optimization through the use of technology to the needs of satisfying business goals, and fostering human capital that is needed to achieve them.

Keywords: business process, human factors, socio-technical, work system.

Culture eats strategy for breakfast
Attributed to *Peter Drucker*

1 Introduction

The epigraph "*Culture eats strategy for breakfast*" chosen for this paper means that no good decision on strategy can be made without considering the human aspects of the enterprise or organization. This statement is in high degree applicable to the area of business process development, and development of business process support (BPS) systems. It is counterproductive to design the best possible process without considering how the people that participate in it would accept and use it. The same is true for BPS systems. A system that contradicts the process participants' view on how to run the process can be rejected or misused if imposed by force. The goal of this paper is to draft a framework that could help in synchronizing processes, BPS systems and organizational culture in order to achieve strategic goals of the enterprise.

The framework consists of two parts, the first part connects external environment to the properties of the process suitable for it. The external environment is defined in

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 109–123, 2014.

the terms of positioning in the market, while properties of the process are defined via proportion of different kinds of rules used in the process. This part of the framework has been built based on the works that belong to the field of management [1,2,3,4] and our own work on the application of deontic logic to the domain of business processes [5].

The second part of the framework connects properties of the process to the internal environment which consists of human process participants, technology and techniques used by process participants in the frame of the process. This part of the framework has been built based on the socio-technical perspective [6] on work systems [7] in the Information Systems (IS) discipline, and works on organizational culture [8,9].

The rest of the paper follows the following plan. In section 2, we give an overview of the research method and background of our research. In Section 3, we introduce a categorization of processes based on the proportion of different kind of rules used in defining and running the process. In Section 4, we investigate the suitability of different process categories to the external environment – positioning in the market. In Section 5, we discuss alignment between the process categories and the internal environment - organizational culture, techniques and technology used. Section 6 is devoted to discussing potential usefulness of the framework, while Section 7 summarizes the results and discusses steps needed for the framework validation.

2 Method and Background

As follows from the introduction, our stated task in this research is the development of a framework that can be useful in aligning processes with their internal and external environment. When coping with this task we used Design Science Research (DSR) approach [10], more specifically its interpretation according to [11]. In this interpretation, DSR, as a way of generating and testing hypotheses for generic solutions, requires researchers to act in two different worlds: (a) the real world of specific problems and solutions in local practices, and (b) the abstract world of generic situation, problems and solutions. There is no specific requirement on the order in which the movement is completed. Researchers can start with searching a solution for a practical problem before or after generalizing it, or they can start with building a solution for the problem "unknown", and then finding for what purpose a solution is good. The main point is to have in the end a description of the triad <generic-situation, generic problem, generic solution> and one or several test cases that shows that the generic solution applied in a specific situation can solve the instantiation of the generic problem that exist in this situation.

The generic problem our solution is aimed to solve is defined as aligning the processes to their environment. The solution consists of using the framework suggested in this paper for analysis of the situation-as-is or to-be in order to find out whether the process is aligned to the environment or not, and introduce measures to ensure alignment. This particular paper concerns the design of the framework as such, how to validate the usefulness of this framework is discussed in Section 7.

The background of this research consists of two parts. Firstly, it is our own experience of building and, especially, introducing BPS system in the practice of organizations that has been partly reported in [12,13]. During this experience a number of practical questions have been raised. For example, why some BPS systems are easy to introduce, while introduction of others takes too much efforts and time. Another question of the same sort is why the same kind of a BPS system can be accepted in one part of the organization, and for one kind of processes, while rejected in another part, or for other processes.

The second part of the background is research and practical literature. Here, in the first place, we did not use literature from the business process management domain, but from management theory and Information Systems (IS) theory. The works from the management theory that we used are works on business cycles [1,2], classical works of Perrow on comparative analysis of organizations [3], works on balance between efficiency and effectiveness [4] and on organizational culture [8,9]. From the IS theory, we used the concept of work systems [7], and a socio-technical view on information systems from [6].

The foundation for our design might look arbitrary for the reader. In connection to this, we need to point out that the goal of this work is to suggest ***a*** heuristic framework, as to the best of our knowledge none exists for the moment. We are not attempting to create ***the*** best possible framework.

3 Process Categories

There are many definitions of the notion of business process, each of them highlighting different aspects, as described by [14]. In this paper, we need to define a process in a general way to be able to deal with different categories of business processes. Firstly, we want to highlight only one particular side of a business process – standardization or structuring of work. Secondly, we want to highlight that term business process encompasses two distinct concepts: business process type and business process instance (or case). Considering both goals we give the following working definitions for both terms:

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

The dichotomy type/instance is accepted in all branches of BPM literature. However, different terms are used to express this dichotomy. For example, the type can be referred to as a process model. Another convention that is often used is to employ term process to refer to the type, while case or run to denote the instance. In this paper we are focused on considering the process types, and therefore will use term *process* to denote BPT.

The plan/template that corresponds to a given process type can reside in any combination of the following (see a similar discussion about work systems in [7]):

- In the minds of staff members who participate in instances of this type of business process (i.e., tacit knowledge). This knowledge guides the participants in the process in what is permitted, obliged, and/or prohibited, without requiring them to reflect about it.

- As written documents, including process maps and other kinds of process descriptions (i.e., explicit knowledge) stored on paper or electronically, e.g., in the form of web-based hypertext. These documents contain explicit instructions of what is permitted, obliged, and/or prohibited.
- In software systems/services used to support the running of the instances of the process (i.e., built-in or embedded knowledge). The usage of such systems forces process participants to carry out some actions in a certain way and/or in a certain order.

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

The way of defining the template can differ dependent on the nature of the process and/or convention used in the given organization. Independent of how a given BPT is represented, we can consider it as consisting of a number of explicit or tacit rules that concern:

- *When* - what situations require action (how to detect the situation)
- *What* - what needs to be done in certain situations, i.e. goals to achieve
- *How* – what actions should be completed to achieve the goals
- *Who* – who is to complete the actions.

There are various ways of defining the rules. To categorize the type of rules used in a particular process independent of the way they are actually formulated, we will use the classification suggested in [5] that was inspired by deontic logic [15]. According to [5], the rules can be divided into four types:

1. *Obligations* – what must be done, independent of the will or judgment of the process participants (e.g. prescribe by law)
2. *Recommendations* – what is normally recommended, but could be overridden by process participants in a particular process instance
3. *Negative recommendations* – what is not recommended, but could be employed by process participants in a particular process instance
4. *Prohibitions* - what must never be done, independent of the will or judgment of the process participants (e.g. prescribe by law).

Note that *Obligations* and *Prohibitions* are essentially different from *Recommendations* and *Negative recommendations*. While the first two establish constraints on what is allowed to do, the second two do not impose any constraints. Note also that constraints can be of two types externally imposed, e.g. physical or juridical laws, or internally introduced. The latter can be a mutual agreement by process participants, or a decision of the upper management.

Dependent on the proportion of different type of rules in a given process type, we can roughly identify four categories of processes *Loose*, *Guiding*, *Restrictive* and

Table 1. Process categories

Rules category \ Process category	**Loose**[1]	**Guiding**[2]	**Restrictive**[3]	**Strict**[4]
Obligations	Some	Some	Many	Many
Recommendations	None	Many	Some	None
Negative recommendations	None	Many	Some	None
Prohibitions	Some	Some	Many	Many

[1] All that is not prohibited is allowed. Some obligation exists.
[2] Differs from *Loose* by presence of guidelines for most typical situations.
[3] Small room for actions that falls outside Obligation + Recommendation and not Prohibited.
[4] All that is not covered by obligations is prohibited.

Strict according to the Table 1. Note that the *flexibility* of the processes decreases while going from *loose* to *strict*, while *rigidness* increases.

4 Alignment between a Process and Its External Environment

There are a number of business and economic cycles theories which provide explanations as to why external economic environment for a company or organization will always be a in a state of flux [1]. There are also some, like [2], that suggests the innovation, both technological and institutions, being the basic drivers of these cycles. Consequently an organization needs a framework to align their business processes with the environment, dependent on their position in the cycle.

For building such a framework, we adapt the uncertainty/exceptions matrix [3]. The resulting matrix, depicted in Fig 1, identifies four distinct types of the process external environment called Market Positions (MPs). MPs are related to the state of the market in which an enterprise can be active, as shown in Fig. 1. Each MP requires different ratio between flexibility/rigidness of the process:

MP1. *Exploration* requires flexible processes and collaboration inside loosely structured teams. Strict division of responsibilities and other strict constraints imposed on the processes can be harmful at this stage.

MP2. *Standardization* requires division of responsibilities, i.e. specialization, and establishing templates for different types of business processes. Without standardization, chances of being able to handle the expansion will be diminished.

MP3. *Optimization* requires highly optimized standard processes for minimizing cost and increasing the profit to stay competitive in the existing market.

MP4. *Freezing* does not warrant any extra efforts; it can employ rigid processes with less people and less specialization than when in MP3 but with rigid routines. Employing flexible processes at this stage can be a waste, as the market is on the decline.

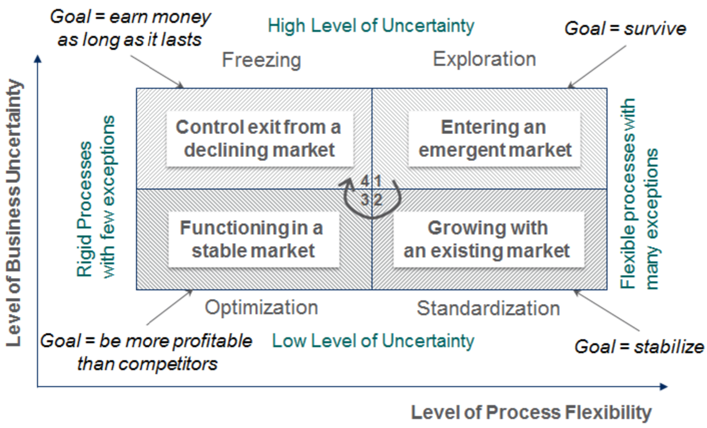


Fig. 1. Market positions – MPs. Adapted from [3]

Note that in practice, an enterprise may find itself in different MP regarding its different products/services segments. This makes the overall picture more complex requiring coexistence of different categories of processes.

Let us consider now which categories of processes suit particular MPs. According to Fig. 1, an organization has different goals in each of the MP quadrant; therefore the process category needs to be related to the goal of the corresponding MP. To relate the goals and process categories, we use the effectiveness/efficiency matrix suggested in [4], see Fig. 2. In tis matrix, effectiveness is defined as doing the right things – fit to the situation at hands, while efficiency as doing the things right – without waste. Relating efficiency to the process categorization in Table 1, we can conclude that moving from loose category to strict *potentially* provides better efficiency as it eliminates disturbances connected to making arbitrary decisions in the frame of flexibility allowed in each category of processes. *Real* efficiency, naturally, depends on the particular process design.

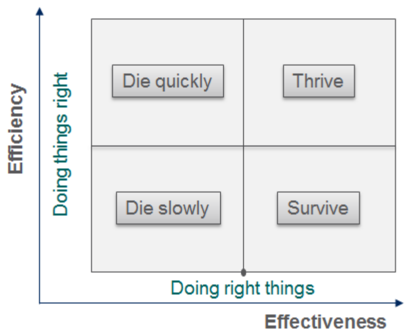


Fig. 2. Efficiency/Effectiveness matrix adapted from [4]

As far as effectiveness of each category is concerned, we consider it dependent on the external environment, and thus on MP in which a given organization operates at the moment. Combining matrixes in Fig. 1 and 2 and relating the combination to the table 2, we get a nested matrix in Fig. 3 that shows the level of efficiency/effectiveness for each process category and each MP. The most suitable category for each MP is denoted via a diamond in the quadrant that corresponds to this MP, the levels of efficiency/effectiveness of other categories are denoted via circles.

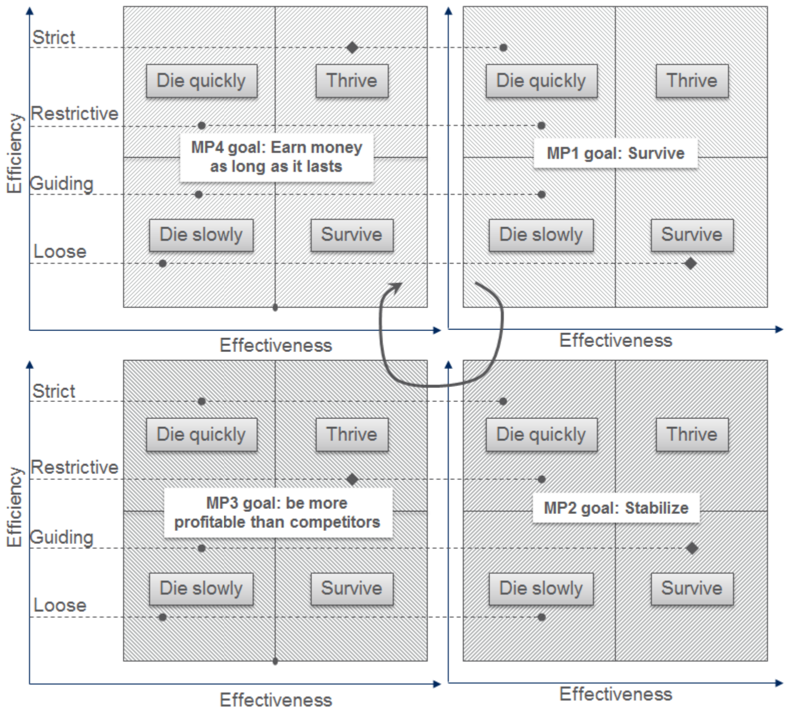


Fig. 3. Suitability of process categories for different MPs based on Fig.1 and 2 and Table 1

Summarizing the nested matrix in Fig. 3, we get the guidelines for alignment between a process and external environment depicted in Table 2. A cross in a cell shows alignment between the process category and MP.

Table 2. Alignment between processes and external environment

Process category / MP	**Loose**	**Guiding**	**Restrictive**	**Strict**
MP1: Exploration	X			
MP2: Standardization		X		
MP3: Optimization			X	
MP4: Freezing				X

5 Alignment between a Process and Its Internal Environment

In this section, we establish the rules of alignment between a process and its internal environment based on the socio-technical view on a work system suggested in [6].

5.1 Components of Socio-technical Systems

The socio-technical view from [6] identifies four components in a work system - people, structure, tasks, and technology. In our case, as a work system we consider all people engaged in a given process type using a particular technology and completing particular tasks when running process instances of the given process type. In our case, the process type creates a structure in which people operate in the work system.

Each component of the work system affects the overall system behavior in its own way, thus the needs arise for the components to be aligned in order for the work system to function properly. To formulate the rules of alignment, we classify the components of the socio-technical system independently of each other, and then consider interconnection between them. The classification is presented in the right hand side of Fig. 4, where the concept of *Organizational Culture* is used to characterize people engaged in the process, *Process Category* from Table 1 is used to characterize the structure of the *work system, Combination of Techniques in Use* is used to characterize tasks completed by people, and *Type of Business Process Support* employed in the process is used to characterize the technology. These concepts are described below.

5.1.1 Organizational Culture

We define organizational culture as a predominant (shared by majority) mental model that guides the behavior of process participants. The concept of organizational culture is widely debated in the management literature [8,9]. Here, we follow the classical work of [9] that identifies seven dimensions in organizational culture. However, we use only three of these dimensions, more exactly, the ones that characterize the existing patterns of behavior, not the efforts of management to create a particular work structure (the latter is covered by the concept of process category). To the dimensions we consider related to the culture as such belong *aggressiveness*, *innovation & risk taking*, and *attention to detail*. While borrowing these dimensions from [9], we both rename them and give them more specific meaning - more suitable for considering alignment of our socio-technical system. The dimensions of organizational culture we use and their values are as follows:

- *World view* (substitutes *aggressiveness*): *competitive/cooperative* – the degree in which process participants consider internal environment as competitive vs. cooperative.
- *Resourcefulness* (substitute *innovation & risk taking*): *low/medium/high* - the degree to which the process participants are able and allowed to find and complete tasks by themselves rather than waiting instructions on what to do and how to do it.

- *Scope* (substitute *attention to detail*): *narrow*, *medium*, *wide* – the context that is taken into consideration by process participants while completing actions in the frame of process instances. *Narrow* corresponds to the immediate surroundings like input and output of a particular activity. *Medium* corresponds to the process instance in the frame of which the current activity is taking place, e.g., the goal to be reached in this process instance. *Wide* corresponds to all process instances under execution, and the objectives of the process as such, e.g., importance of a particular customer even when the customer order is of lesser magnitude.

The three dimensional characterization of organizational culture is aimed to depict the predominant mental model of process participants. It does not exclude the presence of process participants with a different mental model, for example with resourcefulness, or scope that distinguishes itself from the one of the others. Also, assigning particular values to the dimensions of organizational culture does not mean that process participants are not capable to, for example, having a wider scope. Narrow scope may mean that they have adjusted their behavior to the situation in which either they do not have enough information/knowledge to consider the wider scope, or are discouraged to do so.

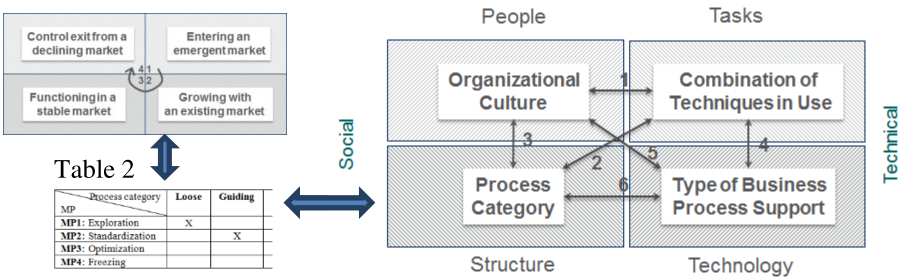


Fig. 4. Characteristics of socio-technical systems, adapted from [6]

5.1.2 Process Category

This concept has already been described in Section 2. In addition to using it as a characteristic of the work system structure, it is used to connect the external and internal environments, as shown in the left hand side of Fig. 4.

5.1.3 Combination of Techniques in Use

We differentiate three techniques used in the frame of business processes:

1. *Follow instructions* – work is completed according to predefined instructions.
2. *Instance based decision making* – decision making based on the information on the development of a particular process instance, e.g., deciding on the course of action in a particular case. We will refer to this type as to *tactical decision making*.
3. *Type-based decision making* – decision making based on the situation in the whole work-system, e.g. prioritizing some process instances from all running instances of the given type when resources are limited. We will refer to this type of decision making as to *strategic decision making*.

A combination of techniques in use is a triple that describes proportions of tasks that corresponds to these three different techniques in the instances of the given process type. For example triple <*high*, *low*, low> characterize a process with high level of well-defined tasks, and very few decision-makings on the tactical and strategic level.

5.1.4 Type of business process support. This concept characterizes systems/tools that assist process participants in running process instances. For our purpose, we characterize such tools from the point of view of how they affect communication/collaboration between process participants in the frame of process instances of the given type. From this perspective, we differentiate three dimensions to characterize the type of business process support (BPS):

1. *Structuredness* of data. This dimension characterizes the degree of structuredness of data (information) introduced by BPS. A low level of structuredness means that the exchange is done informally in a natural language, e.g. in conversations or written document exchange. A high level of structuredness means that the exchange is done through formalized documents, e.g., that include numbers, check lists, or selections from a fixed number of alternatives.
2. *Orderliness* of task flow. This dimension characterizes in what degree the order of the tasks completion is predefined in process instances of the given process type. In a process with high degree of *Orderliness*, the order of tasks execution can be depicted as a flowchart which may contain loops and branches. In a process with a low degree of *Orderliness*, the order is established on the fly, separately for each process instance. The latter does not exclude the existence of some rules that defined partial order in the form of *Obligations* or *Prohibitions*.
3. Information *Logistics*. This dimension referrers to the way information is delivered to process participants that need it for completing their tasks in the frame of a process instance. Roughly, we identify two types of Information *logistics*: (1) *messaging* logistics, when information is sent to a person who needs it, and *shared space* logistics, when all information is placed in a shared space accessible to other participants. The difference between these two schemes is explained in [16], where messaging logistics is referred to as conveyer belt logistics, and shared spaces logistics – as construction site logistics.

Combining these three dimensions we can categorize existing tools that are used. For example:

- Using word processor + emails as BPS = <*Structuredness* = *low*, *Orderliness* = *low*, *Logistics* = *messaging*>
- Using social software, such as a forum, or a wiki, as BPS = <*Structuredness* = *low*, *Orderliness* = *low*, *Logistics* = *shared space*>
- A traditional workflow system with form support = <*Structuredness* = *high*, *Orderliness* = *high*, *Logistics* = *messaging*>
- A case management system = <*Structuredness* = *medium*, *Orderliness* = *low*, *Logistics* = *shared space*>

5.2 Rules of Alignment between Components of Socio-Technical Systems

As follows from Fig. 4, there are six relationships between the four components of socio-technical systems under consideration. The rules of alignments between the components along the six relationships are presented in the form of tables, one for each of the relationships, see Table 3-8. The tables refer to the double edged arrows in Fig. 4. Rows in a table correspond to the categories of one of the component of relationship, while columns correspond to the categories of the other component. A cell states a condition when the column category is aligned to the row category. The meaning of such conditions is explained in the upmost header of the table. If the categories do not need alignment, the cell is given value "-". The tables are constructed via logical analysis of the concepts that corresponds to the rows and columns of each table; these concepts are described in Section 5.1 above. Additional explanations are provided in the footnotes to the tables and the text below.

Table 3. Arrow *1*: Proportion of different kind of tasks in different categories of processes

	Proportion of techniques used		
Process categories	**Follow instructions**[1]	**Tactical decision making**[2]	**Strategic decision making**[2]
Loose	*Low*	*High*	*High*
Guiding	*Medium*	*High*	*Medium*
Restrictive	*High*	*Medium*	*Low*
Strict	*High*	*Low*	*Low*

Footnotes:

[1] The number of rules that describe *when*, *what*, *how* and *by whom* tasks are to be completed grows when going from the loose processes to the strict ones. Thus the proportion of tasks that requires "follow instructions" grows

[2] The more rules that needs to be followed, the less needs for decision making.

Table 4. Arrow *2*: Alignment between Techniques and Organizational culture. The table shows the values of three cultural dimensions required for the three types of techniques.

	Values according to dimensions of organizational culture		
Techniques	**World view**	**Resourcefulness**	**Scope**
Follow instructions	-	*Low*	*Narrow*
Tactical decision making	-	*Medium*	*Medium*
Strategic decision making	-	*High*	*Wide*

Table 5 (arrow 3 in Fig. 4), which describes correspondence between business process categories and organizational culture, has been derived from Table 3 and 4 (arrows 1 and 2 in Fig. 4). This has been done by comparing columns of Table 3 and rows of Table 4, and choosing values of cultural dimensions that correspond to the high proportion of techniques in use. In addition, the need for cooperative behavior for *loose* and *guiding* categories of processes has been taken into consideration (see the footnote in Table 5).

According to Table 5, *loose* and *guiding* processes require the cooperative organizational culture with high enough level of *resourcefulness* and width of *scope*. *Restrictive* and *strict* processes can be conducted in the competitive organizational environment with low-level of *resourcefulness*, and relatively narrow *scope*.

Table 5. *Arrow 3*: Alignment between Process categories and Organizational culture

Process categories	*Values according to dimensions of organizational culture*		
	World view	**Resourcefulness**	**Scope**
Loose	*Cooperative*[1]	*High*	*Wide*
Guiding	*Cooperative*[1]	*Medium*	*Medium*
Restrictive	-	*Medium* or *Low*	*Medium* or *Narrow*
Strict	-	*Low*	*Low*

[1] Loose and guiding processes have high to medium proportion of decision-making. The latter leads to high probability of different decisions being made by different process participants. With lack of cooperation the decision will contradict each-other.

Table 6. Arrow *4*: Alignment between Techniques and BPS. The table shows the values of three BPS dimensions required for the three types of techniques

Techniques	*Values according to dimensions of BPS*		
	Structuredness	**Orderliness**	**Logistics**
Follow instructions	*High*	*High*	*Messaging*
Tactical decision making	*Medium* or *Low*	*Medium*	*Shared space*
Strategic decision making	*Medium* or *Low*	*Low*	*Shared space*

Table 7 (arrow 5 in Fig. 4), which describes correspondence between business process categories and BPS parameters, has been derived from Table 3 and 6 above (arrow 1 and 4 in Fig. 4). This has been done by comparing columns of Table 3 and rows of Table 6, and choosing values of BPS parameters that correspond to the high proportion of techniques in use.

Table 7. Arrow *5*: Alignment between Process categories and BPS

Process Categories	*Values according to dimensions of BPS*		
	Structuredness	**Orderliness**	**Logistics**
Loose	*Low*	*Low*	*Shared space*
Guiding	*Medium*	*Medium or Low*	*Shared space*
Restrictive	*High* or *Medium*	*High* or Medium	*Messaging* or *Shared space*
Stringent	*High*	*High*	*Messaging*

Based on Table 7, the following types of BPS system may suit well different categories of processes:

- *Loose* - social software, e.g. wiki or forums
- *Guiding* - case or adaptive case management system
- *Restrictive* - flexible, e.g. declarative, workflow with form support
- *Strict* - classical workflow with form support

Table 8. *Arrow 6*: Alignment between Organizational culture and BPS.

Dimensions of organizational culture		*Values according to dimensions of BPS*		
		Structuredness	**Orderliness**	**Logistics**
World view	*Competitive*	*High*[1]	*High*[1]	*Messaging*[2]
	Collaborative	-	-	-
Resource-fulness	*Low*	*High*[3]	*High*[3]	-
	Medium	*Medium*	*Medium*	
	High	*Low* or *medium*[4]	*Low* or *Medium*[4]	
Scope	*Narrow*	-	-	*Messaging*[5]
	Medium			*Shared space*[6]
	High			

[1] The competitive organizational culture requires high level of structuredness/standardization to "force" competing personality to contribute to the common goals.
[2] Using shared spaces in competitive culture could be counterproductive, as competitive personalities might be reluctant to sharing extra information between each other.
[3] High structuredness is needed to compensate the low level of resourcefulness
[4] High structuredness might set too many restrictions on a resourceful personality, which may result in him/her using other means outside the BPS system employed, or doing the work formally with low quality, or quitting the job altogether.
[5] Personalities with narrow scope might feel it more comfortable when getting only information related to the task at hands. Too much information in a shared space might be confusing for them.
[6] Using messaging in a culture where personalities are accustomed to getting a wider context of the tasks at hand might lead to extra messages floating around to request and get clarifications on the context. The latter will diminish the productivity. There is also a risk that the culture might change to *scope* becoming *narrow*.

6 Potential Usefulness of the Framework

We see the following *areas of usage* for our heuristic framework:

1. Analysis of past successes and failures of an organization in general, or a particular organizational change completed or tried. Such analysis can be a part of the strategy of becoming a learning organization through reflecting on own practice. The framework covers a wide range of possible failures. A failure could be due to a wrong process category chosen for a given MP, e.g. a restrictive process was used for MP1 – entering a new market. A failure can also be due to the wrong BPS chosen for the given process category, or missing the needs to change the organizational culture when introducing a new process or/and BPS system. For example, introducing a BPS based on shared spaces in the competitive organizational culture

might be unsuccessful, especially if in addition the *scope=narrow* and/or *resourcefulness = low*.

2. Analyzing risks connected to a particular organizational change, especially risks connected to the human factors. For example, there is a risk of *resourceful* people leaving the organization when a *restrictive* or *strict* process is introduced in the culture with high *resourcefulness*. Alternatively, the resourceful people might go around a BPS system that enforces such a process moving the process to the category of *guiding*. Another example is a risk of a BPS with shared spaces not being properly used in the competitive organizational culture, as the competing process participants might avoid uploading all relevant information into the shared space.
3. Devising measures to mitigating the risks involved in a specific organizational change. For example, introducing a *restrictive* or *strict* process in the culture with high *resourcefulness* could be successful if participants of this process are engaged in it only part time, spending the rest of their time working in loose or guiding processes and spending their resourcefulness in the latter. Another example, if a BPS system with shared spaces is introduced in the competitive organizational culture in order to change it to cooperative, it could be helpful for the system having support for messaging as well. This will allow to gradually changing the mode of operation from messaging to using shared spaces along with the progress of changing the organizational culture.

7 Summary and Future Research

The generic situation and problem that all organization face is how to best optimize their resources in any given o possible environment. In this paper, we have presented a generic model in the form of Table 1-8 of a process as a work system that connects possible external environments of this system with its possible internal characteristics. In particular, we have identified four distinct types of the process external environment called Market Positions (MPs) of exploration, standardization, optimization and freezing. Then, we have connected these positions to the categories of business processes and through the latter to organizational culture and characteristics of business process (BPS) systems of structuredness, orderliness, and logistics. In our model, we have connected the three classical dimensions of organizational culture (aggressiveness, innovation & risk taking, and attention to detail) to the properties of BPS systems and the process external environment. To the best of our knowledge, this is a new kind of model, and, as we discussed in the previous section, it could be useful for organizational learning, risk assessment and planning an organizational change that includes changing or introducing a new process and/or BPS system.

Models are by nature a simplification or representation of reality but are of course not reality themselves. According to [17] "all models are wrong bust some of them are useful". To *validate the usefulness of our model*, we plan to use historical data on companies which have changed their market positions, successfully and unsuccessfully, and attempt to extract information on their internal environments at different times. Another source of historical information that could be useful for our purpose is reports on successful and unsuccessful introductions of BPS systems. Having such reports or access to people with experience in such introductions, we can analyze correspondence between the organizational culture, the process and BPS. Another way to validate the usefulness of the model is via using game simulation, though it would require development resources for designing a suitable game. By configuring the game for different processes and BPS, and instructing the players to behave according to the patterns of different organizational cultures, it could be possible to see the results of alignment or misalignment of the socio-technical system from figure 4.

References

[1] Burns, A., Mitchel, W.: Measuring Business Cycles. National Bureau of Economic Research, New York (1946)
[2] Perez, C.: Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages. Elgar, London (2002)
[3] Perrow, C.: A Framework for the Comparative Analysis of Organizations. American Sociological Review, 194–208 (1967)
[4] McDonald, M.: Marketing plans, How to Prepare Them: How to Use Them. Butterworth-Heinemann, Oxford (1989)
[5] Bider, I., Striy, A.: Controlling business process instance flexibility via rules of planning. IJBPIM 3(1), 15–25 (2008)
[6] Bostrom, R.P., Heinen, J.S.: MIS problems and failures: A socio-technical perspective. MIS Quarterly 1(3), 17–32 (1977)
[7] Alter, S.: The Work system method: Connecting people, processes, and IT for business results. Work System Press (2006)
[8] Schein, E.H.: Organizational Culture and Leadership. John Wiley & Sons (2006)
[9] Robbins, S.P., Coulter, M.: Management, 12th edn. Prentice Hall (2014)
[10] Hevner, A.R., March, S.T., Park, J.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
[11] Bider, I., Johannesson, P., Perjons, E.: Design science research as movement between individual and generic situation-problem-solution spaces. In: Organizational Systems. An Interdisciplinary Discourse, pp. 35–61. Springer (2013)
[12] Andersson, T., Bider, I., Svensson, R.: Aligning people to business processes experience report. SPIP 10(4), 403–413 (2005)
[13] Bider, I., Johannesson, P., Perjons, E., Johansson, L.: Design Science in Action: Developing a Framework for Introducing IT Systems into Operational Practice. In: Proceedings of the International Conference on Information Systems, ICIS, Orlando, Florida, US (2012)
[14] Bider, I., Perjons, E.: Evaluating Adequacy of Business Process Modeling Approaches. In: Handbook of Research on Complex Dynamic Process Management: Techniques for Adaptability in Turbulent Environments, pp. 79–102. IGI (2009)
[15] Hilpinen, R. (ed.): Deontic Logic: Introductory and Systematic Readings. Reidel Publishing Company (1971)
[16] Bider, I., Johannesson, P., Schmidt, R.: Experiences of Using Different Communication Styles in Business Process Support Systems with the Shared Spaces Architecture. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 299–313. Springer, Heidelberg (2011)
[17] Box, G.E.P., Draper, N.R.: Empirical Model Building and Response Surfaces. John Wiley & Sons, New York (1987)

Enhancing Feasibility of Human-Driven Processes by Transforming Process Models to Process Checklists

Michaela Baumann, Michael Heinrich Baumann, Stefan Schönig, and Stefan Jablonski

University of Bayreuth, Germany
{michaela.baumann,michael.baumann, stefan.schoenig,stefan.jablonski}@uni-bayreuth.de

Abstract. In traditional approaches business processes are executed on top of IT-based Workflow-Management Systems (WfMS). The key benefits of the application of a WfMS are task coordination, step-by-step guidance through process execution and traceability supporting compliance issues. However, when dealing with human-driven workflows, conventional WfMS turn out to be too restrictive. Especially, the only way to handle exceptions is to bypass the system. If users are forced to bypass WfMS frequently, the system is more a liability than an asset. In order to diminish the dependency from IT-based process management systems, we propose an alternative way of supporting workflow execution that is especially suitable for human-driven processes. We introduce the so-called process checklist representation of process models where processes are described as a paper-based step-by-step instruction handbook.

Keywords: process modelling, process checklists, paper-based process execution.

1 Introduction

Since approximately 20 years process management is regarded as an innovative technology both for the description of complex applications and for supporting their execution [1]. In traditional approaches business processes are executed on top of IT-based Workflow-Management Systems (WfMS) [2]. The key benefits of the application of a WfMS are task coordination, step-by-step guidance through process execution and traceability supporting compliance issues [3]. However, when dealing with human-driven workflows that heavily depend on dynamic human decisions, conventional WfMS turn out to be too restrictive [4]. Especially, the only way to handle exceptions – which regularly occur in human-driven workflows – is to bypass the system. If users are forced to bypass WfMS frequently, the system is more a liability than an asset [4]. In total, users start to complain that “the computer won’t let them” to do the things they like to accomplish [5]. So users like to get more independent from “electronic systems” in order to

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 124–138, 2014.

become more flexible. If original documents are needed for executing a process, in many cases a paper-based execution model is preferred [6].

Furthermore, the introduction of a WfMS is regarded as a huge, cost-intensive project [7]. Many organizations cannot afford to introduce such a system therefore. However, they desire to manage their processes since they regard them as valuable and effective. In order to diminish the dependency from IT-based process management systems, we propose an alternative way of supporting workflow execution that is especially suitable for human-driven processes, like it is the case for example in public administration and authorities. We introduce the so-called process checklist representation of process models. Here, processes are described as a paper-based step-by-step instruction handbook. The process checklist is handed over during process execution from process participant to process participant.

Successful task accomplishments are recorded through signatures of corresponding agents. In principle the most important statement is that at the end of the process all signatures are on the checklist. So it is completely output oriented. Nevertheless, the checklist method describes a valuable form of process usage and widens its spectrum towards non-computer based and extremely flexible process execution. Besides, the process checklist also supports the key benefits of traditional WfMS. The checklist is handed over to responsible agents (task coordination), process tasks are serialized and marked by a unique identifier (step-by-step guidance) and the checklist itself as well as the corresponding signatures ensure traceable process execution. The work at hand provides the general structure of process checklists as well as an elaborate transformation algorithm of basic BPMN process model elements [8] to process checklists. Fig. 1 shows a comparison of traditional IT-based process execution and the paper-based approach provided by the work at hand.

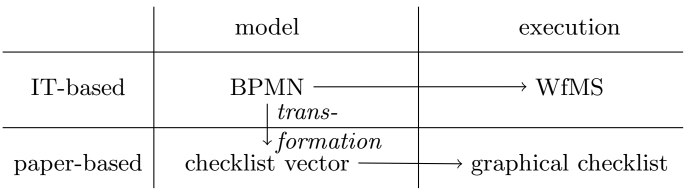


Fig. 1. Schematic approach of distancing from IT-based process management systems

2 Background and Related Work

A checklist is a list of items required, things to be done, or points to be considered, used as a reminder [9]. Checklists are generally seen as a suitable means for error management and performance improvement in highly complex scenarios like clinical workflows [10]. Therefore, we propose to define a generic method for

transforming general process models to the checklist representation. The problem of transforming a model drawn in one business process modeling notation into another notation has been examined in different papers, e.g., [11], [12]. However, to the best of the authors' knowledge, the transformation of process models to a checklist representation has not been discussed so far.
Before specifying the transformation of process models into checklists, we have to determine how suitable process models should look like and what elements a checklist consists of. These specifications are necessary to give concrete mapping rules. For process models, only basic elements of the Business Process Modeling Notation are allowed, as [13] shows this is enough in most cases and as the paper at hand has to be seen as a first approach to this topic.

Definition 1 (Process model). *A process model is defined according to BPMN 2.0 (see e.g. in [8]) allowing for the following basic elements:*

- *flow objects: activities, events (start, end), gateways (AND, XOR, OR)*
- *sequence flows*
- *data (input/output) objects*
- *participants: one pool, possibly separated into different lanes*

As we consider the application of checklists appropriate only within one company, there should not occur processes with more than one pool. Therefore, we do not have to take message flows into account. Which forms of activities, events and gateways can be covered with our transformation rules will become apparent when it comes to the concrete transformation of process models into checklists. We specify a checklist as follows.

Definition 2 (Checklist vector). *A checklist is a vector* $\mathcal{C} = (p_1^t, p_2^t, \ldots, p_n^t)$, $n \in \mathbb{N}$, $t \in \{o, c\}$ *with two different kinds of components:*

$$p_i^o = (ID_i, AC_i, OD_i, AG_i)$$

with ID_i, AC_i, OD_i, AG_i *being strings and*

$$p_j^c = (AN_j, CO_j, GT_j, AG_j)$$

with AN_j, CO_j, AG_j *being strings and* GT_j *being a vector of the form*

$$GT_j = (s_j, a_{j,1}, g_{j,1}, a_{j,2}, g_{j,2}, \ldots, a_{j,k}, g_{j,k})$$

with $k \in \mathbb{N}$, *strings* $a_{j,l}$, *integers (or NULL)* $g_{j,l} \in \{1, \ldots, n\} \cup \{NULL\}$, $l = 1, \ldots, k$, *and* $s_j \in \{0, 1\}$.

This definition uses a lot of different variables that need some explanation: The first component of a checklist vector, p^o, is called operating point. It contains information about incoming data objects (ID), the activity (AC) which may be an activity in the literal sense of BPMN or an event, outgoing data objects (OD), and the performing agent (AG). An operating point gives more or less conrete instructions to the respective agent about what he has to do. The other component of a checklist vector, p^c, is called control point. In general, a control point is a transformed gateway, therefore it contains information about the condition (CO) which may also be empty if it corresponds to a parallel gateway, and the responsible agent (AG as in p^o). AN is a component kept free for special annotations (we will see examples later) and GT is a vector with one boolean component s and k pairs of string (a) and integer (g) components. g refers to other components of $\mathcal{C}$ and is therefore element of $\{1, \ldots, n\}$ or $NULL$. With this formal definition of a checklist, the checklist vector, it is already possible to give concrete mapping instructions as listed in the next section. Before we turn towards this subject, we want to give the reader a visual impression of how the two components p^o and p^c may be illustrated on a graphical checklist in Fig. 2 and Fig. 3. The components of vector GT_j are shown in Fig. 4.

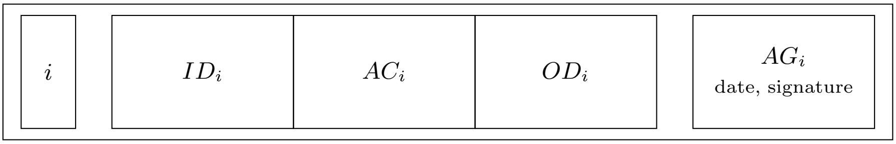


Fig. 2. Visualization of p_i^o which means the i-th component of $\mathcal{C}$ is an operating point

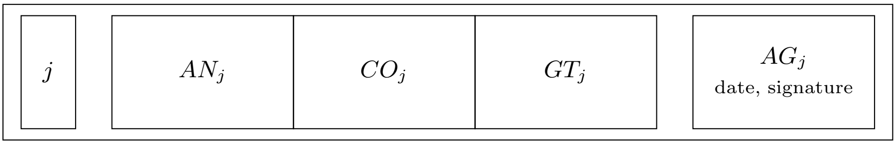


Fig. 3. Visualization of p_j^c which means the j-th component of $\mathcal{C}$ is a control point

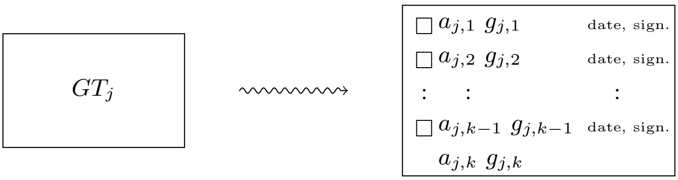


Fig. 4. Visualization of GT_j. Entries date and signature in the third column only appear, if $s_j = 1$. The k-th row never has a square in the first column nor a date and signature. In fact, the k-th row may be empty, i.e., $a_{j,k} =$ "" and $g_{j,k} = NULL$.

In which way these checklist components are filled with information given by the process model and how the resulting operating and control points are represented in the graphical checklist is explained in the next two sections.

3 Transformation of Process Model Elements

This section focuses on generating a checklist, that means it is explained, in which way the single elements of the (BPMN) process model are transferred into either operating points or control points. These steps are basically performed in a simply algorithmic way, except for parallel gateways.

3.1 Transformation of Activities

Activities are transformed straight into operating points p^o. Their description is mapped on the field AC whereas all directly incoming data and directly outgoing data is mapped on the field ID and OD respectively. The participant of the corresponding lane or hierarchy of lanes, that may, e.g., be a single person is mapped onto the field AG. An example of an activity with documents and participant is given in Fig. 5.

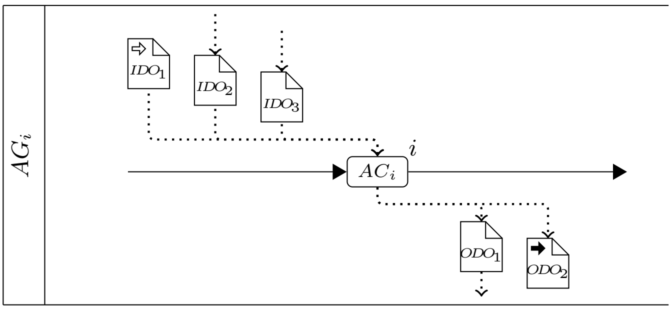


Fig. 5. Exemplary excerpt from a process model with labels according to an operating point p_i^o. $ID_i = IDO_1.IDO_2.IDO_3$ and $OD_i = ODO_1.ODO_2$.

3.2 Transformation of Subprocesses

Occurring subprocesses, marked with a symbol as seen in Fig. 6, may be taken into a checklist in different ways:

1. Include the complete subprocess (comparatively long, but correct checklist)
2. Generate a new checklist for each subprocess (insertion of two operating points into the original checklist ist necessary: one with work instructions for printing/passing on the new checklist, one with work instructions for waiting for the finished subprocess, subprocess checklist as incoming data)

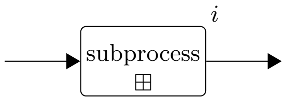


Fig. 6. Symbol for a subprocess in BPMN 2.0

3.3 Transformation of Gateways

Transformation of Exclusive Gateways. An exclusive split gateway (Fig. 7) has to be transformed into a control point in which the decision question and the possible answers with the respective "go to"-numbers ($g_{j,1}, g_{j,2}$ and $g_{j,3}$ in Fig. 7) are mentioned. If there is a exclusive join gateway (Fig. 8) too, at the end of each branch of the respective splitting gateway a jump instruction to the next point in the checklist after the join gateway ($g_{j,4}$ in Fig. 8) must be inserted, except the next point following a branch is the point following the join gateway. The execution of an exclusive gateway may cause problems if at least one "go to"-number is in the past, but this problem will be solved in the next section.

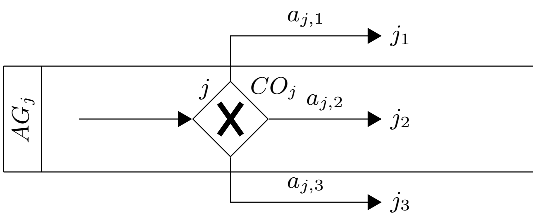


Fig. 7. Exclusive split gateway with question CO_j and possible answers $a_{j,1}, a_{j,2}, a_{j,3}$. p_j^c: AN_j may be used for data. $g_{j,k} = j_k$, $k = 1, 2, 3$, $a_{j,4} =$ "" and $g_{j,4} = NULL$.

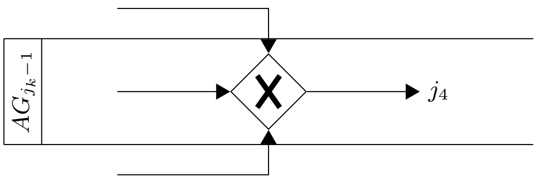


Fig. 8. Exclusive join gateway that does not have to exist if the outgoing branches of the exclusive split gateway end with terminal events. $p_{j_k-1}^c$: $AN_{j_k-1} =$"", $CO_{j_k-1} =$ "*XOR end*", $s_{j_k-1} = 0$, $a_{j_k-1} =$ "*goto :*", $g_{j_k-1} = j_4$, $k = 2, 3$.

Transformation of Parallel Gateways. There are several ways of transforming parallel gateways into a checklist whereby all of them have different advantages and disadvantages. Some of these possibilities are listed below. Note, that a mixture of these transformation possibilities is also conceivable.

Static Sequential Transformation This type of transforming a parallel gateway takes the several branches of the process model, that are between the split and join gateway (Fig. 9), and brings them into an arbitrary order. The gateway itself is not mapped to the checklist.

Dynamic Sequential or Postbox Transformation A parallel split will be transformed to a control point p_j^c. The parallel branches in the process model have to be written down in a sequential way in the checklist. At the end of each branch a jump to p_j^c, realized with a simple control point, is necessary and in p_j^c the number of the point following the respective parallel join has to be noted. There are different ways of executing this parallel split, and some of them correspond to another transformation, but this will be dealt with in the next section.

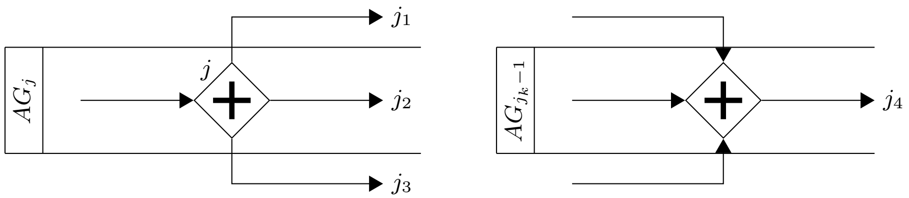


Fig. 9. Parallel split gateway p_j^c: AN_j for annotation, e.g. *DO*s, CO_j = "*AND*", $s_j = 1$, $a_{j,1}, \ldots, a_{j,3}$ ="", $g_{j,1} = j_1$, $g_{j,2} = j_2$, $g_{j,3} = j_3$, $a_{j,4}$ ="Finally go to", $g_{j,4} = j_4$, $k = 2, 3, 4$. Parallel join gateway $p_{j_k-1}^c$: AN_{j_k-1} ="", CO_{j_k-1} ="AND end", $s_{j_k-1} = 0$, a_{j_k-1} ="go to", $g_{j_k-1} = j$, $k = 2, 3, 4$.

Parallel Transformation For each parallel branch a checklist is generated and distributed by the agent of the split gateway (see Fig. 9) to the agents of the first process element of the branches. It is modelled as one control node p_j^c. If the gateway splits into k branches, then $a_{j,k+1}$ ="Finally go to" and $g_{j,k+1} = j+1$. If the name of the current checklist is "Checklist", then CO_j ="AND – print checklists "Checklist_sub1",. . . ,"Checklist_sub$k$", if the names of the sub-checklists are "Checklist_sub1",. . . ,"Checklist_subk". Of course $a_{j,1}, \ldots, a_{j,k}$ have to reference these sub-checklists, $g_{j,1}, \ldots, g_{j,k} = NULL$ and $s_j = 1$.

Transformation of Inclusive Gateways. The transformation of inclusive gateways can be done similar to the transformation of parallel gateways. More precisely, there are the possibilities to use the dynamic sequential or postbox transformation or the parallel transformation. The only difference is, that in p_j^c we have CO_j and $a_{j,1}, \ldots, a_{j,k}$ like in the exclusive gateway transformation, i.e., the condition/question and the answers have to be taken over from the process model.

3.4 Transformation of Events

Direct Transformation of Events. Some events, like signal events, can be transformed like activities, that means to p_i^o, with $AC_i =$ "" or AC_i is used for transmitting some message.

Indirect Transformation of Events. Most events, like time, condition and message events, are requirements for the next point in the checklist and can be modelled this way. This requirement is written down in AC or AN of the following operating or control point.

Ignored Events. Other events, like the start event, can be ignored, that means they have no respresentation in the checklist, because they won't influence the execution.

4 Enactment of the Graphical Checklist

A graphical checklist contains a cover sheet with name of the checklist (name of the process), timestamp, and a list for writing down the current checklist and the current point, i.e., the next point to be worked on. Furthermore, a graphical checklist consists of at least one checklist as described above (resulting from a checklist vector) with a consecutive number, starting with 1, a receipt book and a list for data objects and maybe data objects (documents). An illustrating example for these components is given in Fig. 10.

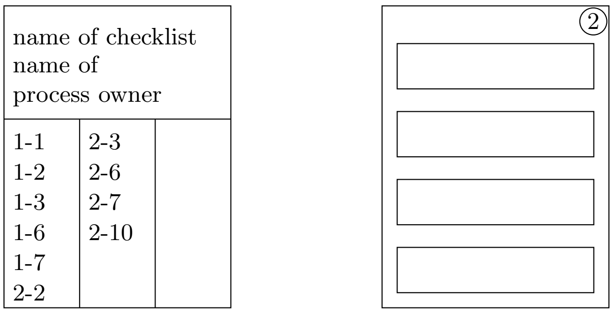


Fig. 10. Cover sheet (left-hand side) with name of the checklist/process, name of process owner and list of the next points to be executed; checklist (right-hand side) with current number in the upper right corner and operating/control points. Obviously, in checklist no. 1 a gateway caused a jump into the past (from point no. 7 to point no. 2)

When starting a process with checklists, the "process owner", i.e., that person starting the execution of the process, has to print the checklist with cover sheet and data object list. Then he assigns the checklist its current number 1.

Input data, that means input documents, have to be added and scheduled in the respective list. On the cover sheet "1–1" is noted, that means, the current status of execution is "checklist 1" and "point 1". In addition, he has to write his name on the cover sheet so that the checklist can be handed over to him after finishing the process. This graphical checklist has to be passed to the agent named in point 1, who has to check for completeness, that means especially if all listet documents are handed over, and quit the delivery. The process owner has to archive the signed receipt for later reconstruction if necessary.

Every time an agent gets the graphical checklist he has to run through this acknowledgement process (checking the documents for completeness, sign a receipt) and then check for the current point of the checklist on the cover sheet. When the last entry is 1–23 he has to look at point 23 of the current checklist, that has number 1, and execute this point, if all necessary documents are available and possible conditions are fulfilled. Of course, the agent named in this point should be correct (otherwise the checklist has not been handed over properly). After execution of the current point he has look which agent is next. If it is himself he executes the next point and writes it down on the cover sheet, else he updates the document list, writes the next point on the cover sheet, hands the checklist over and archives the received receipt. If one agent sends a document directly to another, this document has to be deleted from the data object list and maybe listed again later on by the other agent.

4.1 Execution of Operating Points

Operating points are executed straightaway as described above, performing the task (with possible constraint resulting from a transformed event) as given in AC. If documents are produced, they should correspond to that ones listed in the outgoing documents OD. After performing the task, he signs the operating point for making clear, he has finished this point.

4.2 Execution of Control Points

Execution of Exclusive Gateways. If a control point resulting from an exclusive gateway has to be processed, the agent has to check for the condition or question in field CO. He marks his answer in GT in the box $\square$ in front of the corresponding answer $a_{\cdot,l}$. If there are any documents helping him to decide, they are listed in AN. After marking he gets the number of the next point, $g_{\cdot,l}$. Two possible sceneries may occur: $g_{\cdot,l}$ is greater than the current point number, then everything can go on as before. If $g_{\cdot,l}$ is smaller than the current point number, then there is a problem, as that point with number $g_{\cdot,l}$ may have been processed already in the past and therefore is signed already. If such a return occurs, than the agent of the control point has to print a new checklist (just the checklist itself) and assign it the number $i+1$ if the number of the current checklist was i. On the cover sheet, he writes for the next point to be executed $(i+1)$–$(g_{\cdot,l})$. After doing this, he signs in field AG and passes the new checklist (together with the old one for reconstruction opportunity) to the agent of point

$g_{\cdot,l}$. This agent has to recognize that the consecutive number of the checklist has changed which is obvious on the cover sheet.

Execution of Parallel Gateways.

Static Sequential Transformation If a parallel gateway was transformed in the static sequential way, then it does not appear in the checklist, that means the performing agents do not know, that there has been a gateway in the BPMN process model. All branches are executed in the specific order as chosen by the person who transformed the process model.

Dynamic Sequential Transformation When coming to a control point being the transformation of a parallel gateway with the dynamic sequential method the agent of that point can decide about the execution order of the different branches during the processing of the checklist. He can take into account the current circumstances like availability of the agents in the different branches, or anything else. When he chooses one branch, he marks his decision in the corresponding box □, notes it on the cover sheet and passes the graphical checklist over to the agent of the respective point, on the right-hand side of the marked box. The branch is processed and at the end there is a control point that refers back to the control point where the decision of the branch was made. So, the agent gets the checklist back (with checking for all documents and quitting again) and signs the chosen branch in GT (that one with the marked box, that has not been signed yet). Then he chooses the next branch to be processed the same way as before. If all branches have been marked and signed, then he signs the whole control point in field AG and passes the checklist over to the agent of that point listed after "finally go to" in GT. The whole procedure can be reconstructed with the notes on the cover sheet.

Postbox Method If parallel gateways are performed with the postbox method, the checklist itself looks the same as transformed according to the dynamic sequential way. The difference is in the execution, as the postbox method allows for parallel processing of the different branches. When the performance of a checklist reaches an AND control node the checklist is posted like an announcement in one place together with all documents (that can be stored in postbox) and all agents can look for the next points that have to be executed on the cover sheet, where all first points of the different branches have to be noted in a parallel way. With this method, the documents do not have to be handed over from one point to another. After finishing all branches, the agent of the control node that started the postbox method collects the checklist and all documents now being in the postbox, checks for completeness, signs in AG if everything is okay and goes on as before. This method may become confusing and needs initiative of all agents. But it considers the parallel aspect of parallel gateways.

Parallel Transformation With this method it is also possible to consider simultaneity of the different branches. The agent of the control node prints all required sub-checklists, marks the boxes □ in GT if handed over together with needed documents to the respective agents of the first points in the branches, as listed in GT, and signs every returning sub-checklist in GT. If all sub-checklists have returned, he signs in AG and the execution of the control node is finished. As one can imagine, this method is more elaborate, as multiple checklists have to be generated, but it provides a good overview over the process in contrast to the postbox method. We recommend this method if the branches are relatively long, so that the effort of generating more than one checklist is somehow justified.

The mentioned transformation and execution versions are somehow suggestions, clearly many other versions are imaginable and of course different versions can be mixed.

Execution of Inclusive Gateways. Like the transformation of inclusive gateways, the execution of inclusive gateways can again be seen as a mixture of exclusive and parallel gateways. The agent of the corresponding control point has to choose his anwers (mark the boxes □), in contrast to exclusive gateways possibly more than one, and then for the chosen ones he can proceed like with parallel gateways (except for the static sequential method, as this was no possible transformation for inclusive gateways).

All methods mentioned so far require a well-modelled process model, that means for example, that there are no returns out of AND branches, no document is needed in parallel branches without having a copy of it, or that no document is archived if there is the possibility of a return into the past where this document will be needed again. Changes of the underlying process model involve modifications of the checklist for all future process instances. If problems or questions during the execution of one checklist occur, one should confer with the corresponding process owner.

5 Transformation Example

As an illustrating example the process model given in Fig. 11 is transformed into a checklist vector as seen in Table 12.

For transformation of parallel and inclusive gateways the dynamic sequential method was chosen. That is why the documents “exposé” and “course materials” do not have to be copied for the different branches of the corresponding gateway, as they are performed in a sequential order and the document is handed over together with the checklist. For a graphical checklist, one has to put the table view of the checklist into the visually more appealing form as given in Figures 2, 3 and 4, as it is easier to read. Furthermore, the cover sheet has to be added as well as the receipt book and the document list.

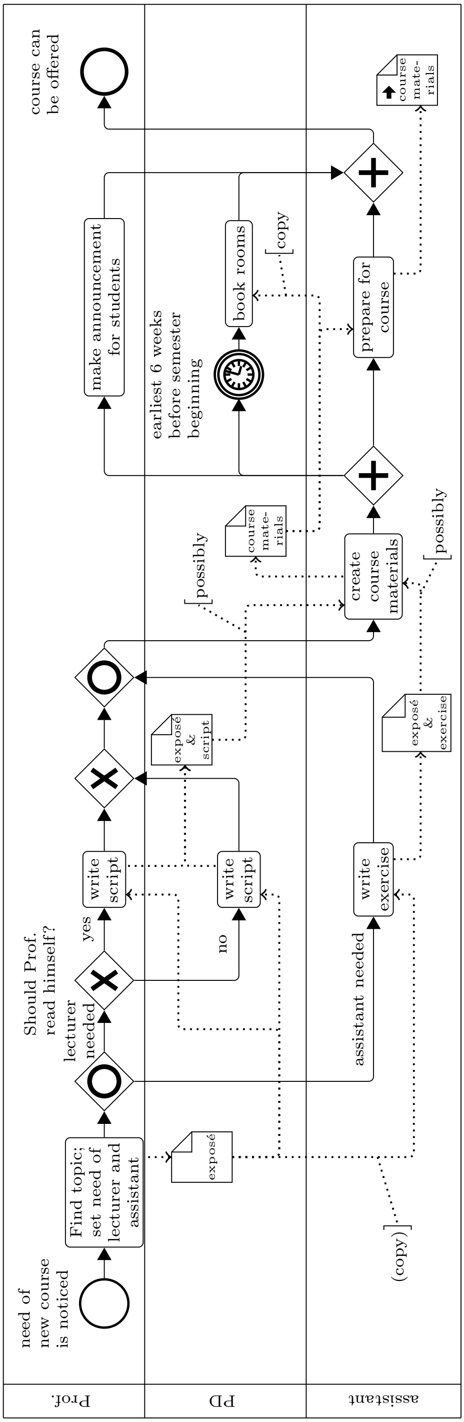


Fig. 11. Exemplary process model: conception of a new course

Table 12. Table view of a checklist vector representating the process model of Fig. 11

No.	type	*ID/AN*	*AC/CO*	*OD/GT*	*AG*
1	*o*		Find topic; set need of lecturer and assistant	exposé	Prof.
2	*c*		OR	$s_2 = 1$ $a_{2,1}$ =lecturer needed $g_{2,1} = 3$ $a_{2,2}$ =assistant needed $g_{2,2} = 8$ $a_{2,3}$ =finally go to $g_{2,3} = 10$	Prof.
3	*c*		XOR Should Prof. read himself?	$s_3 = 0$ $a_{3,1}$ =yes $g_{3,1} = 4$ $a_{3,2}$ =no $g_{3,2} = 6$ $a_{3,3}$ =“” $g_{3,3} = NULL$	Prof.
4	*o*	exposé	write script	exposé, script	Prof.
5	*c*		XOR end	$s_5 = 0$ $a_{5,1}$ =go to $g_{5,1} = 7$	Prof.
6	*o*	exposé	write script	exposé, script	PD
7	*c*		OR end	$s_7 = 0$ $a_{7,1}$ =go to $g_{7,1} = 2$	Prof.
8	*o*	exposé	write exercise	exposé, exercise	assistant
9	*c*		OR end	$s_9 = 0$ $a_{9,1}$ =go to $g_{9,1} = 2$	Prof.
10	*o*	exposé, script OR exercise	create course materials	course materials	assistant
11	*c*		AND	$s_{11} = 1$ $a_{11,1}$ =go to $g_{11,1} = 12$ $a_{11,2}$ =go to $g_{11,2} = 14$ $a_{11,3}$ =go to $g_{11,3} = 16$ $a_{11,4}$ =finally go to $g_{11,4} = 18$	assistant
12	*o*		make announcement for students		Prof.
13	*c*		AND end	$s_{13} = 0$ $a_{13,1}$ =go to $g_{13,1} = 11$	Prof.
14	*o*	course materials	earliest 6 weeks before semester beginning: book rooms		PD
15	*c*		AND end	$s_{15} = 0$ $a_{15,1}$ =go to $g_{15,1} = 11$	PD
16	*o*	course materials	prepare for course	course materials	assistant
17	*c*		AND end	$s_{17} = 0$ $a_{17,1}$ =go to $g_{17,1} = 11$	assistant
18	*o*		course can be offered		Prof.

6 Conclusion, Limitations and Future Work

In order to diminish the dependency from IT-based process management systems, the work at hand proposed an alternative way of supporting workflow execution that is suitable for human-driven processes. We introduced the process checklist representation of process models where processes are described as a paper-based step-by-step instruction handbook. The process checklist is handed over during process execution from process participant to process participant. Successful task accomplishments are recorded through signatures of corresponding process participants.
This way, the process checklist also supports the key benefits of traditional WfMS. The checklist is handed over to responsible agents (task coordination), process tasks are serialized and marked by a unique identifier (step-by-step guidance) and the checklist itself as well as the corresponding signatures ensure traceable process execution. The work at hand provides the general structure of process checklists as well as a transformation algorithm of basic BPMN process model elements to process checklists.
In contrast to the advantages over IT-based process management systems as mentioned before, paper-based checklists can also have disadvantages compared to traditional systems. Checklists represent a single point of access, so support for distributed agents may be difficult. If this is the case, one has to ask if using a paper-based checklist is the right thing for this specific application, as we recommend using checklists for example in administrational environments.
In general, it is possible to transform a procedural process model to a process checklist based on the proposed algorithm. However, due to the serialization of the process, the checklist representation has of course problems when dealing with flexibility and parallelism. Here, process modellers have to choose a suitable transformation method as described in section 4. For future work we will evaluate the proposed approach within a real life business case. Here, we expect useful experiences regarding the acceptance and cooperation of participating agents. Based on these results we will improve methodology, design and representation. Furthermore, we will focus the transformation of loosely-specified processes like declarative process models, e.g., Declare [14].

Acknowledgement. The presented work is developed and used in the project "Kompetenzzentrum für praktisches Prozess- und Qualitätsmanagement", which is funded by "Europäischer Fonds für regionale Entwicklung (EFRE)".

The work of Michael Heinrich Baumann is supported by Hanns-Seidel-Stiftung e.V.

References

1. Jablonski, S.: Do We Really Know How to Support Processes? Considerations and Reconstruction. In: Engels, G., Lewerentz, C., Schäfer, W., Schürr, A., Westfechtel, B. (eds.) Nagl Festschrift. LNCS, vol. 5765, pp. 393–410. Springer, Heidelberg (2010)

2. Zairi, M.: Business Process Management: a Boundaryless Approach to Modern Competitiveness. Business Process Management Journal 3, 64–80 (1997)
3. Reichert, M., Weber, B.: Enabling Flexibility in Process-aware Information Systems. Springer, Heidelberg (2012)
4. Van der Aalst, W., Weske, M., Grünbauer, D.: Case handling: A new paradigm for business process support. Data & Knowledge Engineering 53(2), 129–162 (2005)
5. Condon, C.: The Computer Won't Let Me: Cooperation, Conflict and the Ownership of Information. In: CSCW, pp. 171-185 (1993)
6. Luff, P., Heath, C., Greatbatch, D.: Tasks-in-interaction: paper and screen based documentation in collaborative activity. In: CSCW (1992)
7. Melenovsky, M.: Business process managements success hinges on business-led initiatives, Gartner Research, Stamford, CT, pp. 1-6 (July 2005)
8. Object Management Group Inc.: Business Process Model and Notation (BPMN) Version 2.0 (2011), `http://www.omg.org/spec/BPMN/2.0`
9. Wolff, A., Taylor, S., McCabe, J.: Using checklists and reminders in clinical pathways to improve hospital inpatient care. MJA 2004 181, 428–431 (2004)
10. Hales, B.M., Pronovost, P.J.: The checklist – A tool for error management and performance improvement. Journal of Critical Care 21(3), 213–235 (2006)
11. Hauser, R., Friess, M., Küster, J.M., Vanhatalo, J.: Combining Analysis of Unstructured Workflows with Transformation of Structured Workflows. In: Proc. 10th IEEE International Enterprise Distributed Object Computing Conference, EDOC (2006)
12. Koehler, J., Hauser, R., Küster, J., Ryndina, K., Vanhatalo, J., Wahler, M.: The Role of Visual Modeling and Model Transformation in Business-driven Development. In: Proc. 5th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2006), Vienna, Austria (2006)
13. 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)
14. Pešić, M.: Constraint-Based Workflow Management Systems, Shifting Control to Users (2006)

Fabric-Process Patterns

Towards a Methodology for Fabric-Process Design

Florian Bär[1], Rainer Schmidt[1], and Michael Möhring[2]

[1] Faculty of Computer Science and Mathematics, Munich University of Applied Sciences, Lothstr. 64, 80335 Munich, Germany
{Florian.Baer,Rainer.Schmidt}@hm.edu

[2] Department of Business Information Systems, University of Applied Sciences Aalen, Beethovenstraße 1, 73430 Aalen, Germany
Michael.Moehring@htw-aalen.de

Abstract. Fabric-processes are processes acting upon a fabric. A fabric is a set of virtualized resources and services supporting business processes. Fabric processes do not directly pursue a business goal, but IT-related goals such as the creation of a cloud-service or its configuration. Fabric-processes differ from business processes in their meta-model. Fabric processes include resources and operations not found in business processes. The meta-model developed enables the proper specification of fabric-processes. We address requirements for specifying fabric-processes by so-called Fabric-Process Patterns. We develop a Fabric-Resource-Meta-Model which implements the Fabric-Process Patterns. The Fabric-Resource-Meta-Model extends existing resource meta-models.

Keywords: Business Process Management, Workflow Management, Resource Perspective, Operational Perspective, Workflow Patterns.

1 Introduction

Many IT departments are under high pressure from upper management, because they are primarily regarded as a collection of risks and cost drivers [1] for business process support. Therefore many CIOs focus on new approaches to reduce cost and reduce risks. One approach is to regard data centers as IT-Service providers [2], providing more and more complex cloud-services. Cloud-services are provided using virtual resources, such as computing, storage and networking, integrated and managed by a so-called fabric. A virtual resource is a resource providing the same effects as a real one, without being bound to a physical entity as the real one. Virtual computing resources comprise virtual machines that are defined by a processor and memory. Storage resources allow to abstract the storage location from physical devices, e.g. by providing virtual hard-disks. Virtual networking resources encompass allow the creation of networks completely in software. A fabric is a set of virtualized resources and services supporting business processes.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 139–153, 2014.

Using a fabric with virtualized resources provides a number of advantages for business process support compared to using physical resources. First, a fabric increases the agility of business process support. New services can be provided quickly, with nearly no delay or upfront costs. Second, the virtualization of resources allows the pooling of resourses and thus a much higher scalability in the case of unexpected peaks in demand. Third the availability is increased by the mobility of virtual resources. Fourth, virtual resources can be easily connected to clusters and other configurations increasing the resiliency of business process support.

Fabric-processes are a semi-ordered set of activities that are executed in a coordinated and collaborative manner on a set of virtual resources and meta-services. Contrary to business processes [3] they do not pursue ostensibly a business goal, but an IT-related goal such as the creation of a cloud-service or its configuration. Because of the fact that fabric-processes are mainly executed on a set of virtual resources instead of human resources, they offer rich potential for automation. As a part of IT-Business-Alignment [4], the various business and IT goals are harmonized.

The automation of fabric-processes (Fabric-Process Automation, FPA) provides a number of important benefits. First, by reducing human intervention, automating fabric-processes is an important means for cost reduction in IT. Second, automated processes are much more reliable than manual ones. By this, the quality of cloud-service provisioning is increased. Furthermore, automated fabric-processes can be executed much faster than manual ones. By this, cloud-services can be provided and configured in a fast way. The automation of fabric-processes is crucial for self-service approaches in cloud-computing. In this way, new cloud-services can be deployed quickly, because automated fabric-processes configure the necessary resources.

Fabric-processes are executed on a set of virtual resources instead of human resources. By this, fabric-processes differ from business processes both in their resource meta-model and their operational meta-model. So, special requirements have to be captured by these meta-models in order to enable a proper specification of fabric-processes. There has been a series of academic work in the field of Business Process Management (BPM) so far [5], [6], [7]. Several patterns as well as perspectives have been introduced for specifying business processes and workflows [8], [9], [10]. Also, there have been introduced a number of resource and organizational meta-models for the specification of business processes and workflows from the resource perspective [11], [12], [13]. However, the patterns and resource meta-models that were introduced so far, are mainly focused on processes or workflows that are mainly executed on human resources instead of virtual resources or other non-human resources. By this, processes that are mainly executed on non-human resources like e. g. fabric-processes, cannot be specified in a fully integrated manner by using these existing patterns and resource meta-models. Hence, the existing patterns and resource meta-models have to be extended in a certain way.

In this paper, we systematically address requirements for specifying fabric-processes. To do so, we reviewed the conceptual foundations of 16 existing Business Process Management Systems (BPMS) [3] and Workflow Management Systems (WFMS) [14], [15] such as Camunda [16], Bonita BPM [17], YAWL [18] and Activiti [19] as well as three IT-Process-Automation (ITPA) tools including Microsoft

Orchestrator [20], IBM SmartCloud [21] Orchestrator and HP Operations Orchestrator [22]. We especially examined the number of capabilities provided by these tools in order to interact with virtual resources. Based on these already provided possibilities, we derived a number of fundamental requirements in order to specify fabric-processes. We also, in corporation with the software company Facility Network Technology (FNT) [23], interviewed multiple industry experts about specifying and automating fabric-processes. We interviewed the IT industry experts for one hour in a face to face situation. Based on these interviews we were able to identify certain requirements to specify fabric-processes that are not provided by the existing BPMS and ITPA tools. So, we extended the requirements derived from the provided possibilities of existing BPMS and ITPA tools with the requirements identified by interviewing the industry experts. We indicate the overall set of identified requirements through Fabric-Process Patterns. The Fabric-Process Patterns can be used in order to compare the available BPMS and ITPA tools with respect to their capabilities for specifying fabric-processes. The patterns also can be used to develop a resource meta-model extension that can be implemented in an existing resource meta-model provided by an existing BPMS or ITPA tool. By doing this, the appropriate tool can be used in order to specify fabric-processes as well as the underlying virtual resources in a rich and fully integrated manner. In this paper, we give an example for such a resource meta-model extension, by developing a so-called Fabric-Resource-Meta-Model (FRMM), implementing all the Fabric-Process Patterns.

The remainder of this paper is organized as follows. We first discuss some related work. In chapter three we differentiate fabric-processes from normal business processes by examining the special requirements of virtual resources. In chapter four the Fabric-Process Patterns are described. Afterwards, we develop the FRMM that can be implemented in an existing BPMDS or ITPA tool in order to use this tool for specifying fabric-processes. Section six concludes this paper.

2 Related Work

A series of shortcomings regarding the modelling and description of organizational and policy issues in BPMS and WFMS, were found by Bussler and Jablonski in early work [24]. In subsequent work [10], they also introduced various perspectives to specify (business) processes and workflows in a broad sense and in a more integrated manner. In [25] a meta-model was developed, which describes the dependencies of teams working on a concrete process or workflow instance. Further meta-models describing relationships between different process and workflow concepts, were introduced in [26], [27], [28]. An abstract resource model with focus on the efficient management of resources in a process or workflow context was presented in [29]. Another abstract resource model focusing on the various resource classes and their interactions was introduced in [30]. However, these meta-models and resource models mostly focus on human resources instead of virtual resources or other non-human resources.

In 2005 Russel et al. introduced 43 Workflow Resource Patterns [31], [32]. The patterns were identified by analyzing various BPMS, WFMS and standards for their capabilities in dealing with human resources and work allocation. Our research thus complements these Workflow Resource Patterns, by introducing several specialized resource patterns, which are suitable for virtual resources. Russel et al. also introduced multiple Workflow Data Patterns [33] and Workflow Exception Patterns [34]. These patterns can be seen as an extension to the 43 Workflow Control-Flow Patterns introduced in early work [9], [35]. There is also research on the Business Process Model and Notation (BPMN) [36] focused on its suitability for modelling resources associated with a business process [37]. Further research in this context focused on evaluating BPMN against the Workflow Resource Patterns [38], [39], [40]. An overview of workflow management methodologies and software products is provided by Georgakopoulos et al. in [41]. They also discuss infrastructure technologies that can support workflow automation in complex real-world environments. However, they did not address specific operational requirements for automating these workflows.

Fabrics are a part of dynamic cloud-environments. Dynamic cloud-environments provide a series of cloud-services for consumption within an enterprise [42]. Dynamic cloud-environments also provide a number operations, also called meta-services [43], in order to create these virtual resources from physical resources and to configure them to specific needs. In order to create a cloud-service, a number of these operations have to be executed. There are also a number of approaches addressing the management of virtual resources. Most of them, however, address virtual computing resources only. In [44] the placement of virtual machines is addressed. The basic approach of sharing physical resources more efficiently by virtualizing them is investigated in [45]. A general architecture for resource management is described in [46].

3 Differentiating Fabric-Processes from Business Processes

There is a number of definitions for business processes [3], [47], [48], [49] which are highly overlap. These and other definitions agree upon, that (a) a business processes consists of a set of logically related tasks that are performed to pursue a defined business goal (b) a business process is performed in an organizational environment (c) a business process are executed in a coordinated manner on a set of human and / or other physical resources such as machines or equipment. Fabric-processes differentiate themselves from normal business processes regarding these three points. These differences are depicted in Table 1.

Table 1. The differences between fabric-processes and business processes

	Fabric-processes	Business processes
Pursued goal	IT-related goal / Creation or Configuration of cloud-services	Business goal / Output that is of value to an organization or its customers
Performed in …	Fabric / Cloud-environment	Organizational environment
Involved resources	Virtual resources / Meta-Services	Mainly human resources / Other physical resources
Operators and operands	High identification	Low – high identification

Taking these differences into account, a fabric-process can be defined as the following: A fabric-processes is a semi-ordered set of activities that are executed in a coordinated and collaborative manner in a cloud-environment on a set of virtual resources and meta-services in order to pursue an IT-related goal such as the creation or configuration of a cloud-service.

The most essential difference between fabric-processes and normal business processes is the fact that fabric-processes are executed on a set of virtual resources instead of human resources. In contrast to human resources, it is only possible to interact with virtual resources via predefined operations often referred to as meta-services [43]. Only by calling these predefined operations, a virtual machine can be created form a physical resource or configured to specific needs.

However, the activities of a process are usually defined in a very abstract manner. So, in contrast to human resources, virtual resources are not able to interpret these abstract process activities. Hence, certain mappings has to be defined in order to translate each activity of the process to an appropriate operation provided by the cloud-environment in order to interact with the virtual resource. By this, fabric-processes differ from business processes both in their resource meta-model and their operational meta-model. So, special requirements have to be captured by these meta-models in order to enable a proper specification of fabric-processes.

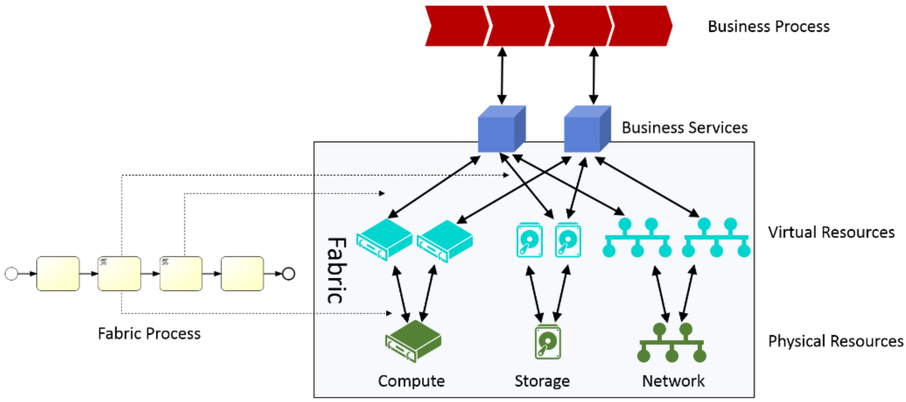


Fig. 1. The interaction between fabric-processes and business processes

Fabric processes are the base for the technical support of many business processes (see Fig. 1). Without these technicial business process support, the process quality (e.g. because of media discontinuity) would be highly limited. For instance, if e-commerce retailers want to make their credit decision processes via a cloud-based credit score provider, various technical (fabric) processes must be provided [50], [51]. Therefore a huge amount of data must be processed in a short time period, using different scalable information systems and their technical base (e.g., different VM's, VLAN's). In the following chapter we systematically address requirements for specifying fabric-processes. We indicate these requirements using Fabric-Process Patterns.

4 Fabric-Process Patterns

In this chapter we describe and discuss a set of Fabric-Process Patterns that were identified by reviewing the conceptual foundations of 16 existing BPMS / WFMS and three commercially available ITPA tools as well as by interviewing several industry experts about specifying and automating fabric-processes. Referring to [52], a pattern is "the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts". The Fabric-Process Patterns are mainly focused on virtual resources. So, there exist certain specialization relationships [35] between the resource patterns described in this paper and those described in [31], [32]. Patterns should be described through [9], [53]: conditions that have to hold for the pattern in order to be applicable, exemplary business situations, semantic problems and potential implementations strategies. Due to limited space, we do not describe specific semantic problems for each of the Fabric-Process Patterns. However, like the original Workflow Control-Flow Patterns [9], [35] and the Workflow Resource Patterns [31], [32], the Fabric-Process Patterns described in this chapter are intended to be language independent too. So, the Fabric-Process Patterns are applicable to a broad range of BPMS and WFMS.

We give a potential implementation strategy for each of the Fabric-Process Patterns, which can be used to extend existing resource meta-models in order to specify Fabric-processes properly. We develop a semantically rich FRMM by extending existing resource and organizational meta-models with this set of implementation strategies. The FRMM can be seen as a mix of a resource meta-model and an operational meta-model, because it also considers the operations or meta-services provided by the cloud-environment for each virtual resource in order to create or configure it. The FRMM is illustrated in Fig. 3 in the form of an UML class diagram. For each pattern, the relevant part of the FRMM is described.

4.1 Pattern 1: Virtual Resource Specification and Classification

Description. The ability to specify virtual resources and their relevant technical properties as well as the ability to classify the virtual resources based on their properties at design time by defining certain virtual resource classes (e. g. computing, storage and networking), to which the relevant virtual resources have to be assigned.

Example. All virtual resources that represent a certain virtual machine have to be assigned to the computing-class. Each virtual resource that represents a virtual machine and provides a storage capacity up to 100 GB has to be assigned to the small-computing-class.

Motivation. A process activity can have a number of functional and non-functional requirements. An activity only can be undertaken by virtual resources that matches its functional and non-functional requirements. Hence, it is necessary to specify the technical properties for each virtual resource. So, based in on its technical properties each virtual resource can be assigned to a certain activity that has to be undertaken. In some cases, it is not clear at design time which specific virtual resource has to undertake a certain activity of the process. But often, it is known which certain class of virtual resources is needed in order to undertake a particular activity of the process. So, by defining virtual resource classes based on the technical properties of the virtual resources it is possible at design time to assign a certain virtual resource class to the particular activity of the process. At runtime, the workflow engine then can decide which specific virtual resource of the particular virtual resource class has to undertake the relevant activity of the process. For example in modern data centers, the workflow engine can select the virtual resource that has the least degree of capacity utilization. Also, it might be necessary in some cases to define a number of subclasses for a specific virtual resource class. The classification of the virtual resources via virtual resource classes and subclusses helps to provide a framework for comparing and structuring the various heterogeneous virtual resources of modern data centers.

Implementation. Fig. 3 illustrates a potential implementation strategy as part of an overall FRMM that can be used to classify the virtual resources. Both, an virtual resource as well as an virtual resource class, are exactly identified by their attached ids. It is also possible to specify the name of the virtual resource and virtual resource class. A virtual resource has at least one technical property. A technical property is exactly identified by its attached id. Each technical property also has a specific name. In order to classify the virtual resource, the value of each technical property has to be specified.

A virtual resource has to be part of up to one virtual resource class. In contrast, an virtual resource class is associated with at least one virtual resource. Though, in practice it might be necessary to create an virtual resource class that is not associated with an virtual resource for a certain time period. In this case, the UML class diagram has to be adapted to this specific kind of situations by setting the minimum cardinality on the virtual resource side of the association to zero. A virtual resource class also can be subdivided into several subclasses. These subclasses also are composed of at least one virtual resource.

4.2 Pattern 2: Definition of an Abstract Virtual Resource Lifecycle and Implementation of the Relevant Abstract Operations Provided by the Cloud-Environment

Description. The ability at design time to define an abstract virtual resource lifecycle specifying the limited set of relevant abstract operations provided by the cloud-environment for each virtual resource in order to interact with it.

Example. In order to create a virtual machine, the create-operation provided by the cloud-environment for virtual machines has to be executed. This and other operations provided by the cloud-environment in order to interact with virtual machines have to be specified in a virtual resource lifecycle that is associated with the virtual machine-class.

Motivation. The interaction between an activity of the process and the relevant virtual resource that has to undertake this activity strongly depends on the actual state of the virtual resource within the virtual resource lifecycle. The virtual resource has to be already defined, created, configured and started in order to undertake the activity. In order to trigger a state transition of a virtual resource, the appropriate operation provided by the cloud-environment has to be executed. However, cloud-environments are consisting of various virtual resources. It is not possible to implement the relevant operation for all the relevant virtual resources in a completely generalized manner. It is obvious that the operation that has to be executed in order to create a certain virtual machine has to be implemented in different way than the operation that has to be executed in order to create a virtual hard-disk. Hence, the virtual resource lifecycle has to specify the state transition operations in an abstract manner. So, the operations are implemented by each virtual resource individually. However, in some cases it might be possible to implement this set of operations for a class or subclass of virtual resources. For example it might be possible to specify and implement only a part of this set of operations. So, these operations have to be extended for each virtual resource individually.

Implementation. As illustrated in Fig. 3, an abstract virtual resource lifecycle has to be defined, which has to specify the relevant operations in an abstract and / or concrete manner. A virtual resource lifecycle has to be associated to exactly one virtual resource class. Thereby, the operations that are implemented for the virtual resource lifecycle in a concrete manner do not have to be implemented or extended for the virtual resources, which are associated with the virtual resource class. So, for each virtual resource associated with the virtual resource class the exact same concrete operation can be used in order to undertake the appropriate elementary actions. However, the operations that are specified by the virtual resource lifecycle in an abstract manner, have to be implemented or extended for each virtual resource individually.

4.3 Pattern 3: Definition and Implementation of an Abstract Set of Elementary Operations Provided by the Virtual Resources Itself

Description. The ability at design time to define and implement an abstract set of elementary operations for a certain virtual resource in order to enable it to interpret and undertake a specific abstract process activity.

Example. A certain virtual machine has to undertake an abstract process activity. In order to enable the virtual machine to interpret and undertake this abstract activity, it has to be mapped to the appropriate elementary operation that is provided by the virtual machine. By executing this elementary operation, the virtual machine is able to undertake the relevant process activity.

Motivation. Process activities are specified in an abstract manner. Though, virtual resources only are able to interpret and execute a limited set of individually implemented operations. In order to enable a virtual resource to interpret and undertake an abstract process activity, this abstract process activity has to be mapped to one of these individually implemented operations. By mapping these individually implemented elementary operations to the abstract activity, the virtual resource is able to interpret and undertake this abstract process activity.

Implementation. A potential implementation strategy for specifying and implementing a set of elementary operations for a virtual resource is illustrated in Fig. 3 as part of an overall FRMM. In order to enable the virtual resources to interpret and undertake abstract process activities, an abstract set of elementary actions has to be specified for each virtual resource class. Thereby, it is possible to implement a set or subset of elementary operations for an entire class of virtual resources in a concrete manner. So, this set or subset of elementary operations do not have to be implemented or extended for each member of this class individually. However, those elementary operations that are only specified in an abstract manner, have to be implemented or extended for each virtual resource individually. A virtual resource (class) can be assigned to an abstract process activity that has to be undertaken. By assigning a virtual resource (class) to an abstract process activity it is possible to map the relevant elementary operations that are implemented for the virtual resource (class) to the abstract process activity. Hence, the virtual resource is able to translate the abstract activity into the relevant elementary operations.

5 A Meta-model for Fabric-Processes

We develop a semantically rich FRMM that can be seen as a mix of a resource meta-model and an operational meta-model that implements the identified special requirements of virtual resources described in chapter four. A meta-model describes the structure of models on a high level [54]. Thus meta models are design frameworks, that define the model elements and the relations between it as well as the semantics [54], [55]. Therefore meta models using modelling techniques (e.g. UML, E-R approach) to model all needed aspects [54], [55]. To define a meta-model and e.g. its elements special requirements of fabric-processes must be observed. In order to be generally applicable, resource meta-models have to satisfy a number of requirements. The requirements can be derived from the four core quality requirements for software systems [56]: flexibility, scalability, robustness and domain-indipendence. Two different approaches can be distinguished when modelling resources [11]: an organizational approach and a technological approach. These approaches have an impact on the design of the resource meta-model. In case of following the organizational approach, the resource meta-model has to depict the organizational structure of the enterprise. Resource meta-models like this are often referred to as organizational meta-models. Following the technological approach, the structure of the resources that are needed in the control-flow meta-model are derived from the control-flow meta-model itself.

There is already a large number of resource meta-models such as the Object-Orientated Organizational Model (OMM) introduced in [57], the Organization and Resource Model (ORM) presented in [13], [12] and the resource management facility introduced by Huang et al. [58], [59]. Bussler described a generic organizational meta-model in [60], [24]. Another organizational meta-model was presented in [11]. These existing resource meta-models mainly focus on representing human resources and their appropriate roles and positions within an organizational structure. So, these existing resource meta-models can be seen as organizational meta-models. The existing organizational meta-models can be summarized by a semantically rich organizational reference meta-model that is illustrated in Fig. 2.

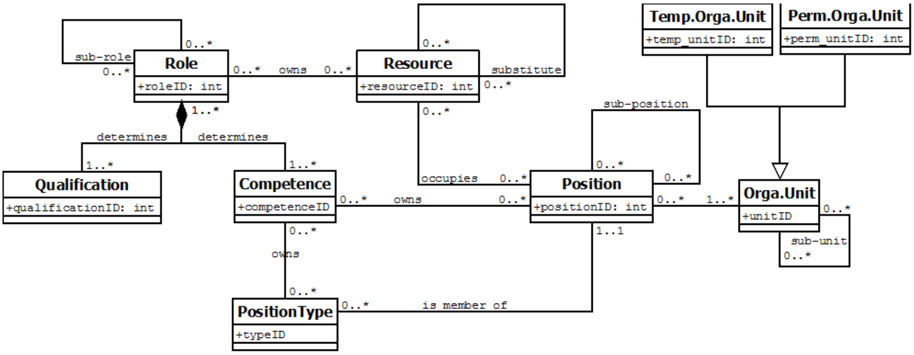


Fig. 2. An organizational reference meta-model [27], [26]

Central class of the organizational reference meta-model is the human resource. A human resource can have a number of substitutes which have to be a human resource too. A human resource can also own several roles which are determined by at least on qualification and competence. A number of positions can be occupied by a human resource. A position has to be part of at least one organizational unit. There are two different types of organizational units that can be distinguished: a temporal organizational unit and a permanent organizational unit.

However, the organizational reference meta-model and so the existing organizational meta-models mainly focus on human resources instead of virtual resources or other non-human resources. A generic resource meta-model has been presented in [11]. This generic resource meta-model extends the organizational reference meta-model by adding a resource type attribute to the resource class. So, the generic resource meta-model distinguishes between human resources and non-human resources. The generic resource meta-model also provides the possibility to group certain resources. These resource groups provide a common set of properties to their members.

Though, this generic resource meta-model as well as the organizational reference meta-model does not capture the identified special requirements of fabric-processes. Both resource meta-models do not include any kind of resource lifecycle for virtual-resources and other non-human resources. The meta-models also do not provide the possibility to specify the technical properties of the virtual resources and other non-human resources in detailed manner. Finally, the meta-models do not provide the possibility to specify and implement certain sets of elementary operations for the virtual resources and other non-human resources. So, the virtual resources as well as other non-human resources are not be able to interpret and undertake the abstract process activities.

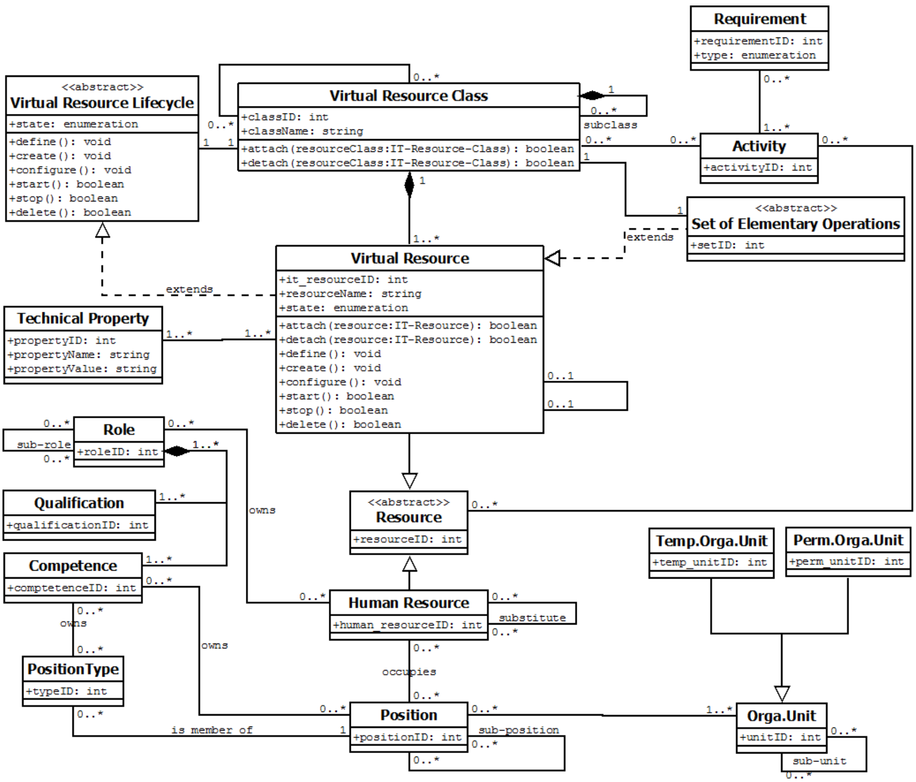


Fig. 3. The FRMM that can be seen as a combination of a resource meta-model and an operational meta-model

In order to fully specify fabric-processes, the organizational reference meta-model has to be extended in a much more detailed way than it was done by the generic resource meta-model introduced in [11]. A meta-model has to capture all the special requirements of fabric-processes and so has to implement the Fabric-Process Patterns in order it can be used to specify fabric-processes properly. In chapter 4 a potential implementation strategy is described for each Fabric-Process Pattern. These implementation strategies can be used in order to extend existing resource and operational meta-models. We extended the organizational reference meta-model with this set of implementation strategies and so developed the FRMM. The FRMM extends the organizational reference meta-model in a detailed manner. Fig. 3 illustrates the FRMM. The FRMM distinguishes between virtual resources and human resources. Thereby, the FRMM defines a concrete virtual resource lifecycle which is extended by the virtual resources. Also, a limited set of concrete elementary actions for each virtual resource is defined. By combining these elementary actions, a virtual resource is able to undertake a certain abstract activity. Each virtual resource has at least one technical property that has to be matched against the technical requirements of the activity that has to be undertaken.

6 Conclusion

Virtual resources provide a number of important advantages for the support of business processes and thus help IT departments to reduce cost and risk. Fabric processes are crucial for the efficient management of virtual resources in cloud-environments. Therefore, the development of a methodology for fabric process design is an important is an important challenge for research. It is the foundation for the proper automation of fabric processes. To pursue these goals, a set of Fabric-Process Patterns that indicate the special requirements of fabric-processes are defined in this paper. We also introduced a FRMM that can be seen as a mix of a resource meta-model and an operational meta-model. This FRMM implements all the Fabric-Process Patterns and so captures all the special requirements of fabric-processes.

Therefore research can contribute on the use of meta-modelling in the area of process automation and validation of it. Practical engineers can may better manage IT automation processes by adapting this approach. Therewith an IT implementation with focus on cost reduction, well quality and reduced implementation time can be applied.

There are some limitations to discuss. Not all business or fabric-processes are specified to implement it without manual steps. For instance in some high regulated sectors (e.g. Pharmacy / health care industry) some fabric-processes must reviewed, configured and implemented manually.

Future research must focus on industry-specific implementations and development of this approach. There can be differences between IT enterprises and enterprises of other sectors (e.g. chemistry), because of the maturity of fabric-processes. Empirical research (e.g. prototyping and expert interviews) are needed to validate and improve our approach.

References

1. Carr, N.G.: IT doesn't matter. Harv. Bus. Rev. 81, 24–32
2. AXELOS Ltd.: ITIL glossary and abbreviations (2011), http://www.itil-officialsite.com/nmsruntime/saveasdialog.aspx?lID=1180&sID=242
3. Weske, M.: Business Process Management Concepts, Languages, Architectures. In: Business Process Management, pp. 305–343. Springer, Heidelberg (2007)
4. Luftman, J.: Assessing It/Business Alignment. Inf. Syst. Manag. 20, 9–15 (2003)
5. 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., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
6. van der Aalst, W., Desel, J., Oberweis, A.: Business Process Management Models, Techniques and Empirical Studies. Springer (2000)
7. van der Aalst, W.M.P.: Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) ACPN 2013. LNCS, vol. 3098, pp. 1–65. Springer, Heidelberg (2004)

8. van der Aalst, W.M.P., Barros, A.P., ter Hofstede, A.H.M., Kiepuszewski, B.: Advanced Workflow Patterns. In: Scheuermann, P., Etzion, O. (eds.) CoopIS 2000. LNCS, vol. 1901, pp. 18–29. Springer, Heidelberg (2000)
9. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distrib. Parallel Databases. 14, 5–51 (2003)
10. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation. International Thomson Computer Press (1996)
11. Muehlen, M.Z.: Resource Modeling in Workflow Applications. In: Proceedings of the 1999 Workflow Management Conference (WFM 1999), pp. 137–153 (1999)
12. Rupietta, W.: Organization and Role Models for Workflow Processes. In: Lawrence, P. (ed.) Workflow Handbook 1997, pp. 165–172. John Wiley & Sons, Inc., New York (1997)
13. Rupietta, W.: Organization models for cooperative office applications. In: Karagiannis, D. (ed.) DEXA 1994. LNCS, vol. 856, pp. 114–124. Springer, Heidelberg (1994)
14. Schael, T. (ed.): Workflow Management Systems for Process Organisations. LNCS, vol. 1096. Springer, Heidelberg (1998)
15. Dogac, A.: Workflow Management Systems and Interoperability. Springer, Heidelberg (1998)
16. Camunda Services GmbH: Camunda BPM platform, `http://camunda.com/bpm/features/`
17. Bonitasoft Inc.: Bonita BPM, `http://www.bonitasoft.com/`
18. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: Yet another workflow language. Inf. Syst. 30, 245–275 (2005)
19. Activiti Committer: Activiti BPM Platform, `http://www.activiti.org/`
20. Ziembicki, D., Cushner, A., Rynes, A.: Microsoft System Center: Designing Orchestrator Runbooks. Microsoft Press, Redmond (2013)
21. IBM SmartCloud Orchestrator, `http://www-03.ibm.com/software/products/us/en/smartcloud-orchestrator/`
22. HP Operations Orchestration software, `http://www.hp.com/hpinfo/newsroom/press_kits/2009/HPSoftwareUniverseHamburg09/HPOODataSheet.pdf`
23. GmbH, F.N.T.: FNT Software - Infrastruktur und Service Management für IT und Telekommunikation, `http://www.fntsoftware.com/`
24. Bussler, C., Jablonski, S.: Policy resolution for workflow management systems. In: Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, 1995, vol. 4, pp. 831–840 (1995)
25. Van der Aalst, W.M.P., Kumar, A.: A reference model for team-enabled workflow management systems. Data Knowl. Eng. 38, 335–363 (2001)
26. Zur Muhlen, M.: Evaluation of workflow management systems using meta models. In: Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, HICSS 1999, vol. 32, p. 11 (1999)
27. Rosemann, M., Muehlen, Z.: Michael: Evaluation of Workflow Management Systems - A Meta Model Approach. Social Science Research Network, Rochester (1998)
28. Muehlen, M.Z.: Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-driven Process Information Systems. Michael zur Muehlen (2004)
29. Lerner, B.S., Ninan, A.G., Osterweil, L.J., Podorozhny, R.M.: Modeling and Managing Resource Utilization for Process, Workflow and Activity Coordination TITLE2. University of Massachusetts, Amherst (2000)

30. Ouyang, C., Wynn, M.T., Fidge, C., ter Hofstede, A.H.M., Kuhr, J.-C.: Modelling complex resource requirements in Business Process Management Systems. In: Rosemann, M., Green, P., Rohde, F. (eds.) ACIS 2010 Proceedings, Queensland University of Technology, Brisbane (2010)
31. Russel, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Ressource Patterns (2005), `http://www.ww.workflowpatterns.com/documentation/documents/Resource%20Patterns%20BETA%20TR.pdf`
32. 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)
33. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns: Identification, Representation and Tool Support. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 353–368. Springer, Heidelberg (2005)
34. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Workflow Exception Patterns. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 288–302. Springer, Heidelberg (2006)
35. Russell, N., Hofstede, A.H.M.T., Mulyar, N.: Workflow ControlFlow Patterns: A Revised View (2006)
36. OMG: Object Management Group. Business Process Model and Notation, `http://www.bpmn.org/`
37. Grosskopf, A.: An extended resource information layer for bpmn. Hasso Plattner Institut, Potsdam (2007)
38. Awad, A., Grosskopf, A., Meyer, A., Wesek, M.: Enabling resource assignment constraints in BPMN. Hasso Plattner Institute, Potsdam (2009)
39. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: Pattern-based Analysis of BPMN. Fac. Sci. Technol. (2005)
40. White, S.: Process modeling notations and workflow patterns (2004)
41. Georgakopoulos, D., Hornick, M., Sheth, A.: An overview of workflow management: From process modeling to workflow automation infrastructure. Distrib. Parallel Databases 3, 119–153 (1995)
42. Schmidt, R.: A framework for comparing cloud-environments. In: 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 553–556 (2011)
43. Schmidt, R.: Augmenting Cloud Requirements Engineering with Meta-Services. In: 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 488–493 (2011)
44. Nguyen Van, H., Dang Tran, F., Menaud, J.-M.: Autonomic virtual resource management for service hosting platforms. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, pp. 1–8. IEEE Computer Society Press (2009)
45. Sotomayor, B., Keahey, K., Foster, I.: Overhead matters: A model for virtual resource management. In: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, p. 5. IEEE Computer Society (2006)
46. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Comput. 13, 14–22 (2009)

47. Davenport, T.H., Short, J.E.: The New Industrial Engineering: Information Technology and Business Process Redesign, http://sloanreview.mit.edu/article/the-new-industrial-engineering-information-technology-and-business-process-redesign/
48. Pall, G.A.: Quality Process Management. Prentice Hall PTR (1987)
49. La Rosa, M.: Business Process Modelling, Brisbane (2013)
50. Schmidt, R., Mohring, M.: Strategic Alignment of Cloud-Based Architectures for Big Data. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 136–143 (2013)
51. Schmidt, R., Möhring, M.: Two-Phase Composition of Complex Business Services in Cloud-Environments. In: Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), Vancouver, Canade, pp. 63–70 (2013)
52. Riehle, D., Züllighoven, H.: Understanding and using patterns in software development. Theory Pract. Object Syst. 2, 3–13 (1996)
53. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston (1995)
54. Rosemann, M., Green, P.: Developing a meta model for the Bunge–Wand–Weber ontological constructs. Inf. Syst. 27, 75–91 (2002)
55. Steele, P.M., Zaslavsky, A.B.: The role of meta models in federating system modelling techniques. In: Elmasri, R.A., Kouramajian, V., Thalheim, B. (eds.) ER 1993. LNCS, vol. 823, pp. 315–326. Springer, Heidelberg (1994)
56. Dunn, R.H.: Software-Quality: Concepts and Plans. Englewood Cliffs (1991)
57. Cheng, E.C.: An object-oriented organizational model to support dynamic role-based access control in electronic commerce applications. In: Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, HICSS 1999, vol. 32, p. 9 (1999)
58. Huan, Y.-N., Shan, M.-C.: Policy-Based Resource Management. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 422–428. Springer, Heidelberg (1999)
59. Huang, Y.-N., Shan, M.-C.: Policies in a resource manager of workflow systems: Modeling, enforcement and management. In: Proceedings of the15th International Conference on Data Engineering, p. 103 (1999)
60. Bussler, C.: Analysis of the Organization Modeling Capability of Workflow-Management-Systems. In: Proceedings of the PRIISM 1996 Conference, Maui, Hawaii (1996)

Comparing Business Process Variants Using Models and Event Logs

Joos C.A.M. Buijs[1] and Hajo A. Reijers[1,2]

[1] Eindhoven University of Technology, The Netherlands
{j.c.a.m.buijs,h.a.reijers}@tue.nl
[2] Perceptive Software, The Netherlands

Abstract. Organizations realize that benefits can be achieved by closely working together on the design of their business processes. But even when there is a joint design for a particular business process, the way individual organizations carry out that process may differ – either wittingly or unwittingly. This paper proposes an analytical approach that helps to compare how different organizations execute essentially the same process. This comparison is based on the alignment of recorded process behavior with explicitly defined process models. The distinctive feature of the proposed approach is that it supports the comparison of the actual execution of a process within a particular organization with its *intended* design, as well as with the *variants* of that design by other organizations. In this way, organizations can develop a better understanding of how they can work together and further standardize a process of common interest. We include an industrial case study from the context of the CoSeLoG project to demonstrate the value of this comparison approach.

1 Introduction

All around us, we see signs of the rise of the 'sharing economy' [7]. We may primarily think of individuals who can collaboratively make use of under-utilized capacity, as in the example of Airbnb[1]. Increasingly, also professional organizations realize the benefits of sharing information, resources, and expertise among them. For example, partners in a supply chain may share market analyses to jointly arrive at better demand forecasts.

Another way of sharing knowledge for organizations is to work according to a consciously designed common plan for their operations. In this way, an autonomous organization may strike a balance between (a) reaping efficiency advantages through a standardized way of working for a particular process and (b) addressing local priorities through incorporating local deviations from that standard plan.

In this paper we focus on one of the challenges that such cooperative yet independent organizations face: Even if they execute a jointly designed process, how to identify the commonalities and differences between the ways they actually work? While some deviations may be planned for explicitly, others may be

[1] http://www.airbnb.com, last accessed on February 19, 2014.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 154–168, 2014.

unexpected. Recent work has indicated that, for example, people can be highly creative in working around intended procedures [16].

The context for this paper is the CoSeLoG project[2] in which a group of 10 Dutch municipalities participate. Five of these have decided to start working more closely together while maintaining their legal autonomy. After having executed a commonly designed process for a prolonged amount of time, they have an interest in the type of comparison we sketch: How is each organization carrying out that process and how do they differ from each other in this respect?

Our contribution is a new analytical technique that allows for a *dual* comparison. In the first place, it allows for a comparison between the intended and the actual execution of a business process. Secondly, it supports the comparison of the execution of process variants, for example in case different organizations carry out a similar process. These comparisons are visualized through a so-called *alignment matrix*. The paper also describes a comparison framework that shows the methodic application of this aid.

This paper extends [3] by its explicit incorporation of the process model in the comparison, i.e. the intended behavior. By (a) replaying the actual behavior on that initial model, as witnessed through event logs, and (b) showing where different organizations deviate, the process model can be used as a common means to compare against. Notably, in [3] there was no means to visualize this comparison. This cross-comparison can help organizations to provide a better understanding of how a process is executed and to act upon that insight. Such actions may be diverse: It may be decided to fix the common process if it allows for too much deviation, but individual organizations may also want to imitate the practices of another partner when these seem preferable.

The remainder of this paper is organized as follows. In Section 2 we will reflect on related work that aims to analyze and compare processes. A running example is introduced in Section 3. The main contribution is provided in Section 5, which is preceded by an explanation of fundamental concepts in Section 4. The overall comparison approach is applied in a case study, which is described in Section 6. Section 7 concludes our paper.

2 Related Work

For a considerable time now, organizations seek to learn from others on how to adapt their own processes towards improved competitiveness [18]. Process *benchmarking*, however, is primarily a manual process, requiring the involvement of experts to collect and interpret process-related data [19]. A main problem that has been recognized is that processes across different organizations are often modeled on different levels of granularity and for different purposes. This makes their comparison hard. Previous research in the area of process benchmarking has mainly focused on semantic approaches to overcome these types of barriers, e.g. [6, 9, 19].

[2] `http://www.win.tue.nl/coselog/wiki/start`, last accessed on February 19, 2014.

In the context of our work, the processes that are to be compared can be considered *variants* of each other [12]. This means that the processes are different, but share essential characteristics through their conformance to a shared set of constraints [13] or their derivation from a common template [8]. Because of this starting point, their semantical matching is not really an issue. Yet, the emphasis of existing work on model variants is on the management, specification, and comparison of *models*, i.e. the design-time perspective on these processes. Our paper widens this scope by incorporating the actual *behavior* of these variants, i.e. the run-time perspective. In other words, we extend process model variant management with analytical approaches that allow for comparing the supposed/intended behavior of processes with their actual execution.

Two categories of approaches with respect to comparing the supposed or intended behavior with the actual behavior of a process can be identified. The first of these encompasses approaches that pursue delta analyses between a *predefined* process model on the one hand and the *discovered* model derived from event logs on the other [5, 10]. Here, also generic approaches play a role that relate to process model matching, cf. [21]. The second category aims to project the actual behavior of a process *onto* the predefined process model, as in [11]. The aim is then to show how individual instances relate to pre-defined process model parts. Our research is most related to the latter category. In contrast to existing work, however, it will specifically build on the notion of *process alignments* [1], which we will discuss in more detail in Section 4. Another innovative angle in this context is our interest in the comparison of multiple, related processes.

Since our work also strongly emphasizes the visualization of the analysis results, it also relates to other approaches that help to better make sense of process models. These cover the usability aspects of the employed notation [15], ways to emphasize the logical relations between model elements [17], and bringing in new perspectives [2], to name a few.

In summary, our work is at the intersection of the streams of *analytical* and *visualization* research to support *process benchmarking* across *process variants*. We extend existing work by taking both the *supposed behavior* and the *actual behavior* of the process variants into account.

3 Running Example

Throughout this paper we use a running example to illustrate our approach. The running example consists of four process model variants, shown in Figure 1, and four corresponding event logs, as shown in Table 1. All four variants describe the process for handling loan applications. Even though the processes differ slightly, each process sends an e-mail (activity `A`) and in the end either accepts (activity `E`) or rejects (activity `F`) the application. The order in which the activities can be executed, however, differs. Moreover, each variant differs as to which activities are included. For instance, whether activity `B` is part of the variant or both activities `B1` and `B2`, which are more fine-grained. The corresponding event logs describe possible executions of the corresponding process model. Please note

Table 1. Four event logs for the four different variants of the loan application process of Figure 1

(a) Event log for variant 1

Trace	#
A B C D E G	6
A B C D F G	38
A B D C E G	12
A B D C F G	26
A B C F G	8
A C B E G	1
A D B C F G	1
A D B C E G	1
A D C B F G	4
A C D B F G	2
A C B F G	1

(b) Event log for variant 2

Trace	#
A B1 B2 C D2 E G	20
A B1 B2 C D2 F G	50

(c) Event log for variant 3

Trace	#
A C B E	120
A C B F	80

(d) Event log for variant 4

Trace	#
A B1 D B2 C E	45
A B1 D2 B2 C F	60

that in this example the traces align perfectly with the corresponding process model. This is generally not the case for real-life processes.

4 Preliminaries

In order to relate the observed behavior to the modeled behavior, so-called *alignments* [1] between cases in the event log and a single run through the process model can be created. Since an observed execution of the process may not always fit the described behavior of the process model, deviations need to be detected. However, often multiple solutions to align a deviation are present. Adriansyah et al. [1] propose a technique to assign cost to particular deviations and to then find the alignment between the observed and modeled behavior with the least

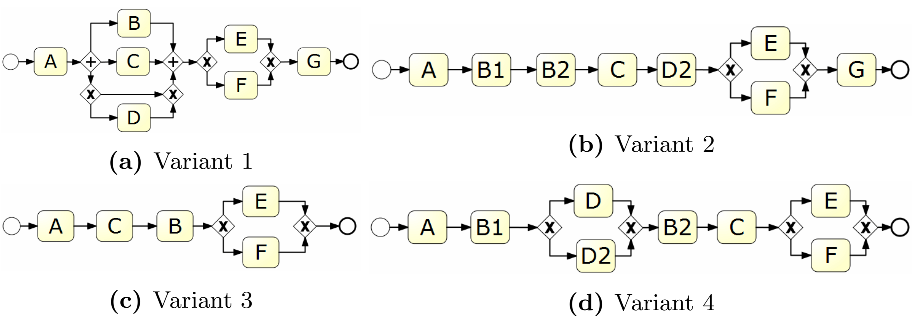


Fig. 1. Four variants of a loan application process. (A = send e-mail, B = check credit, B1 = send check credit request, B2 = process check credit request response, C = calculate capacity, D = check system, D2 = check paper archive, E = accept, F = reject, G = send e-mail).

cost. This technique has been proven to be robust to different variations of deviation and provides detailed insights in the (mis)alignment between observed and modeled behavior.

An example of an alignment between the modeled and observed behavior is shown in Figure 2a. Here, the trace $\langle A, B1, D, B2, C, E \rangle$ from event log variant 4, as shown in Table 1d, is aligned with process model variant 2 of Figure 1b. The first two activities A and B1 can be observed in both the trace and the process model. Since both the trace and the process model move synchronously, we call this a *synchronous move*: This indicates that there is no error and the observed behavior matches the modeled behavior. Next, activity D occurs in the trace, but the process model prescribes activity B2 to take place. In order to obtain an optimal alignment, the best option is to move forward on activity D in the trace, and to do nothing in the process model. This results in a *log move only* alignment step, which indicates that behavior is observed that is not described by the process model. The two following activities B2 and C can be again performed synchronously. Now the process model "expects" activity D2, while in the trace activity E is recorded. The best option is to perform activity D2 in the process model, resulting in a *model move only*. This type of move indicates that certain behavior was expected according to the process model, but was not observed in the trace. Next, activity E can be observed in the trace and executed according to the process, which is again a synchronous move. However, even though the trace is finished, the process model does not yet describe a final state. Therefore, the last alignment step consists of performing a model move only on activity G.

Alignments provide the connection between the expected, modeled behavior of a process to its observed behavior as recorded in an event log. This connection is crucial when using event logs and process models, since deviations between these are commonplace. As we indicated in our discussion of related work, currently only the setting where one event log is aligned with one process model has been investigated. In this paper, we apply this technique to different combinations of event logs and process models, which stems from our motivation to study a process that is commonly designed and used by different organizations. Unfortunately, the existing ways of visualizing alignments, i.e. by showing the traces and process models in detail, is not applicable in such a cross-organizational setting.

5 Facilitating Cross-Organizational Comparison

In order to facilitate the comparison of process models and behavior across organizations, we will first propose a comparison framework in Subsection 5.1. This framework allows to compare different statistics between the organizations. In order to provide more insights into how the behavior of one organization behaves in relation to the way another organization intends to execute that process, we will propose a new artifact, i.e. the alignment matrix visualization in Subsection 5.2. The alignment matrix can hence be used within the comparison framework to get a more detailed insight into the commonalities and differences of process executions between organizations.

Table 2. Application of the comparison framework on the running example, with event log statistic set to number of traces, process model statistic is number of nodes in the model and the comparison statistic is the replay fitness

	Config 1	Config 2	Config 3	Config 4	Log Stat
Event Log 1	1.000	0.644	0.575	0.580	100
Event Log 2	0.622	1.000	0.488	0.745	70
Event Log 3	0.933	0.618	1.000	0.656	200
Event Log 4	0.579	0.795	0.553	1.000	105
Model Stat	15	11	9	14	

5.1 The Cross-Organizational Comparison Framework

In order to compare processes between organizations we proposed a cross-organizational comparison framework in [3]. The framework aims at facilitating a comparison of business processes by usage of both the process models *and* the observed behavior. An example of the comparison framework as presented in [3] is shown in Table 2. The comparison framework distinguishes three types of metrics: process model (quality) metrics, event log metrics, and comparison metrics.

Process model metrics are metrics calculated using only the process model. Some common examples would be the various structural and complexity metrics that exists for process models [14].

Event log metrics are generally related to different performance indicators that can be defined on the process. Simple examples include the number of traces and events recorded, the number of people working on the process, the average trace duration, etc.

The third category of metrics are *comparison metrics*. These relate to comparisons between an event log and a process model. The alignments discussed in Section 4 are an example of a comparison metric.

The comparison framework has been implemented as a plug-in in the ProM framework [20]. ProM is a process mining framework which allows for a simple implementation of different analysis techniques on event logs and/or process models. All results of the comparison framework, including the alignment matrices we discuss later in this section, have been implemented in the 'Comparison-Framework' package. This package is available from the nightly build package repository which can be used by installing the ProM nightly edition[3].

An application of the comparison framework on the running example is shown in Table 2. The specific process model metric chosen here is the number of nodes in the process model. The number of cases in the event log defines the event log metric. Each event log-process model comparison cell displays the replay fitness score [1], calculated on the alignments. Higher values indicate better alignments, which are emphasized by increasingly darker shades of green as background color.

Comparing the size of the event logs, one can see that event log 3 has the most traces, and event log 2 contains the fewest traces. The process model statistic

[3] ProM 6 nightly can be obtained from `http://www.promtools.org/prom6/nightly/`

indicates that organization 3 with 9 nodes has the smallest process model, while organizations 1 and 4 have the biggest process models, with 15 and 14 nodes respectively. When we investigate the replay fitness scores, we can see that the diagonal has a perfect score of 1.000. This means that the process model of each organization perfectly explains the observed behavior. Furthermore, the process model of organization 1 describes the observed behavior of organization 3 quite well. However, the process model of organization 3 does not explain the observed behavior of any of the other organizations very well. Organizations 2 and 4 have reasonable replay fitness scores on each other's process models, which might allow these organizations to start a collaboration.

The simple replay fitness scores give some preliminary insights, but do not provide a deep understanding of the (dis)similar behavior between the different organizations. To provide more in-depth insights we propose the alignment matrix visualization as a comparison metric.

5.2 Visualizing Alignments: The Alignment Matrix

The purpose of the *alignment matrix visualization* is to visualize the alignments, as calculated using both the process model and the event log, in a concise but clear way. However, we do not project alignments on either the event log or the process model. Instead, we want to exploit the utilization of the available space, whether this concerns a display or a physical canvas, to allow for a wider exploration. Furthermore, we synchronize the settings of the different alignment matrices to ensure all matrices are indeed comparable.

The input for the alignment matrix consists of the alignments for the traces of the event log. Figure 2a shows such an alignment between a trace and a process model. The alignment consists of several *alignment steps*. Each alignment step contains information as to which trace and process model it relates. It also contains a relation to an event in that trace or a relation to an activity in the process model or both.

Within the alignment matrix, alignment steps are assigned to one or more cells, which are distributed over columns and rows. An example is shown in Figure 2b. Here, the columns are defined to be the activities, while each row

Trace	A	B	C	≫	D	E	G
Model	A	≫	C	B	≫	E	≫

(a) Alignment between the trace $\langle A, B, C, D, E, G \rangle$ from Table 1a and the process model of Figure 1c

	A	B	C	D	...
Trace 1	(A,A)	(B,-) (-,B)	(C,C)	(D,-)	...
⋮	⋮	⋮	⋮	⋮	⋱
Trace 100	(A,A)	(B,B)	(C,C)		...

(b) A concrete instance of the alignment matrix with alignments steps shown in the cells

Fig. 2. Alignments and the construction of the alignment matrix

is an alignment instance[4]. In this way, each cell contains those alignment steps which for a particular alignment are related to a certain activity. In the example, most of the time there is one alignment step in each cell. An exception is the cell for trace 1 and activity B. Since the alignment contained both a log move and a model move on this activity, this cell contains two alignment steps. Furthermore, since trace 100 (the last trace of event log Table 1a) did not contain activity D, and the process model did not enforce the execution of this activity, the corresponding cell is empty.

Since in general there can be many alignment steps in a cell, we do not show these individual steps. Instead, we aggregate them and express them by various colors:

- If the cell is empty, i.e. there are *no alignment steps*, we color the cell white;
- In case the cell mainly contains *log move steps*, we color the cell black;
- If the cell mainly contains *model move steps* we color that cell gray;
- In case the cell mainly contains *synchronous steps*, we color the cell according to a pre-defined color that is assigned to that activity (red, yellow, green, blue, purple, etc).

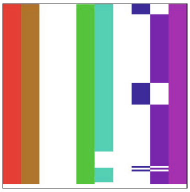
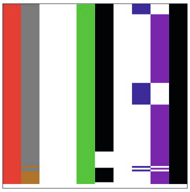
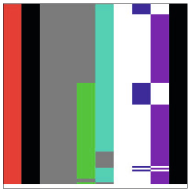
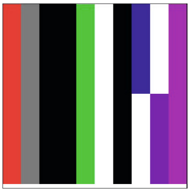
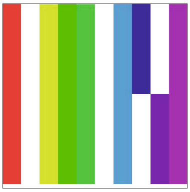
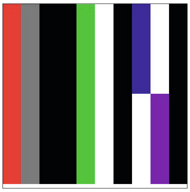
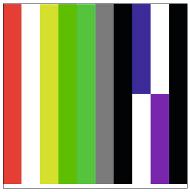
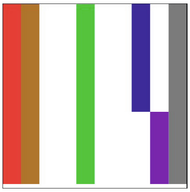
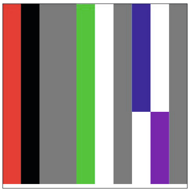
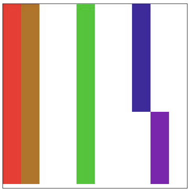
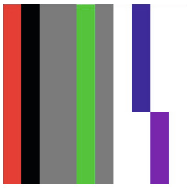
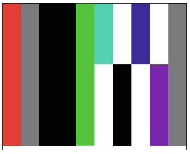
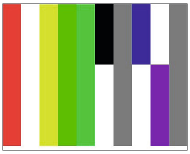
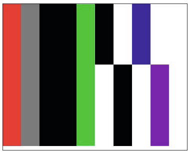
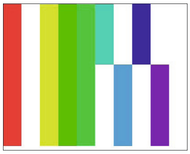
An application of the comparison framework using exactly the settings as discussed is shown in Table 3. Here, the four event logs of Table 1 are replayed on the four process models of Figure 1. Each of these replays is visualized using the alignment matrix. The columns in the alignment matrix represent the activities (A through G), while each row is a single alignment of a trace.

Let us examine, for the example, the replay of event log 1 on process model variant 3. It shows both black and gray cells, which indicate mismatches, log move and model move steps respectively. It can be seen that activity A can be replayed correctly, as indicated by the red color. The gray column, however, indicates that activity B cannot be replayed correctly, except in the last couple of traces as visualized by the orange color in that column.

We can now also further investigate the previous observation that the process model of organization 1 seems to match quite well with the observed behavior of organization 3. The alignment matrix of this combination shows mainly colored columns, but the last column for activity G is completely grey. This indicates that activity G is always a move on model. Therefore, if the process model of organization 1 simply allows the option to skip activity G, the same process model can be used without any problems by organization 3. In other words, these organizations basically work in the same way, which could be exploited in various ways.

[4] Other settings for the column and row definition are possible. One could, for instance, change the rows to represent the different users in the process, and the columns to represent a day or week each. This visualizes when certain users are active and if they execute the activities according to the process model.

Table 3. Application of the comparison framework on the running example. Each column of the comparison framework represents a process model variant from Figure 1 and each row an event log from Table 1. Inside each cell an alignment matrix is shown where the columns are activities, the rows are traces and the color is determined by move type and activity.

	Variant 1	**Variant 2**	**Variant 3**	**Variant 4**
Event log 1				
Event log 2				
Event log 3				
Event log 4				

6 Case Study

In order to validate our comparison framework we applied it on a building permits process. Five municipalities from the CoSeLoG project are collaborating on the building permits process and jointly selected and configured an information system to support this process. However, the five municipalities use their own instance of the system with slightly different settings for each. Moreover, the system allows for some flexibility during the execution of the process. Because of these reasons, several differences still exist in the way the municipalities execute the process. The long-term goal of the municipalities is to centralize and standardize the process to reduce the costs, but this goal can only be attained by making gradual steps. For this reason, it is crucial for the municipalities to understand individual differences between these processes and address them one by one.

In this section we describe the set-up of the case study and how it was executed. We will also provide the insights that we extracted from it.

6.1 Setup

We planned a meeting in February 2014 and invited representatives of each of the five involved municipalities. The meeting was set up to consist of two parts. The aim of the first part is to present general information (number of cases and average throughput time), together with dotted chart and social network visualizations of cases from 2013, detailed along different case types. No process models or activity details are given in that part. Roughly one hour is devoted to this first part.

In the second part, planned to cover approximately another hour, we set out to explain the global idea of the comparison table, i.e. comparing the behavior of a municipality with the discovered model from the behavior of another municipality. The idea for this part is to show the comparison table with the replay fitness scores, as shown in Figure 3a. This is then followed by an example of the alignment matrix (the matrix of event log 1 on variant 4 from Table 3). During a small break of 5 minutes we would lay out the 25 (5 by 5) printouts of the alignment matrices on a table. After the break the idea is then to gather everyone around the table and provide each participant with an individual color marker. In this way, each participant can mark observations on the printouts. During this part, participants are stimulated to make observations and initiate discussion.

We invited seven representatives for the meeting, of which six eventually joined. The expertise from all participating municipalities was present except for municipality 2, whose representative was unable to attend. Fortunately, the representative of municipality 5 also had knowledge about the process in municipality 2. Two representatives were present for municipalities 1 and 3. For each of these two municipalities a coordinator of the process within their respective municipality was present, who both also collaborated in the process. For municipality 4 a building permits expert working in the process was present. As such, these three people had a very good understanding of the whole process. The remaining three representatives were a coordinator of automation and internal affairs (municipality 1), a specialist on internal control and electronic services (municipality 3), and a policy officer for the environmental law (municipality 5). As such, these three people had a more high-level understanding of the whole process, but also detailed knowledge of parts of the process.

The event logs used in the case study contain those cases that were started at some point in 2013 within either of the five municipalities. The logs cover between 150 and 300 cases for each of the municipalities. Both the event logs and the process models contain the 47 most frequent activities across all municipalities. The process models used were automatically discovered using the ETM algorithm [4] based on the data of the event logs. The reason for this is that the municipalities in question immediately configured the information system to their individual preferences without the use of an explicit process model. While the logic of a configuration setting in principle could be translated into a process model, we chose for the use of the discovered model as a reasonable proxy for it.

6.2 Execution

First, before showing the alignment matrices, we showed the replay fitness table as shown in Figure 3a. We first explained that each number roughly corresponded to the number of correctly explained events. The participants quickly noticed that these ratios were not overly high, and in many combinations even very low. They also noticed differences between municipalities. After asking if they could identify distinct clusters of municipalities they replied they could recognize a group consisting of municipalities 1, 3 and 4 which is likely to display highly similar behavior.

Next a small break was introduced and the 25 alignment matrices were distributed on the table, where municipality 2 was moved between municipalities 4 and 5, so that 1, 3 and 4 (as a group of similar municipalities) were close together.

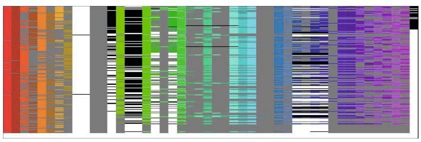
One of the first things that was noticed on basis of the alignment matrices was that there were a considerable number of black cells, which the participants understood to be 'bad'. One of the representatives of municipality 3 contributed that he noticed that each municipality has significantly less black cells on their own models, which can be expected.

Another observation made was that one municipality had a lot of white cells in the alignment matrix. From this, the participants concluded that the specific case types dealt with by this municipality could be different than that of the others, since they show different activities in their behavior.

Based on the alignment matrix of municipality 4 on its own model two observations were made, as is shown in Figure 4b. The first observation, denoted by the bigger blue circle on the left, is that first of all there is not much black

	M1	M2	M3	M4	M5
M1	0.800	0.530	0.764	0.694	0.560
M2	0.482	0.567	0.431	0.458	0.560
M3	0.761	0.511	0.829	0.724	0.551
M4	0.736	0.482	0.752	0.854	0.535
M5	0.477	0.567	0.419	0.459	0.631

(a) Comparison table shown with replay fitness scores



(b) Example of one of the 25 alignment matrices shown, more specifically of the behavior of municipality 5 on the model of municipality 1

Fig. 3. Two of the analysis results shown to the case study participants

('zwart' in Dutch) visible in this matrix. A second observation, made by two of the participants, was that in two of the columns there was a mix of color and black. They correctly concluded that this was caused by sometimes correctly executing this activity, and sometimes deviating from the process model.

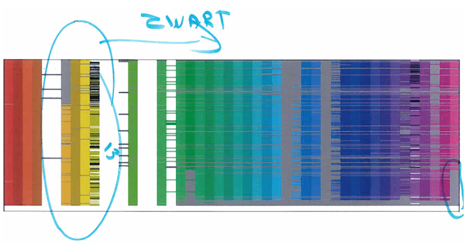
The other observation made on this matrix, as indicated by the blue circle in the bottom-right, is that the last activity shows a lot of grey for the last few traces. A brief remark by us that the cases were sorted from old in the top rows to the newer cases in the bottom rows, quickly resulted in the correct conclusion that these cases did not reach that particular activity in the process just yet. Quickly after this, the participants observed that some of the newer cases actually were further along in the process. They expected a diagonal line from bottom left to upper right. They also noticed that this was not the case for all municipalities.

All-in-all, 22 observations were counted. Each of these triggered a discussion and an exploration of explanations for it between the participants. In the end, one participant remarked that he would like to rearrange the alignment matrices and only show the matrices of the replay on the municipalities' own process models. In this set-up, another 11 observations were made.

We then gently ended the discussion and asked the participants if they thought this approach was easy to understand and use. Although we noticed that the participants seemed somewhat overwhelmed during the introduction of the rather colorful pictures in the beginning, this was not mentioned during the evaluation by themselves. All people involved noted that detecting the grey and black, and also white, worked well. Furthermore, the colors helped participants in relating parts of the process across alignment matrices. A further remark that was made is that the colors made it easy to distinguish between irregular behavior and more structured executions of the process: This was considered highly useful.

From a content perspective, the participants expressed satisfaction with what they could observe using the alignment matrices. From the various insights that were obtained on basis of observing the alignment matrices of each others pro-

(a) Photo of the set-up and the participants. Participants faces are obstructed, the third person from the left is the first author.



(b) The alignment matrix of municipality 4 on its own model, shown with annotations

Fig. 4. A photo and an annotated alignment matrix with some observations made during the case study

cesses, we provide two striking examples. The first of these relates to the observation that one municipality actually did not execute certain steps, while their products still adhered to the regulations. All people involved mentioned that it would be valuable to investigate whether this way of working could be adopted by all municipalities. Secondly, the participants could recognize the effect of a change of personnel within a particular municipality. They expressed that it would be interesting to keep on following the execution of the process to see if the caused behaviorial differences would stabilize over time.

Improvement suggestions with respect to the comparison approach were also made. One suggestion was to add more *visual anchors*, which could help to better determine the location in the alignment matrices. This would help to remember which column represented which part of the process, and for the rows in which month the case arrived. Also, the participants would favor additional features for increased interactivity with the data. They were particularly interested to select specific case types for comparison to see how these compared across municipalities. Another interactive feature they proposed was to select only certain users, since they had the intuition that certain users performed well (or badly), in particular municipalities.

6.3 Results

The goal of our visualization is to provide insights in the commonalities and differences in behavior between organizations. After a brief explanation of the alignment matrix, we let all participants observe and discuss based on alignment matrix printouts. We noticed that some picked-up how to read the figures quicker than others, but after a few minutes almost all participants joined in the discussion. All but one of them regularly made observations, supported other observations or came with possible explanations for observations. Moreover, only very few times was actual input from the organizers required to clarify certain things after the initial explanation of the alignment matrix. Overall, we counted over 30 observations in about half an hour of discussion, which underscores how helpful the approach is to compare the involved processes.

One of the main comments we noted was that the participants would favor more interaction opportunities with the visualization. By hovering over a cell they would like to see more details of that cell, such as the activity, resource and case involved. They also showed real interest in the ability to filter on case types and resources, in order to validate certain assumptions they would have.

However, the main thing we noticed was that the alignment matrices, and the comparison of process executions in general, triggered a lot of discussions between the participants. Participants often asked each other questions of the type "But how do you do this?", or "Why are you faster?", or "Does this role perform this type of activity?". We see these as an indication that the comparison approach triggers a meaningful discussion based on actual analysis results.

7 Conclusion

Organizations increasingly pursue ways to share knowledge about the design and execution of their business processes. In this paper, we presented an analytical approach for the comparison of *similar* business processes across *different* organizations. Since the execution of a business process can deviate from the prescribed way as recorded in the process model, we use both the *observed* behavior as stored in an event log and the *described* behavior as specified in a process model. In order to provide more detailed insights into the way the observed and modeled behavior inter-relate, we proposed the *alignment matrix* visualization. Using a case study, we demonstrated the applicability of the alignment matrix visualization as part of an encompassing comparison framework.

Based on the feedback obtained during the case study we plan to follow up on the presented work by improving the interactivity that interested parties can have with the alignment matrix. Although the rows and columns of the matrix can be flexibly configured, the representatives of the municipalities indicated their wish for further filtering features on the visualized cases. In particular, we received the request to emphasize information on the time and resource perspectives. It would be interesting anyway to focus more on the commonalities and differences between processes on other dimensions than control flow. Additionally, more information about particular cells can be provided, for instance by selecting them and providing on-demand, aggregated information.

On a more abstract level, we plan our future work to focus on the development of additional analytical techniques that help organizations to synchronize and standardize their operations. We believe there are still various opportunities to improve on this highly beneficial but as-of-yet laborious endeavor.

Acknowledgements. The authors would like to thank the representatives from the municipalities for their participation in the case study. Furthermore, the authors would like to thank Wil van der Aalst and Boudewijn van Dongen for their input and Dennis Schunselaar for his help during the case study session.

References

1. Adriansyah, A., van Dongen, B., van der Aalst, W.M.P.: Conformance Checking using Cost-Based Fitness Analysis. In: Proceedings of EDOC. IEEE Computer Society (2011)
2. Betz, S., Eichhorn, D., Hickl, S., Klink, S., Koschmider, A., Li, Y., Oberweis, A., Trunko, R.: 3D Representation of Business Process Models. In: MobIS (2008)
3. 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)
4. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In: Meersman, R., et al. (eds.) OTM 2012, Part I. LNCS, vol. 7565, pp. 305–322. Springer, Heidelberg (2012)

5. Esgin, E., Senkul, P.: Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 296–304. Springer, Heidelberg (2011)
6. Fill, H.-G.: Using Semantically Annotated Models for Supporting Business Process Benchmarking. In: Grabis, J., Kirikova, M. (eds.) BIR 2011. LNBIP, vol. 90, pp. 29–43. Springer, Heidelberg (2011)
7. Friedman, T.: Welcome to the 'Sharing Economy' (2013), `http://www.nytimes.com/2013/07/21/opinion/sunday/friedman-welcome-to-the-sharing-economy.html` (last accessed on February 19, 2014)
8. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and Management of Process Variants. In: Handbook on Business Process Management 1, Springer, Heidelberg (2010)
9. Juan, Y.C., Ou-Yang, C.: A Process Logic Comparison Approach to Support Business Process Benchmarking. The International Journal of Advanced Manufacturing Technology 26(1-2) (2005)
10. Kleiner, N.: Delta Analysis with Workflow Logs: Aligning Business Process Prescriptions and their Reality. Requirements Engineering 10(3) (2005)
11. Kriglstein, S., Wallner, G., Rinderle-Ma, S.: A Visualization Approach for Difference Analysis of Process Models and Instance Traffic. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 219–226. Springer, Heidelberg (2013)
12. Lu, R., Sadiq, S.: Managing Process Variants as an Information Resource. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 426–431. Springer, Heidelberg (2006)
13. Lu, R., Sadiq, S.: 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)
14. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. LNBIP, vol. 6. Springer, Heidelberg (2008)
15. Moody, D.L.: The "physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 35(6) (2009)
16. Outmazgin, N.: Exploring Workaround Situations in Business Processes. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 426–437. Springer, Heidelberg (2013)
17. Reijers, H.A., Freytag, T., Mendling, J., Eckleder, A.: Syntax Highlighting in Business Process Models. Decision Support Systems 51(3) (2011)
18. Rolstadås, A.: Performance Management: A Business Process Benchmarking Approach. Chapman & Hall (1995)
19. Teuteberg, F., Kluth, M., Ahlemann, F., Smolnik, S.: Semantic Process Benchmarking to Improve Process Performance. Benchmarking: An International Journal 20(4) (2013)
20. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 60–75. Springer, Heidelberg (2011)
21. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP Framework: Identification of Correspondences between Process Models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)

Business Process Modeling: A Multi-perspective Approach Integrating Variability

Oumaima Saidani[1] and Selmin Nurcan[1,2]

[1]Université Paris 1 Panthéon-Sorbonne
Centre de Recherche en Informatique
90, rue de Tolbiac 75634 Paris cedex 13 France
[2] Sorbonne Management School
Université Paris 1 Panthéon-Sorbonne
21, rue Broca 75005 Paris France
{Oumaima.Saidani,Selmin.Nurcan}@univ-paris1.fr

Abstract. In the current economic and technological context, changes of different kinds affecting the organization and its processes are inevitable. They can come from government regulations, the emergence of new competitors, the resources availability, etc. To maintain their efficiency and competitiveness, organizations are constrained to adapt their processes continuously to these changes. Thus business processes have to be efficiently modeled in order to give them their capacity to be adaptable. In addition, the factors whose variations require changes in the processes execution have to be identified and formalized. We introduce in this paper a multi-perspective approach for business process modeling which include five perspectives, i.e. the intentional perspective, the organizational perspective, the functional perspective, the non-functional perspective and the non-organizational resource-perspective. The proposed approach integrates variability - in both organizational and functional perspectives - providing several possible representations of the same process, it also allows to capture change factors related to roles of actors and quality requirements. Furthermore, it allows taking into account change factors related to the context.

Keywords: Business process modeling, Multi-perspectives, Variability, Role, Context-awareness, Adaptability, Non-functional requirements.

1 Introduction

The BPM aims to help organizations to improve their efficiency by the means of a better coordination of the human resources and the systems [13]. The benefits of BPM are multiple, in particular in the improvement of the productivity and the quality of services. This fact explains the great interest that the research focuses on this area and particularly on the definition of adaptive business process models. Indeed, several change requirements exist and require the adaptation of business process models according to these requirements which can be related to the context, to the quality, etc.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 169–183, 2014.

Furthermore, many deviations with regard to the predefined process model can be observed at run-time. These deviations can be explained by a rigid definition of the business process model that takes into consideration only idealized and limited modeling situations. Furthermore, For the most part of the business process modeling approaches, the change requirements as well as the non-functional requirements are not taken into consideration. Certain processes parts can be performed in a similar manner. For example, the "the Order to Cash" process is present in a vast majority of organizations. But while sharing common characteristics, this process can vary from one company to another. Despite these differences, it would be inefficient for an organization to start from scratch each time it models business processes regardless of existing business process models. Reference process models such as SCOR (Supply Chain Operations Reference) or the SAP model [21], are designed to enable the systematic reuse of proven parts in projects of (re) design process. Ideally, analysts use reference models gathered in libraries of business process models with their associated documentation for deriving process models meeting the specific needs of the organization. Thus, the reference process models provide an alternative to design process models "from scratch" [22]. However, they do not allow representing variation points while highlighting those that are different.

This paper introduces a multi-perspectives business process modeling approach integrating variability. Our aim is to be able to represent business processes in a way to give them their capacity to be adaptable, on the one hand, and to identify and to formalize the factors whose variations require changes at run-time (i.e. context, and quality requirements), on the second hand. The proposed approach allows to build several possible representations of a business process and to capture change requirements that affect the process execution.

The reminder of this paper is organized as follows. Section 2 introduces a meta-model for business process representation. We discuss in Section 3 the contextualization of business process models based on the proposed meta-model. In section 4, we briefly discuss adaptability issues. Section 5 introduces related work. Finally, we conclude in section 6.

2 A Meta–model for Business Process Representation

We introduce in this section the concepts of the proposed meta-model BPVM (Business Process Variability meta-Model). Fig.1 shows the meta-model BPVM using the notation of UML class diagram. The proposed meta-model include five parts that cover the following perspectives: the intentional perspective, the functional perspective, the organizational perspective, the non-functional perspective and the non-organizational resource- perspective. The following sections describe the concepts of the different perspectives of BPVM. In order to illustrate the proposed concepts, we choose examples from two case studies: the process of reservations and purchases of tickets and the process of loan handling. As shown in Fig.1, the core concept in BPVM is that of *business process fragment* (*BPF*). The perspectives of the meta-model are interconnected through this concept.

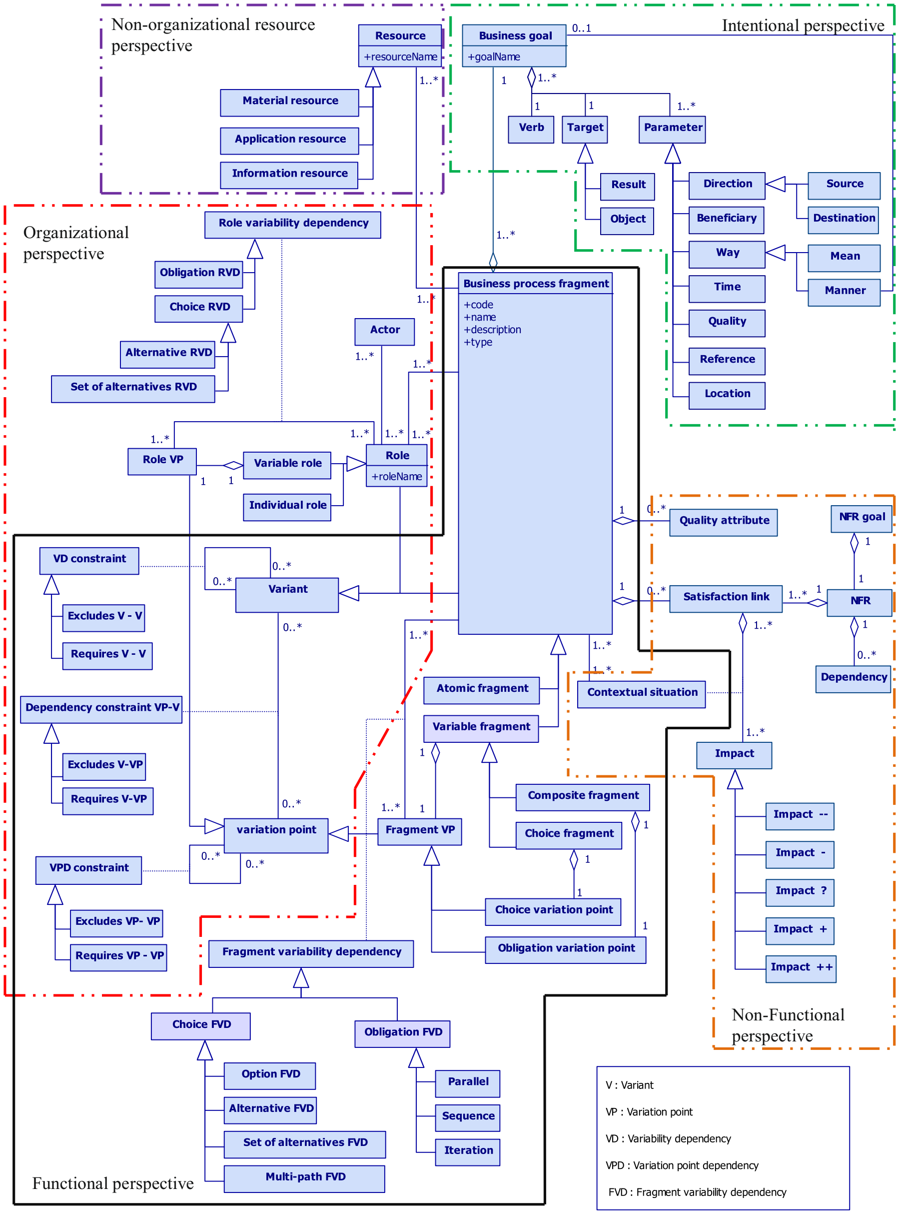


Fig. 1. Business Process Meta-model

2.1 The Intentional Perspective

The intentional perspective allows expressing the goals that processes have to meet. It represents the intentional perspective in business process modeling which is represented by the fact that a BPF achieves a goal. The relationship between BPFs and goals which can be achieved by these BPFs is formalized by the link between the classes *Business process fragment* and *Business goal*. In BPVM, a business goal specifies an objective that we have to achieve without detailing how to achieve it. It identifies the needs and the expectations attached to a business process. We define a business goal as an objective of the organization in carrying out its activities which is satisfied through the realization of one or several BPFs. Some kinds of business goals may be common to many organizations (e.g., supplier invoice handling) while others are specific to a business and/or to a given organization.

In order to formalize business goals, we use a linguistic approach that is based on the formalism proposed in [5] and [6]. This formalism provides a support for the business processes engineering based on goals [5]. It supports goal reduction allowing to detail goals in order to make their definition operational. There are two types of goal reduction: *AND reduction* and *OR reduction*. For an *AND reduction*, for satisfying a given goal, all its sub-goals have to be satisfied. For an *OR reduction*, the satisfaction of a sub-goal is sufficient for satisfying a given goal. Reducing a goal stops when the goal can be operationalized, that is to mean that when all of its sub-goals can be directly satisfied by carrying out actions under agents' control [5]. In BPVM, the OR operator is used to define alternatives and thus to express variation points. The AND operator allows to decompose a business goal into sub-goals.

The linguistic template of a goal includes a verb, a target and a set of parameters that play specific roles related to the verb. For example, the *Way* parameter describes the way in which the goal can be met [6]. The list of parameters is as follows: *Source* and *Destination* (which are generalized by the parameter *Direction*), *Means* and *Manner* (which are generalized by the parameter *Way*), *Beneficiary*, *Time*, *Quality*, *Reference* and *Location*. The verb and the target are mandatory, whereas the parameters are optional. The target designates the entity affected by the goal. It can be of two kinds: *object* or *result*. The object refers to the used entity; it exists before the goal is achieved. The result represents the entities that are affected by the goal; it can be of two kinds: (i) entity that does not exist before achieving the goal, (ii) abstract entity that exists in an abstract form but is made concrete as a result of the goal achievement. The direction parameters are of two types: *source* and *destination*. The source identifies the starting point. The destination identifies the location of entities produced by achieving the goal. The beneficiary refers to a person or to a group of persons in favor for whom the goal is achieved. The way is specialized in two parameters: (i) the parameter *means* which defines the entity (e.g. the tool) by which the goal has to be accomplished; and (ii) the parameter *manner* that defines the way in which the goal is achieved. The *time* situates the goal in time. The *quality* defines a property that has to be attained or preserved. The *reference* refers to the entity with regard to which an action is performed or a state is maintained. The different actors' intentions and the different ways allowing to achieve them require to define variations in the business process model. These variations are expressed in the functional and the organizational perspectives of BPVM. The two following section deal respectively with these two perspectives.

2.2 Dealing with Variability in the Organizational and the Functional Perspectives

A business process model is composed of a set of BPFs which can be achieved in different contexts and by different actors that can have various preferences on the manner in which their intentions are achieved. Thus, a BPF can be achieved in different ways. This fact requires to define the different alternatives for the accomplishment of a business process model.

Variability Modeling. In order to represent the variability in BPVM, we introduce the key concepts of variability: *variation point* and *variant* which are based on OVM (Orthogonal Variability Model) [14]. In our approach, we consider the BPFs and the roles as the variability units. We extend OVM by the concepts that are specific to our meta-model: *role* and *BPF*. These two concepts refer to the concept of variant in OVM. Fig.1 shows the meta-model of OVM extended by the concepts of BPF and role (which specialize the concept of variant in the original model) as well as the concepts of *variation point role* and *variation point fragment*. According to the meta-model, a variation point is a point in the business process where a change occurs indicating the existence of various realization alternatives. A variant is a possible alternative related to a variation point. The variants and the variation points are connected by *variability dependencies*. The variability dependencies can be of two types: *choice* and *obligation*. As shown in Fig.1, we define the dependency constraints between the variants, between the variation points, and between the variants and the variation points.

The Dependency Constraints. The dependency constraints between the variants, between the variation points, and between the variants and the variation points are rules that have to be followed to ensure the consistency of the business process instances. We distinguish two types of dependency constraints similar to those defined by FODA (Feature Oriented Domain Analysis): the *Requires* constraint and the constraint Excludes.

- The *Requires* constraint means an "involvement", that is to say that if an alternative is chosen, another one have to be chosen. This constraint specifies that the selection of a BPF (respectively a role) requires the choice of another BPF (respectively another role) in the same business process instance. *Requires V –V* means that the selection of a variant V_i requires the selection of a variant V_j (regardless of the variation points to which they belong). *Requires PV- PV* means that a variation point VP_i requires the selection of a variation point VP_j.

- The *Excludes* constraint means a mutual exclusion, for example, if a variant V1 - related to a variation point PV1- excludes a variant V2 (related to a variation point PV2), then the variant V2 can not be selected at PV2 if the variant V1 is chosen at PV1. This constraint can specify for example that the choice of a BPF (respectively a role) prohibits the selection of another fragment (respectively of another role) in the same business process instance.

2.3 The Functional Perspective

The functional perspective represents the BPFs by specifying their functional composition of units of finer granularity. This composition follows a hierarchical structure whose leaves fragments represent atomic processes. For example, in the business process “Loan handling”, the BPF “Request evaluation” is an atomic fragment.

This perspective represents a business process model in terms of BPFs which have to be achieved as well as their structures, the composition links and the variability dependencies between them, and the conditions and the constraints governing their achievements.

The Concept Of Business Process Fragment (BPF). A BPF is defined as a part of a business process model that (i) creates value for the organization, (ii) can be reused in several process models, (iii) can be placed under the responsibility of one or more roles (iv) and whose implementation allows to satisfy a business goal. This concept aims to define multiple levels of abstraction. It is similar to the concept of sub-process defined by the WfMC [10] and the OMG [11]. This concept is useful for defining reusable components that allow to build other business process fragments in several process models. BPFs define the structure of a process and they can cover the following modeling situations: *atomicity*, *composition*, *sequence*, *parallelism*, *optionality* and *choice* (*alternative* or *multiple*). Most approaches of business process modeling, such as the workflow control patterns defined in [12], take into account these modeling situations; however they do not deal with all the needs related to the reuse, the modularity and the intentionality. The concept of BPF that we propose allows to define modular and reusable components which are linked to goals to satisfy.

Expressing Variability in the Functional Perspective. The composition links and the variability dependencies between the BPFs as well as the dependency constraints expressed in this perspective are based on the variability model OVM. As shown in Fig.1, we define two types of BPFs: *variable fragments* and *atomic fragments*. In the remainder of this section, we detail each type of fragment as well as the other concepts related to the functional perspective of BPVM.

Atomic BPF. It is a BPF that is associated to an operational goal for which a sequence of operations is defined. Atomic BPFs may be associated to business process models, using a standard business process modeling language (such as EPC) which can be translated into a process execution language such as BPEL.

Variable BPF. It entails variability in its composition or in the way of its achievement. It can be composed of other BPFs. It can also have several possible manners allowing its achievement. Thus, the class *Variable BPF* (see Fig.1) is specialized in the classes: *Choice fragment* and *Composite fragment.* A variable BPF locates the point where the variation is possible as well as each achievement alternative. A variation point is assigned to each variable BPF.

Fragment variation point. It is a representation of one or more places to which an obligation of selection or a choice decision is attached. The choice decision is made based on the intention of the actor, the context, the responsible role and the desired quality properties. Each variable BPF is associated to a fragment variation point.

Fragment variability dependency (FVD). It is a relationship which characterizes the association of a BPF to a variation point. Fig.1 shows two types of FVD: obligation and choice. An obligation FVD can be of three kinds: parallel, sequence or iteration. A choice FVD can be of four types: option, alternative, set of alternatives or path.

Composite BPF. It is a BPF that includes other atomic and/or variable BPFs. As shown in Fig.1, we distinguish three types of variability dependency: *Sequence*, *Parallel* and *Iteration*. Sequence BPFs, parallel BPFs and iteration BPFs establish links of kind AND between the component fragments. They also allow to move from a given granularity level to a finer level.

Sequence BPF. It is a BPF which comprises two or more BPFs and which the associated goal satisfaction requires the satisfaction, sequentially, of goals associated with fragments that compose it.

Parallel BPF. It is a BPF that consists of two or more BPFs and whose satisfaction of the associated goal requires the satisfaction, in a simultaneous manner, of the goals associated to the BPFs that compose it.

We consider the business process "Booking and purchasing air ticket", the payment of a reservation can be made by the mean of a credit card and / or a check. At run time, the purchaser have to select at least one payment mean.

Iterative BPF. It is a BPF whose associated goal satisfaction requires the repeated achievement of the same set of operations which compose the BPF while a condition is not met (it is equivalent to while programming). The condition is reviewed at each loop.

Choice BPF. It allows to model a situation that requires the exploration of different alternatives: situations in which there are different ways to achieve a goal. This concept allows to introduce variability in the way of achieving the goal associated with the BPF. A choice BPF corresponds to an OR decomposition in alternative BPFs in order to satisfy the associated goal. Achieving the goal of a choice BPF consists in choosing the best alternative which is suited to the situation and to achieve it. The variants of a choice BPF can have differences on its achievement with regard to resources, roles, etc. By using the concept of choice BPF we can prevent the multiplication of business process models as well as the deviations from the initially defined business process model. We distinguish four kinds of choice BPFs: *Alternative*, *Set of alternatives*, *Option* and *multi-Path*. The number of BPFs that can be chosen at a variation point depends on the kind of the choice BPF. This number is restricted by the cardinality (*min*, *max*).

Alternative BPF. It is a BPF that expresses a variation in the process by grouping the fragments which are mutually exclusive. It is composed of a set of alternatives linked with an exclusive choice dependency which express an exclusive choice between the fragments; at run time, only one alternative is selected. Each alternative represents a different way to achieve the goal associated to the BPF.

Set of alternatives BPF. It is a BPF that establishes an OR link between the component fragments and offers choices in the manner of achieving the goal associated to a given fragment. It expresses variability in the business process model by grouping a set of BPFs from which at least one fragment is chosen.

Multi-path BPF. It expresses a variation that focuses on alternative BPFs considered individually. It includes a variation in the path of goals to satisfy. Each possible combination of intermediate goals constitutes a distinct path.

Option BPF. It is a BPF whose selection at run time is optional. As shown in Fig.1, the functional perspective is linked to the other perspectives through the class Business process fragment. Thus, the meta-model represents explicitly the business goals that the BPFs have to achieve, the roles that are responsible for their achievements, the resources used by the BPFs, and the quality attributes associated with them. Furthermore, the BPFs are contextualized. The contextual conditions required for their execution are formalized by the class *Contextual situation*. The link between the classes *Business process fragment* and *Contextual situation* expresses the relationship between the meta-model BPVM process and the context meta-context that we will present in future works.

2.4 The Organizational Perspective

This perspective allows to express the organizational resources which are required for the business process realization. These resources are the actors and the roles they play. The core concept in this perspective is that of *role*. In addition to the actors and the roles, the organizational perspective expresses the variability dependencies between the roles. Like the dependencies of variability between process fragments, the dependencies between the roles are based on the variability model OVM. In the remainder of this section, we detail the concepts of *role* and *actor* as well the other concepts related to the organizational perspective of BPVM.

The Concept of Role. We define a role as an organizational entity which is responsible for the achievement of a BPF and that can be assigned to one or more actors. A role can represent a skill, a competency or qualification, e.g teacher, or an authority or a responsibility, such as director. It can also represent a group of individuals, for example, a team. The concept of role is also considered as a means allowing to assign the actors to the BPFs instances. This concept is similar to the concepts of *business role* and *business entity* defined in BPMN, to the concept of *organizational unit* defined in EPC, and to the concept of *organizational role* defined by the WfMC [10]. As shown in Fig.1, we define two kinds of roles: *individual role* and *variable role*.

The Concept of Actor. An actor is a resource that is involved in the execution of a process instance fragment since it is assigned to a role responsible for the achievement of this fragment. An actor is assigned to one or more roles based on their qualifications and skills. An actor may be responsible for the achievement of one or more instances of BPFs according to the roles they can play. This concept is similar to that of *participant* defined by the WfMC.

Expressing Variability in the Organizational Perspective. A BPF can be achieved under the responsibility of several actors playing different roles. At the run-time, the most suitable role is selected. We represent in our approach the variability in the organizational perspective using particularly the concept of *variable role*. Roles and variability dependencies between them constitute a role hierarchy whose leaves represent individual roles. The purpose of this representation is to provide a mechanism for flexible assignment of the BPFs to the actors playing various roles. Thus, the same BPF can be achieved by different roles in different situations.

Individual role. An individual role is a role that does not include other roles. Director is an example of individual role.

Variable role. A variable role is an entity that expresses an organizational variability by grouping a set of roles. We identify three kinds of variables roles: (i) *composite role* which consists of two or more roles, (ii) *alternative role* which includes mutually exclusive roles and (iii) *set of alternatives-roles* which includes a set of roles from whom at least one role is selected at run-time. A variation point is associated to each variable role.

Role variation point. A role variation point is one or more places in a hierarchy of roles to which an obligation of selection or a decision of choice is attached. Each variable role has an associated variation point role.

Role variability dependency. Role variability dependency (RVD) characterizes the link between a role and a variation point. We identify two kinds of RVD: *obligation RVD* and *choice RVD*. Choice RVD is specialized in two types: *Alternative RVD* and *set of alternatives RVD*.

Composite role. Some BPFs are placed under a collective responsibility which involves several roles. For example, the BPF "Loan evaluation by financial pre-evaluation strategy" is achieved under the responsibility of the following roles: "Agent", "Financial Service" and "Loan Manager". The participation of the above-mentioned roles for achieving the BPF is mandatory. Thus, the definition of a composite role including these three roles expresses the collective responsibility of them. We define a composite role as a combination of two or more roles that expresses a collective responsibility. Assigning a composite role to a BPF expresses the fact that the business fragment process is achieved under the responsibility of all roles which compose the composite role. The obligation variability dependency establishes an AND link between the corresponding roles. In the example of loan handling business process, the evaluation of a loan request with a financial strategy is under the responsibility of a composite role "Team_of_evaluation_with_a_financial_strategy" which is composed of the following roles: "Agent", "Loan Manager" and "Financial Service".

Alternative role. An alternative role is a role that expresses an organizational variability by grouping the roles that are mutually exclusive. It consists of a set of roles related by an exclusive choice dependency: only one role is selected for the achievement of a BPF.

Set of alternatives role. A set of alternatives role is a role that expresses an organizational variability by grouping roles from which at least one role must be selected for the achievement of a BPF. A set of alternatives role establishes an OR link between a set of roles.

The organizational perspective is related to the functional perspective through the relationship between the classes *Business process fragment* and *Role*. This relationship represents the fact that a BPF can be performed under the responsibility of one or more roles and a role may be responsible for the realization of one or several BPFs.

2.5 The Non-functional Perspective

This perspective formalizes the non-functional requirements that a business process have to meet and the qualitative goals of the organization which allow improving the quality of the business processes.

This section deals with modeling the quality requirements related to business processes as well as the satisfaction links between the goals and the BPFs, and the impact values according to the context. "Accuracy", "safety", and "flexibility" are examples of quality requirements. We follow a top-down approach which begins with the study of the desirable quality features related to a business process family. These features are considered as goals to be achieved by the organization, from which other goals can be diverted. We use the concept of *Soft-goal* proposed in [15] in order to model non-functional business goals.

The non-functional perspective of the meta-model BPVM is shown in Fig.1. This part of the meta-model is based on the quality model proposed in [17] and completed by the context awareness. The information about the impact of a non-functional requirement (NFR) on every fragment is considered as a quality attribute for this fragment. In this section, we present the part of the meta-model of BPVM without consideration of the context. In the following section, we present the contextualization of BPVM including the context issues and the non-functional perspective.

In our approach, the quality of the business process is expressed through the quality of its components, i.e. the BPFs. As shown in Fig.1, the quality of a BPF is formalized by the use of the links between the classes *Business process fragment* and respectively the classes *Quality attribute* and *Satisfaction link*. According to the meta-model, this relationship express the relationship between the non-functional perspective and the functional perspective.

The Concept of Non-functional Business Goals. The quality attributes are used as selection criteria to choose the variant of BPF the most suited in a given context. NFR goals introduced in the meta-model models the goals which are of qualitative nature. They include additional quality properties such as the *accuracy* (e.g. "lack of evaluation errors of a loan request"), the *safety* (e.g. "privacy of personal data") and the *performance/time* (e.g. "fast handling of a loan request"). We establish the satisfaction links (++, +?-,-) between the NFR and the BPFs. NFRs are decomposed in quality sub-goals [15], [16]. The non-functional goals are related to the functional goals by the satisfaction links.

Quality Features and Attributes. In order to guide business analysts in the determination of quality factors which are associated to a business process, we propose a set of quality features and attributes that are relevant to BPM. We consider that the quality of a process is determined according to the quality of the associated BPFs. We base our reflection on the works proposed in the literature [17], [18], [19] particularly on the standard ISO 9126 [18]. We have adapted the quality attributes defined by this standard for the software quality to the quality of business processes. We consider six quality features; each of them is composed of a set of attributes. Note that the considered quality features and attributes can be relevant for some BPFs and not relevant for others. Table 1 shows all the quality features as well as the attributes which correspond to every feature. Every attribute have metrics which are measurable indicators. The metrics are specified according to the business domain and the business process. For example, the attribute *Efficiency* can be measured by the metric *Achievement time*. Also, the attribute *User satisfaction* can be measured by the indicator *Average number of the users complaints per month*. We detail in what follows these features and these attributes by providing the definitions which highlight their adequacy to BPM.

Table 1. Quality features and attributes for BPM

Quality features	Quality attributes	Explanation
Functional capacity	Accuracy	Indicates the capacity of a BPF to provide results having the necessary precision degree.
	Security	Refers to the capacity of a BPF to protect the data from unauthorized accesses
	Suitability	Concerns the adequacy to the objectives defined by the actor.
Reliability	Reliability	Refers to the capacity of a BPF to maintain a specific level of performance in given conditions.
Ease of use	Learnability	It is the capacity of a fragment of process to allow the actors its learning.
	Understandability	Refers to the capacity of a BPF to allow the actors to understand how to use it in given conditions
Efficiency	Time efficiency	It is the capacity of a BPF to be supplied one time of answer and treatment suited in given conditions.
	Resource efficiency	It is the capacity of a fragment of process to use resources suited in precise conditions (in terms of number and type of resources)
	Efficiency with regard to the goals	It is the capacity of a fragment of process to allow the actors to reach goals in a given situation.
Safety	Safety	It is the capacity of a BPF to be implemented in acceptable levels of damage risk regarding people, processes, etc.
Actor satisfaction	Actor satisfaction	It is the capacity of a BPF to satisfy the actors in a given context.

3 The Contextualization of Business Process Models

This section deals with the contextualization of the business process models. At a first time, we base our reflection on the business process variability model introduced in this paper. The contextualization of a business process model (obtained by the instantiation of BPVM) consists in informing all its conditions of applicability of the BPFs. This fact requires to represent the context characteristics and the contextual conditions. We propose two kinds of contextualization: *the functional contextualization* and the *non-functional contextualization*.

3.1 Functional Contextualization

It consists in expressing the contextual conditions related to BPFs and to the roles and in representing the impact of the context on the way of executing these BPFs and of choosing the appropriate BPFs and roles at run-time. To every BPF, we associate a contextual condition allowing to specify the conditions under which the execution of a BPF is possible. Contextual conditions are formalized by the use of the class contextual situation. For example, in the business process loan handling, these contextual conditions can refer to the time pressure, the experience or the availability of an actor, etc. So, a BPF can be accomplished only if the associated context is actual.

3.2 Non-functional Contextualization

In some situations, the context has an impact on the contribution value of the variants in the satisfaction of a quality goal, i.e. according to the context, and according to the desired quality purposes, it is better to select an alternative rather than another one. The non-functional contextualization consists in adding the contextual conditions to the quality attributes. In the example of business process of Reservation and purchase of tickets, the registration can be done according to three manners: by internet, by the use of a self-service border, or at the counter. The context knowledge considered in this example is of temporal nature: the period during which the reservation is made.

4 Business Process Adaptation

The adaptation has for objective to determine the way a process is configured by taking into account adaptation factors i.e. the context, the quality requirements and the roles responsible for the achievement of the business process. The resultant business process model is so determined according to these factors. The context is taken into account to determine the executability of a BPF. The context is also taken into account during the choice of an alternative of execution of a BPF. The context has an impact on the quality of the process, i.e. an impact on the contribution value of the alternatives to the satisfaction of the quality goals. Thus, according to the context, and according to the desired quality goals, it is better to select an alternative rather than another one. As well, the roles, the actors and the associated contexts (example: availability of the actors) can also determine the executability of a BPF. We distinguish two categories of business process adaptation: the adaptation at the build-time and the adaptation at run-time.

Build-Time Adaptation. We indicate by build-time adaptation the adaptive configuration made before the execution of the process. The approach consists in configuring the business process model before its exploitation to divert models adapted to given contexts and to required quality requirements. So, several models diverted from an initial model are determined from the design phase. At the run-time, the instantiation will be based on a single model among the derived models. The determination of the most adequate model is made in two stages:

- To determine the models which can be used among the derived models. This is made by comparing the current context to contexts associated to the various predefined variants.

- If the selection of several model variants is possible, the system proposes a classification of the model variants basing on the desired quality criteria. The actor responsible for the process can so choose a model variant among those proposed.

Run-Time Adaptation. We mean by run-time adaptation the adaptive configuration which consists in configuring process models during its execution. This fact consists in insuring a controlled instantiation of the business process model, on one hand, by the actor to whom we offer many possible choices for every variation point of the process and that can choose in a dynamic way the fragment which suits him best, and on the other hand, by the system which, according to the current context, the desired characteristics of quality and to the roles of the actors, proposes the variants the most suited to the situation. The adaptation strategies will be detailed in our future works.

5 Related Work

Numerous business process modeling approaches that deal with the adaptation and the variability were proposed, but they are insufficient. In [1], the authors introduce a configurable reference modeling language. This approach as well as [22] proposes to indicate some artifacts of the process model as configurable items; from a single process model, a personalized model can be derived by selecting an alternative for each configurable element. The approach of Korherr integrates goals and variability and represents business process models according to a set of perspectives, i.e. the business process context perspective, the behavioral perspective, the functional perspective, the organizational perspective, and the informational perspective [2]. [23], [24] support variability and express it by organizing business processes in families and manage process variability and common parts in the family in order to enable the reuse and the adaptability of process models.

Even though the above mentioned approaches support variability, only [2] and [23] provide a variability model. Furthermore, business modeling approaches that deal with variability take into consideration variability related to the functions [22], [2], to the business process paths [1], to the strategies to achieve goals [23] and to the activities [2]. In [22], in addition to the functions, the actors' roles are also considered as part of variability. We assume that the variability related to the organizational perspective, i.e. the actors' roles is an important issue and needs to be represented.

Furthermore, even if some approaches such as [2] represent business process models according to various perspectives, none of them support the non-functional perspective. We believe that this perspective have to be captured.

The main contribution of this paper is that it provides an approach that allows to represent a business process model according to many perspectives. What's more, we propose to model variability in both the functional and the organizational perspectives. Hence, variations are defined with respect to the way of achievement of business process fragments and to the actors' roles.

6 Conclusion

We have presented in this paper a multi-perspective approach for business process modeling integrating the variability. Our approach is based on business process meta-model named BPVM. The proposed meta-model offers several possible representations of the same family of processes by considering the requirements of change. It includes five modeling perspectives which are: (i) the intentional perspective allowing to express the business goals that the business process has to satisfy, (ii) the functional perspective allowing to represent a business process in terms of BPFs and to capture the variability in the way of realizing the goals associated to the BPFs, (iii) the organizational perspective allowing to represent the organizational resources, including the actors, the roles, and to express the variability related to the roles, (iv) the non-functional perspective representing the quality requirements related to the business process and (v) the perspective of the non-organizational resources representing the data and the business objects used, produced or consumed by the business process. We have also discussed issues related to the contextualization of business process models using BPVM as well as issues related to the adaptability. In future works we will develop in detail a context management approach allowing to model and to manage context. We will also develop adequate strategies and tools for the adaptation of business process models.

References

1. Rosemann, M., van der Aalst, W.M.P.: A Configurable Reference Modelling Language. Information Systems 32(1), 1–23 (2007)
2. Korherr, B.: Business Process Modelling. Languages, Goals and variabilities. VDM VERLAG (2013)
3. Debauche, B., Mégard, P.: BPM Business Process Management: Pilotage métier de l'entreprise, Paris, Hermès (2004)
4. Gestion des processus métier orientée objectifs - Création de processus métier adaptables aux environnements imprévisibles. Rapport Tibco (2007), `http://www.fr.tibco.com/multimedia/goal-oriented-bpm_tcm18-6974.pdf`
5. Prat, N.: Goal formalisation and classification for requirements engineering. In: Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ 1997, Barcelona, Spain (1997)
6. Prat, N.: Aide au processus de BPR: une approche fondée sur le raisonnement par cas. In: Colloque de l''Association Information et Management (AIM), pp. 212–227. Cergy-Pontoise, France (1999)
7. Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., Rolland, C.: A Strategy Driven Business Process Modelling Approach. Special issue of the Business Process BP management Journal on Goal-oriented Business Process Modeling, Emerald 11(6) (2005)
8. Rolland, C., Prakash, N.: On the Adequate Modeling of Business Process Families. In: The 8th Workshop on Business Process Modeling, Development, and Support (BPMDS 2007), held in conjunction with CAiSE 2007 (2007)

9. Rolland, C., Prakash, N., Benjamen, A.: A Multi-Model View of Process Modelling. Requirements Engineering Journal (1999)
10. Workflow Management Coalition Specification: Workflow Management Coalition Terminology and Glossary. Technical Report Number WFMC-TC-1011 (1999)
11. Business Process Model and Notation (BPMN), Version 2.0, Document number formal/2011-01-03 (2011), `http://www.omg.org/spec/BPMN/2.0`
12. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Mulyar, N.: Wokflow control-flow patterns. BPM Center Report BPM-06-22. BPM center.org (2006)
13. Electronic resource, `http://www.tibcosoftware-events.com/mailing/nouvelle_generation/`
14. Pohl, K., Böckle, G., van der Linden, F.: Software Product Line Engineering Foundations, Principles and Techniques. Springer, Heidelberg (2005)
15. Chung, L., Nixon, B.A., Yu, E.: Dealing with Change: An approach Using Non-functional Requirements. In: Proceedings of the Second Internation Symposium on Requirements Engineering. Springer Verlag London Limited, York (1996); Requirements Engineering Journal
16. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Boston (2000)
17. Rohleder, C.: Representation of software variability requirements. PhD Thesis. Université paris1-Panthéon-Sorbonne (2009)
18. ISO/IEC-9126-1: Software engineering - product quality: Quality model, Geneva. International Organization for Standardization, Switzerland (2001)
19. Heravizadeh, M., Mendling, J., Rosemann, M.: Dimensions of Business Processes Quality (QoBP). In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008 Workshops. LNBIP, vol. 17, pp. 80–91. Springer, Heidelberg (2009)
20. Ploesser, K., Recker, J., Rosemann, M.: Challenges in the context-aware management of business processes: A multiple case study. In: 19th European Conference on Information Systems (ECIS), Helsinki, Finland (2011)
21. Curran, T., Keller, G.: SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Upper Saddle River (1997)
22. La Rosa, M., Dumas, M., ter Hofstede, A.H.M., Mendling, J.: Configurable multi-perspective business process models. Information Systems 36(2) (2011)
23. Rolland, C., Nurcan, S.: Business Process Lines to deal with the Variability. In: Hawaii International Conference on System Sciences - System Sciences (HICSS), 43rd Hawaii International Conference, États-Unis (2010)
24. Schnieders, A., Puhlmann, F.: Variability Mechanisms in E-Business Process Families. In: Proceedings of the International Conference on Business Information Systems (2006)

A Model-Driven Approach for Accountability in Business Processes

Anderson Santana de Oliveira[1], Anis Charfi[2], Benjamin Schmeling[3], and Gabriel Serme[1]

[1] SAP AG, 805 Avenue Maurice Donat, Mougins, France
[2] SAP AG, Bleichstr. 8, Darmstadt, Germany
[3] UBL Informationssysteme, Carl-Ulrich-Strae 4, Neu-Isenburg, Germany

Abstract. Accountability provides the necessary assurance to different stakeholders (customers, auditors, regulators) about the correct execution of the obligations concerning compliance requirements. Modeling accountability in a business process is an important problem, as SOA is the generally accepted standard for IT systems. This requires the orchestration of several non-functional concerns across services (such as authentication, authorization, logging, among others) to attest the correct operation of control activities. In this paper, we show how a model-driven framework for non-functional concerns can integrate accountability in business processes. Using the NFComp modeling framework, we define and compose a set of non-functional concerns that securely assert that subjects have fulfilled their responsibilities, towards realizing accountability. The approach allows the reuse of the composed accountability concerns in different processes.

1 Introduction

Accountability is the security property stating that subjects (persons, organizations, etc.) must be liable by the execution of their authority towards the fulfillment of obligations, for example in the context of a business process. Such obligations are expressed in terms of the objectives of a given system (or service) generally defined in contracts or service level agreements (SLA's)[15]. Achieving accountability is extremely important to show conformance with certain regulations, such as PCI DSS[1], HIPAA[2], BASEL III[3], EU's data protection[4], SOX [5]; to resolve commercial disputes among the stakeholders involved in business transactions; or to support auditing activities.

In outsourced scenarios, accountability increases the level of trust among business partners as it assures stakeholders that risks related to the compliance with

[1] `pcisecuritystandards.org/security_standards/index.php`
[2] `hhs.gov/ocr/privacy/`
[3] `bis.org/bcbs/basel3.htm`
[4] `ec.europa.eu/justice/data-protection/index_en.htm`
[5] `sec.gov/spotlight/sarbanes-oxley.htm`

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 184–199, 2014.

respect to contractual or regulatory constraints have been mitigated. In order to realize it, a number of controls are put in place to protect sensitive data, to track data flows, and to provide evidence about obligation execution.

The successful adoption of process-oriented approaches in the last decade, in particular in the context of service-oriented architectures, demands a comprehensive approach to integrate the support for accountability in business process modeling and in business process execution. Process-oriented languages make explicit the composition logic of activities, allowing the encapsulation of atomic behavior in platform independent services with standard interfaces (e.g., WSDL) and message exchange formats (e.g., SOAP). In this context several process modeling languages have appeared, such as BPMN2 [7].

Existing frameworks for business process modelling have no explicitly support for capturing accountability requirements. We can mention for instance TIBCO[6], IBM solutions[7], and SAP Netweaver Business Process Composer[8]. The problem we address in this paper is the translation of these requirements into concrete actions (i.e., primitive or composite behavior) that ensure a compliant implementation of a business processes.

In this paper we present a model-driven approach to accountability in business processes. We focus on the technical realization through the composition of individual actions, regarding distinct security properties. The actions and their compositions are then combined with the base process model using the NFComp modeling framework [11]. This generic framework covers the life cycle of non-functional concerns from modeling to execution. It provides a rich toolset consisting of different modeling editors and code generators. NFComp supports defining non-functional concerns (NFC's) and the actions realizing them in business processes. In this paper we focus on the composition of various security concerns such as identity management, secure logging, privacy, and auditing already at the modeling phase to achieve accountability.

The contributions of this paper are many-fold. First, we define accountability and discuss its related security requirements for their technical realization in automated business processes. Second, we apply the NFComp methodology throughout our case study to define accountability actions and support their composition with each other, with other non-functional properties, and also with the core business process.

The remainder of the paper is structured as follows. Section 2 presents the motivation for accountability in business processes. Section 3 presents the related security requirements introduced by the accountability property. Section 4 presents the NFComp framework. Section 5 explains how we use that framework to support accountability and illustrate its usage through a running example. Section 6 discusses related works and Section 7 concludes the paper.

[6] http://developer.tibco.com/business_studio/default.jsp

[7] http://www-03.ibm.com/software/products/us/en/business-process-manager-family

[8] http://help.sap.com/saphelp_nw73/helpdata/en/1e/b250a408ff44c28ea7f1a53b5e7791/content.htm

2 Motivation

In order to clarify the concept of accountability under the perspective of business process modeling, we provide the following definition:

Definition 1. *Accountability for business processes produces trustworthy evidence about the compliance state of its composing activities, by associating responsibilities for the execution of contractual and regulatory obligations to the distinct roles of the entities (persons or systems) authorized to execute tasks in the process.*

In order to illustrate the need for accountability in business process consider a loan origination business process that evaluates the credit situation of a bank customer and proposes a loan solution to her. Part of the process is outsourced to an external Credit Bureau, who evaluates the credit risk associated to a given person. Such agencies exist in several countries such as Schufa[9] in Germany and Equifax[10] in the USA. In order to propose the most interesting loan packages to its clients, the bank also checks whether they are eligible to benefit from special low interest rates provided by the government. The BPMN diagram for such business process is depicted in Figure 1.

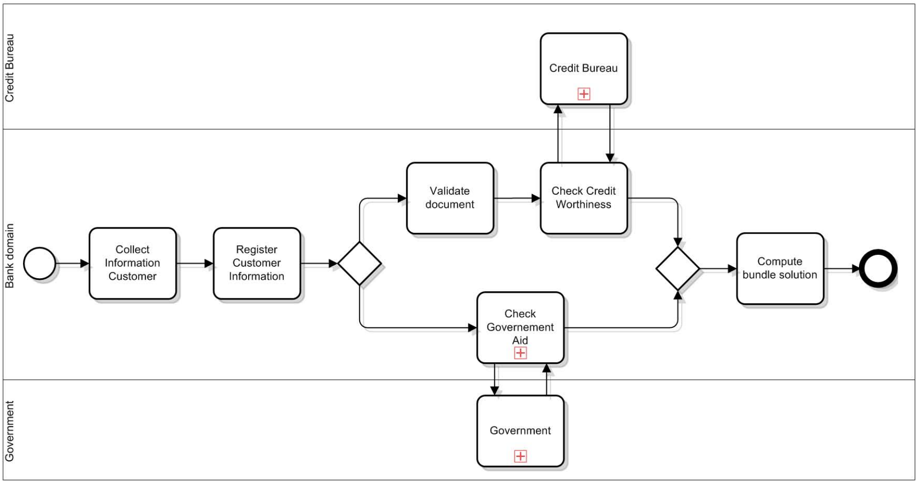


Fig. 1. BPMN diagram for the loan origination process

This scenario raises several questions related to accountability. First, all actors are liable to the correct handling of personal information with regards to data protection regulations. Second, the contract between the bank and the

[9] http://www.schufa.de/
[10] http://www.equifax.com/

credit bureau imposes non-repudiation restrictions concerning the credit report requests about the bank customers. Third, assuming that threats to the business process can be carried out either by external or internal actors, e.g. by creating fraudulent requests, it is important to collect all evidences about the process instance that produces a given loan package, in order to be able to solve potential disputes.

Accountability is rightfully discussed along a multitude of dimensions, including regulatory (legal) and financial obligations and compliance, as well as technical security criteria. In this paper, we focus especially on the modeling of the technical security requirements related to accountability.

An example evidence is an acknowledgment message signed with the digital certificate of the bank's clerk responsible for validating the data provided by a given bank customer. Therefore, an accountability framework for business processes must enable the collection of evidences ensuring the achievement of the contractual obligations to which the legal entities are liable, taking into consideration the regulatory framework of the business process in question. The accountability framework must also provide traceability for the collected data. This allows stakeholders such as auditors to assert whether tasks were executed completely and correctly by the responsible actors.

3 Requirements for Accountability

In order to achieve accountability in a business process, we identified a number of requirements we consider necessary to ensure compliance. In the following, we present these requirements.

Authentication: The identity of all parties involved in a business process shall be uniquely determined.

Authorization: Only authorized parties are allowed to execute specific activities in the business process.

Confidentiality: Data exchanged between process partners can only be read by authorized parties. This requirement is combined with integrity. Cryptographic protection between communicating parties can prevent data from being read by unauthorized subjects.

Integrity: The trustworthiness of accountability data needs to be ensured by the business process, in order to prevent malicious agents from tampering with the evidences provided. The collected evidence must also be protected from unauthorized access.

Non-repudiation: A fair non-repudiation protocol [4] to provide evidence to all participant entities in the business process about the execution of the business process activities. In fair protocols, no participant can obtain more evidences than another for message reception or origination. The generated proofs will later be used in audits and in dispute resolution.

Liability: It is always possible to clearly identify the individual responsible for executing a given activity in an instance of a business process. Additionally, the business process must implement controls to enforce personal data usage constraints, e.g., using the data exclusively for the purpose they were collected for. The maximum retention periods must also be observed. Therefore, assuming responsibility for correct data handling must also be demonstrated.

Traceability: All messages produced or consumed by the process have to be logged. The satisfaction of this requirement provides trusted audit trails, allowing identifying the actors involved in a given instance of the business process.

The business process presented in Section 2 illustrates all these constraints. Since the bank hires the services of an external agency, ensuring that personal data is correctly handled and that the issued reports correspond to authentic requests is important for dispute resolution.

For instance, the bank employees need to be personally identified in order to check whether separation of duty constraints have been correctly enforced. This will also rely on authorization, e.g., only managers can approve loan requests. Here we assume that some attribute and role based access control model is in place.

During an audit, we shall be able to verify the system traces to investigate whether some user performed actions using incompatible roles, e.g., the same person could be able to create a loan request, and to approve it at a later moment using the manager role. Second, it is also necessary to have "machine" authentication in cross-domain interactions.

In our use case, non-repudiation is necessary when the interaction among processes takes place across administrative domains, as for instance in the interactions between the bank and the credit bureau.

With respect to liability in the loan origination use case, a number of requirements concerning retention of personally identifiable information can be identified. In this context we refer to the new European Data Protection Directive [11] that establishes a "data minimization" principle for handling personal data, which limits the accumulation of data by requiring its deletion - "Personal data must be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed. Member States shall lay down appropriate safeguards for personal data stored for longer periods for historical, statistical or scientific use". For instance, service providers have the obligation to determine a maximum retention period for the data, which must be deleted when the expiration date is reached. The bank and the credit bureau handle several personally identifiable data in their business processes, which we assume to be stored in some internal database for each participant. Ideally, an automated sub-process, which we call here private data removal process, should take care of the deletion of that data in due time. This process must run on behalf of the data controller which is the bank in this case. The process must provide evidence to the audit

[11] `http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf`

trail that the data has been properly removed, such that an auditor can certify that the participant organizations are compliant with respect to this legal requirement.

These requirements are not so simple to handle at design time. Security experts and business analysts often do not talk the same technical "language". Misunderstandings may lead to incorrect (insecure) implementations of accountability. Visual modeling can clarify the actions related to accountability. The NFComp approach, explained in the next section, considers the implementation of each non-functional concern in isolation, but also, the interplays among them through composition primitives.

4 Composing Non-functional Concerns for Services

In this section, we provide an overview of the NFComp approach [11,12] for composing non-functional concerns in business processes. NFComp is a model-driven approach for concern composition that superposes non-functional concerns (NFCs), such as security, performance, or transactional behavior, into fine-grained non-functional actions (NFAs). NFAs contain abstract non-functional behavior for encryption, message signing, logging, etc, and can be composed in different ways.

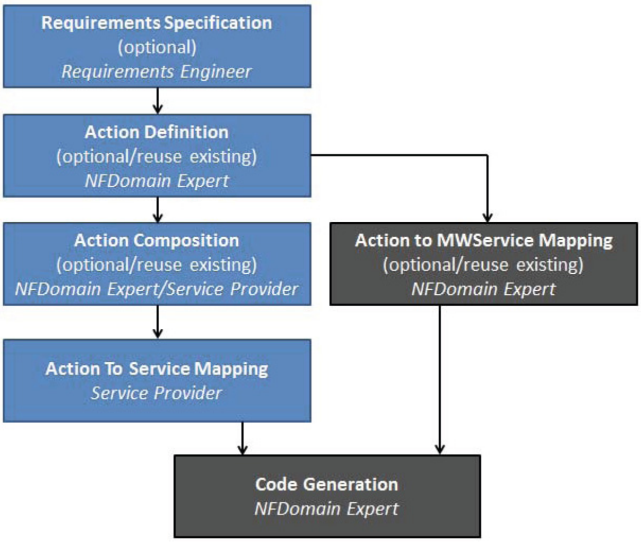


Fig. 2. The different phases of NFComp

Firstly, actions can be defined, which support the realization of non-functional requirements. These actions can be composed with each other in composite non-functional activityies. Then, the atomic actions and the non-functional activities can be composed with web services or with the business processes underlying composite web services. When several actions are mapped to a single web service, an execution order at that point needs to be specified (vertical composition). In case the non-functional actions have a stateful nature (i.e., they have a process nature e.g., transactions or secure conversations) the execution order needs to be specified. The composition in our approach is modeled graphically, making use of BPMN for vertical and horizontal composition complemented by its own mapping notation.

The modeling of NFCs is only one feature of the NFComp approach and is referred to as specification of NFCs. Another feature is the realization of the model at runtime which is achieved by code generation and an appropriate architecture which respects web service properties such as platform independence and distribution.

The whole approach can be separated into different logical phases involving different user roles shown in Figure 2. In the following, we will elaborate on the phases.

In the first phase, **Requirements Specification**, requirements engineers model the non-functional requirements which are relevant for the (composite) service. The requirements are turned into non-functional attributes categorized by NFCs, e.g. *liability*, *transparency*, or *assurance*. In this phase all roles and responsibilities regarding private information are identified, and specific privacy controls are determined depending on the use case.

In the second phase, **Action Definition**, NFAs such as *encrypt*, *decrypt*, *log* etc. are modeled by the experts of the respective non-functional domain, e.g. security or reliability experts. In this phase properties and interdependencies between actions can be specified in order to ease the task of vertical composition. This is necessary because in this vertical composition where different actions from different concerns will be composed, knowledge from different domains is required.

In the next phase, the **Action Composition**, the horizontal and vertical composition can be specified in terms of BPMN 2.0. The actions from the action definition can be used to model processes with control and data flow using BPMN gateways. In terms of accountability, we use action composition to group the actions and generate evidence about the correct personal data handling.

In the **Action to Service Mapping** phase individual actions or the results of vertical compositions (called non-functional activity) can be mapped to web services. The input is the WSDL of the service or in case of composite web services the BPMN process. Actions and activities can be mapped to service subjects such as the service, its operations and process subjects such as tasks and events. The mapping is represented by a typed association between subject and action/activity. The types define when a certain action should be executed, e.g. before a task is executed, before a message is sent, after a message is received or after a task is executed. At this point, accountability supporting services are involved, e.g. a data removal process.

The next two phases contribute to the realization of the composition model. In the **Action to Middleware Service Mapping** phase NFAs are mapped to concrete software components implementing the behavior reflected by the actions. These components could be web services, e.g. a Logging Web Service to implement a “log” action. The NFComp framework provides a default architecture to enforce the composition model which is based on proxies. The idea is that all messages target by or sent by web services are passing this proxy which is then deciding — based on the model — which actions have to be executed and in which order. NFComp makes use of the open-source implementation Apache Synapse ESB which is used to provide web service proxies and middleware services. NFComp provides a model to model transformation for Synapse which transforms, during the **Code Generation** phase, the NFComp model into the XML-based configuration language of the ESB. Finally, NFComp strongly supports the separation of concerns even at runtime by modularizing functional concerns into web services, non-functional concerns into middleware services and the composition logic into proxies.

5 Modeling Accountability in Business Processes

In this section, we apply the NFComp methodology to our loan origination case study. The accountability property is represented by a set of non-functional concerns. It assures the correct execution of obligations rather than providing consumable business functionality for processes or services. The different non-functional concerns such as authentication, authorization, and logging which should be composed correctly and follow well-defined process semantics. For example, in the case of authorization, context creation should be executed before authentication is made by the user, and finally the context should always be terminated. To make those non-functional processes visible, reusable, and composable, it is necessary to model them using the same means as those used for business processes.

On one hand, such an approach avoids the need for learning new languages. On the other hand, the BPMN standard is also suitable for non-functional concerns. Reusability and composability can only be achieved if the specification of non-functional concerns is completely separated from the business processes. Once understood and modeled, NFComp allows deploying accountability concerns indepently of any concrete business processes and not only for the specific case of the loan origination process. Furthermore, it is extremely important that the accountability specifications are directly enforced at runtime. The manual implementation of the specification is error-prone as errors can be introduced while performing such a translation. Automated code generation helps to assure that the specification is realized and can be enforced at runtime. All these aspects are supported by the NFComp methodology and for these reasons we use it to specify and enforce accountability as we elaborate in the following.

The first phase in NFComp is the *Requirements Specification* which helps to structure the actual requirements in form of non-functional attributes. The requirements have already been identified in Section 3 and can directly be used as non-functional attributes.

5.1 Defining Actions

After specifying what goals to achieve, the **Action Definition** phase allows to define how or by which actions the requirements can be realized. Figure 3 depicts the actions and their properties as well as their interdependencies modeled through labeled connections.

Example 1. Each action has four properties. Consider the *SignRequest* action:

1. The first property describes the impact of the action with respect to the message which is intercepted. *Add* means that some content (in this case a signature) is added to the message.
2. The second property defines the part of the message that is affected by the action. In this case the *header*.
3. The third property defines whether the action will be executed on incoming or outgoing messages, here, messages are signed by the subject executing a given process before the message is sent.
4. The fourth property relates the action to the non-functional attribute it supports, in this case, *Integrity*.

This action *SignRequest* has a corresponding reverse action namely *VerifySignature*, which checks the signature of a incoming message in a process. This dependency is indicated in Figure 3 with an interdependency connection labeled *inverse*. Furthermore, the figure shows some *precedes* and *requires* interdependencies between other actions. For example the execution of *Match* has to precede that of *Authenticate*. The *Match* action compares data subject's privacy preferences to the business process task being executed. The data subjects must have agreed to the specific processing in question.

5.2 Composition of Actions

In the **Action Composition** phase we have to decide in which order those actions are executed in case of a superimposition, that is, to define what happens if more than one of those actions is mapped to the same point in the process. We have six super-impositions which have been modeled by non-functional activities. Non-functional activities contain process (BPMN2) logic defining the execution order and control flow of NFAs. Figure 4 shows the activities that have been defined for accountability.

The *MatchUpdatePolicy* activity relates privacy preferences from the data subjects about whom the business process is collecting data. In this activity, the preferences are matched against the purpose and context of the current task. For instance, the data subject may state in her policy that she wishes to be notified whenever her data is used for commercial or research purposes.

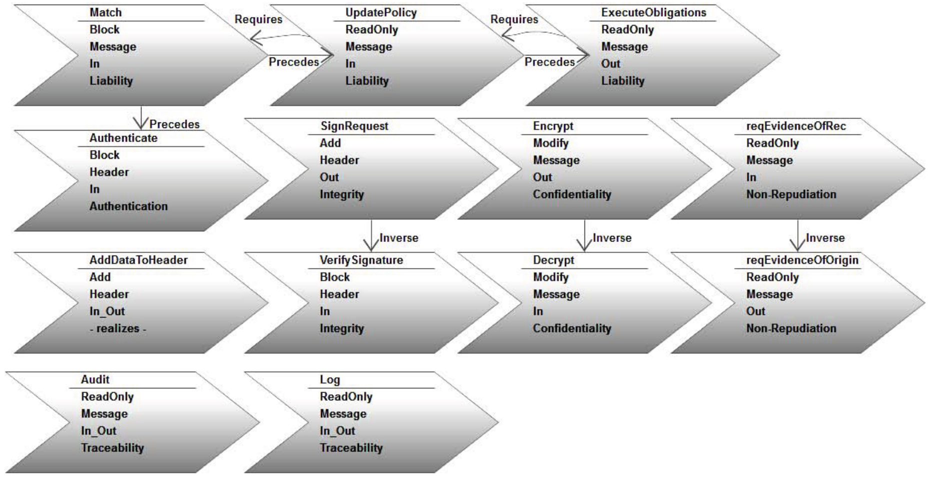


Fig. 3. Action definitions and their interdependencies

The result of the matching will update the obligations to the data processor (in our running example, the bank), who will be liable to its execution. This is automated by the *ExecuteObligationsLog* activity, which will make sure that all updated privacy preferences are executed correctly and timely. In order to provide evidence, when sending out notifications or when deleting personal data as the maximum retention period is reached, this action logs securely a proof that the obligation was carried out.

The *LogWithId* activity defines that the ID stored in *AuthenticateVerifyLog* should be added to the message header before it is logged. This activity is reused as a sub-activity in other activities. This will help to provide the necessary assurance to the business process.

The *SignEncNROLog* activity defines that a message should be signed before being encrypted. Then a request for the evidence of origin is sent which produces a token represented by the data item shown in the activity. However, this request is sent only if the condition *isNonRepudiation* is true. This allows to reuse the activity across different process tasks, e.g., when this activity is mapped to an internal task the *reqEvidenceOfOrigin* action is not needed and thus not executed. The log action is the last action to be executed which means that logged messages will always contain the signature and be encrypted. Such action composition contributes to multiple accountability requirements such as liability, transparency, and assurance.

The *DecryptVerifyNRRLog* activity firstly decrypts messages and verifies the signature thereafter in order to be compatible to the strategy used in *SignEncLog*. The *AuthenticateVerifyLog* activity defines that *Authenticate* will first check the identity of the message sender and then verify the signature. The data item ID is produced by the authenticate action which means that this information is stored for the lifetime of the ongoing instance of the loan origination business process.

The ID will be consumed by the LogWithId activity. This action composition is the counterpart of the *SignEncNROLog* activity.

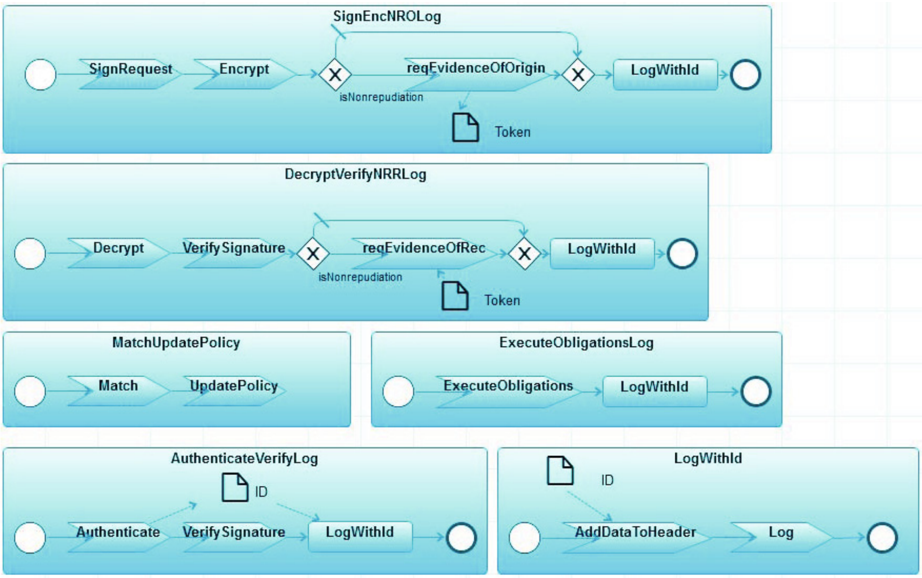


Fig. 4. Activity Editor: Vertical composition of accountability NFAs

5.3 Mapping Actions to Services

In the **Action to Service Mapping** phase NFAs and non-functional activities from the previous two phases are mapped to web services or BPMN processes. Hence, the BPMN specification of the loan origination process has been imported which can be seen in Figure 5. Additionally, the action definition and action composition model have been imported.

The process is started when a message is received by the *CollectCustomerInformation* task. The *AuthenticateVerifyLog* activity has been mapped to incoming messages shown by the incoming message symbol on the association line. This means that a message will only be accepted if it is correctly signed and if the identity of the process consumer is valid. Additionally, also the *AuthenticateVerifyLog* activity has been mapped to the same action for the same task.

In the next step of the process the service task *RegisterCustomerInformation* is started. It calls a web service provided by the bank's internal web application and initiates the *LogWihtId* activity because it is mapped for outgoing messages. By receiving the customer registration message an input form is rendered to bank employees in order to type in additional customer information. The web service will send the answer back to the process. Therefore the bank employee has to be authenticated. This is realized by the mapping of *Authenticate* to incoming messages of the *RegisterCustomerInformation* task.

The *ValidateDocument* task is similarly implemented. It also communicates with the web application and also requires authentication. The *CheckCreditWorthiness* task sends (resp. receives) messages to (resp. from) an external service of the Credit Bureau. Thus, confidential information has to be protected and non-repudiation should be guaranteed. Thus, *SignEncNROLog* and *DecryptVerifytNRRVLog* is mapped to outgoing respectively incoming messages. At the same time, the Credit Bureau becomes liable with respect to the privacy constraints in place, as there is evidence of the reception of personal data.

Finally, the *LogWithId* action is mapped to all tasks in order to enable tracing for all relevant messages exchanged between the process and its partners (the internal web app, the credit bureau, and the government agency). Since all these messages are tracked an audit can be initiated whenever necessary, attesting that only authorized users have had access to personal data.

Figure 5 also depicts the results of validating this mapping against the interdependencies that were defined in the action definition phase. The validation algorithm[12], which has been implemented using Prolog found one constraint violation. We have defined that the action *Match* has to be executed before the action *Authenticate*. However, in the mapping there is no ordering restriction between both actions because they are mapped separately using the same direction (incoming messages). For such a mapping no specific order can be assumed, e.g., the activities *MatchUpdatePolicy* and *AuthenticateVerifyLog* could be executed in different orders. The violated task is marked red as well as the violated process branch(es) and a message showing the actual error is shown when the mouse cursor is moved over the violated action or activity. However, this mapping problem could easily be fixed by merging the two violated activities to one and has only been introduced here to demonstrate the validation feature which is helpful, especially in case of complex mappings as required for accountability.

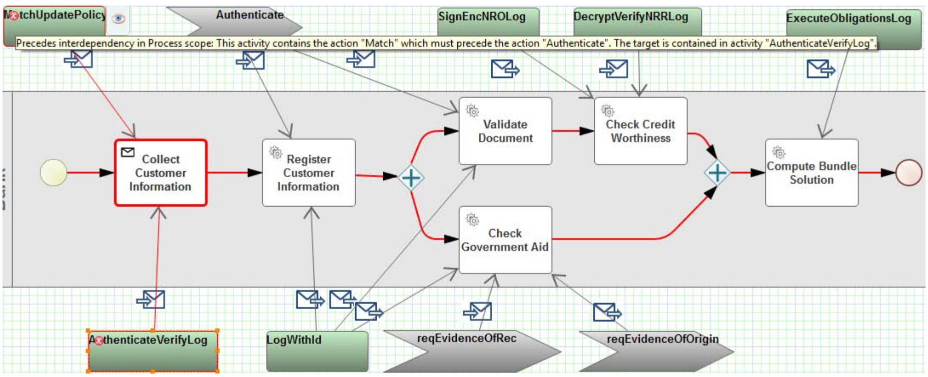


Fig. 5. Mapping Editor: Mapping of NFAs and Activities

[12] For more details on this refer to [10].

5.4 Final Mapping and Code Generation

In the **Action to Middleware Service Mapping** phase, we need to map the actions to realizing software components. As NFComp already provides some code generators for an ESB (Enterprise Service Bus) infrastructure where the ESB acts as a proxy in front of web services we make use of this infrastructure. In this architecture it makes sense to use middleware web services and components (so called mediators) in the ESB to implement the actions. The mapping of an action to a middleware web service can be done via the unique service and operation name or the unique mediator name in case of components inside the ESB. Additionally parameters for configuring those components can be specified as key-value-pairs for each action. For example, for the Encrypt action we specify amongst others the key length, encryption algorithm etc. However, NFComp assumes that those components already exist. Thus, we used the set of available ESB mediators (for encryption, signing authentication, *addToHeader* etc.) and complemented those by additional middleware web services: the *LiabilityWebService* with three operations *match*, *updatePolicy*, and *ExecuteObligations*; the *LoggingWebService*; the *NonrepudiationWebService* with *requestEvidenceOfRec*; and *requestEvidenceOfOrigin* operations.

After the middleware mapping in the **Code Generation** phase the code generator can be used to generate the XML configuration for the ESB. This configuration defines which mediators or external web services have to be invoked, in which order, and for which service. This information is extracted by the code generator from the NFComp model. This approach allows using the NFComp model as configuration input for the ESB which makes sure that the modeled composition is also enforced at runtime.

In outsourced scenarios, an agreement must be set among the parties to adopt similar control activities, in cross-organizational business processes. Our approach promotes uniform and transparent implementations of the security controls and provides means to map them to the underlying implementations at each organization taking part in the business process.

6 Related Work

We have organized the related work in two parts. First, we present models and solutions related to SOA modeling of non-functional concerns in orchestration of services. Then, we describe the different approaches of modeling accountability in workflows and business processes.

With respect to solutions for modeling SOA-based systems and non-functional concerns in service orchestration we can mention [13] who introduced the SecMoSC (Security for Model-oriented Service Composition) methodology for incorporating security requirements into service compositions. The work uses NF-Attributes and NF-Actions to describe the security requirements and the design decisions, algorithms, data structures and configurations implementing those requirements. An NF-Attribute can be composite or primitive and it is an abstraction for the enforcement mechanism realizing the attribute. The work also provides an enforcement mechanism for their model by generating a generic WS-BPEL with service annotations and then a platform-specific WS-BPEL and a platform-specific security configuration (which is an extension to WS-Policy). However, it provides only static sequential ordering of actions whereas with NF-Comp we can define dynamic orderings with data flow which we made use of in our work.

The work presented in [1] specifies security properties at the service composition level using APEL, which is a model-based orchestration tool. APEL supports the orchestration view with its own meta-model. This view can be extended by other views providing meta-models integrated with the orchestration meta-model via meta-links. The authors developed a security meta-model with integrity, authentication and confidentiality as the main concepts. Those requirements can be mapped directly to activities in the orchestration model and an Axis2/WSS4J configuration is generated. This approach of Chollet and Lalanda is comparable to NFComp's requirements specification, mapping and code generation phases. The advantage of NFComp is the support for action definition and composition consisting of superimposing (accountability) actions.

In [14], the authors model accountability as assigning responsibility and ownership to activities in business processes as separate concerns. It is hard to separate business from accountability requirements in their proposal. Our approach is more general in the sense we collect evidence about the (correct) execution of the processes without interfering in the core definition of the activities.

Several works consider accountability as ensuring quality of service and SLA fulfillment in diverse contexts. For example, the work described in [5] highlights requirements in SOA systems towards accountability: monitoring of services and identification of faults, inspection of internal state of services, and reconfiguration of services and service processes. The framework described detects, diagnoses and defuses service deficiencies while focusing on the quality of service and SLAs for service mashups. These are important aspects of accountability, but it is unclear how these approaches can provide security assurance.

In the same vein, Yao et al.[16] proposed a model to provide accountability services in the scope of service-oriented architectures. According to the authors, accountability is primarily supported by logging, monitoring and auditing, and dispute resolution mechanisms. The paper claims that these mechanisms are incorporated to business process descriptions in BPEL, but only logging is shown. The same level of protection against repudiation is provided, since PKI is used, in a similar manner as in our work. There is no support for clearly separating concerns. In [17], Zou *and al.* address accountability in the context of business processes. They provide a service ontology framework to ease capturing non-functional requirements during business process modeling. We consider this approach to be complementary to our work as it helps to understand accountability concepts, however, without support for non-functional concern composition.

The forensic web services [3,2] approach considers obligation fulfillment in multi-party web services. It supports evidence preservation for the interactions among services to solve disputes, based on a layered framework ensuring trust in the process. A similar approach is shown in [9]. The main distinction from these works to ours, is that we not only consider machine to machine trust, but we focus on privacy management, in the same vein as in [6,8], but here in the context of business processes. Another distinguishing feature of our approach is the design and modeling support with visual notations and tools..

7 Conclusions

In this paper we presented an approach to model and realize accountability in business processes and illustrated it through a case study. Accountability is shown as a thorough use case for non-functional concern composition, since multiple actions have to be composed to ensure the correct execution of obligations. We used the NFComp framework to define accountability related actions, action composition, and the mapping of actions to particular business process steps. The approach allows having an overview of all non-functional concerns applied to a business process as well as analyzing their composition and its implication during process execution. NFComp also supports the realization of accountability by providing code generation support for the configuration of an ESB, which enforces the modeled requirements at runtime. For the realization only the non-functional actions had to be implemented by providing middleware web services for logging, encryption, signature, authentication, etc. As future works we will extend the application of this approach to a number of industrial use cases in order to obtain insights about any difficulties users would have in implementing accountability controls using our approach.

Acknowledgments. The work presented in this paper was partly performed in the context of the Software-Cluster project InDiNet. It was partially funded by the German Federal Ministry of Education and Research under grant no. 01IC10S04A and by the European Commission Seventh Framework Programme Research Project A4Cloud under grant number 317550. The authors assume responsibility for the content.

References

1. Chollet, S., Lalanda, P.: Security specification at process level. In: IEEE International Conference on Services Computing, SCC 2008, vol. 1, pp. 165–172 (July 2008)
2. Gunestas, M., Wijesekera, D., Elkhodary, A.: An evidence generation model for web services. In: IEEE International Conference on System of Systems Engineering, SoSE 2009, May 30-June 3, pp. 1–6 (2009)
3. Gunestas, M., Wijesekera, D., Singhal, A.: Forensic web services. In: Ray, I., Shenoi, S. (eds.) Advances in Digital Forensics IV. IFIP International Federation for Information Processing, vol. 285, pp. 163–176. Springer, Boston (2008)
4. Kremer, S., Markowitch, O., Zhou, J.: An intensive survey of fair non-repudiation protocols. Computer Communications 25(17), 1606–1621 (2002)
5. Lin, K.J., Panahi, M., Zhang, Y., Zhang, J., Chang, S.H.: Building accountability middleware to support dependable soa. IEEE Internet Computing 13, 16–25 (2009)

6. Casassa Mont, M., Pearson, S., Bramhall, P.: Towards accountable management of privacy and identity information. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 146–161. Springer, Heidelberg (2003)
7. OMG: Business Process Model and Notation (BPMN) 2.0 (January 2011)
8. Pearson, S., Charlesworth, A.: Accountability as a way forward for privacy protection in the cloud. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 131–144. Springer, Heidelberg (2009)
9. Robinson, P., Cook, N., Shrivastava, S.: Implementing fair non-repudiable interactions with web services. In: EDOC Enterprise Computing Conference, 2005 Ninth IEEE International, pp. 195–206 (September 2005)
10. Schmeling, B., Charfi, A., Martin, M., Mezini, M.: Towards Conflict-Free Composition of Non-functional Concerns. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 80–94. Springer, Heidelberg (2012)
11. Schmeling, B., Charfi, A., Mezini, M.: Composing Non-Functional Concerns in Composite Web Services. In: IEEE International Conference on Web Services (ICWS 2011). IEEE Computer Society, Washington DC (2011)
12. Schmeling, B., Charfi, A., Thome, R., Mezini, M.: Composing Non-Functional Concerns in Web Services. In: The 9th European Conference on Web Services (ECOWS 2011). IEEE Computer Society, Lugano (2011)
13. Souza, A.R.R., et al.: 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)
14. Tseng, M.M., Su, C.J., Qinhai, M.: Accountability centered approach to business process reengineering. In: HICSS, vol. (4), pp. 345–354 (1998)
15. Weitzner, D.J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., Sussman, G.J.: Information accountability. Commun. ACM 51, 82–87 (2008)
16. Yao, J., Chen, S., Wang, C., Levy, D., Zic, J.: Accountability as a service for the cloud. In: IEEE International Conference on Services Computing (SCC) 2010, pp. 81–88 (July 2010)
17. Zou, J., De Vaney, C., Wang, Y.: A meta-modeling framework to support accountability in business process modeling. In: Yang, J., Ginige, A., Mayr, H.C., Kutsche, R.-D. (eds.) UNISCON 2009. LNBIP, vol. 20, pp. 539–550. Springer, Heidelberg (2009)

Modeling and Verifying Security Policies in Business Processes

Mattia Salnitri[1], Fabiano Dalpiaz[2], and Paolo Giorgini[1]

[1] University of Trento, Italy
{mattia.salnitri,paolo.giorgini}@unitn.it
[2] Utrecht University, The Netherlands
f.dalpiaz@uu.nl

Abstract. Modern information systems are large-sized and comprise multiple heterogeneous and autonomous components. Autonomy enables decentralization, but it also implies that components providers are free to change, retire, or introduce new components. This is a threat to security, and calls for a continuous verification process to ensure compliance with security policies. Existing verification frameworks either have limited expressiveness—thereby inhibiting the specification of real-world requirements—, or rely on formal languages that are hardly employable for modeling and verifying large systems. In this paper, we overcome the limitations of existing approaches by proposing a framework that enables: (1) specifying information systems in SecBPMN, a security-oriented extension of BPMN; (2) expressing security policies through SecBPMN-Q, a query language for representing security policies; and (3) verifying SecBPMN-Q against SecBPMN specifications via an implemented query engine. We report on the applicability of our approach via a case study about air traffic management.

Keywords: Information systems, Security policies, BPMN, Compliance.

1 Introduction

Information systems are becoming increasingly large, complex, and decentralized. Air Traffic Management (ATM) systems, smart grids, and smart cities are not simple monolithic systems, but rather they consist of a high number of autonomous, heterogeneous, and mutually interdependent components. These systems require new design techniques, in order to prevent crashes with effects on both organizations and society [33].

These systems manage a large amount of private and confidential information; as such, their design shall ensure information assurance and security both in technical terms and from an organizational perspective [26]. Business process models are an adequate abstraction to do so, for they express an information system in terms of the interactions between humans, organizations, and technical systems.

Several modeling languages have been proposed that extend BPMN (Business Process Modelling and Notation) [23]—the de-facto standard notation for representing business processes—with security annotations that individual BPMN elements shall comply with [26,34]. For example, Rodriguez et al. [26] extended BPMN with a predefined set of security annotations (e.g., attack/harm detection and privacy) that constrain the execution of the annotated tasks of the business process.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 200–214, 2014.

However, the security annotations in [26] constrain individual elements in a business process, and do not allow expressing security policies that specify the admissible behavior of the whole business process. Some extensions of BPMN employ a predefined set of policies—BPMN patterns which specify the behavior of the business process—but they do not allow the definition of custom security policies [6, 20]. This is the case, for instance, of SecureBPMN [6], which introduces security elements for expressing, e.g., separation and binding of duties.

Furthermore, business process modelers need to verify whether a process model complies with the specified policies. This verification is required not only when designing the process, but also whenever the participating components do change after its deployment. For example, in an ATM system, a failure in the pilot to control tower communication component requires a quick reconfiguration of the system, which shall be checked for compliance with the security policies.

Existing approaches for compliance checking are inadequate: some focus on general-purpose policies and do not provide support to security policies [3, 11, 17, 28], while others use a too limited set of policies, mainly concerning access control [19, 20, 26].

In this paper, to overcome the limitations of existing approaches, we propose a framework for modeling and verifying the compliance of a business process model with a set of security policies. To do so, we take BPMN-Query (BPMN-Q) [3] as our baseline; BPMN-Q is a query language that enables expressing and verifying generic queries over a BPMN model. We extend BPMN-Q with a number of annotations for expressing security policies. We make the following contributions:

1. The SecBPMN language, which extends BPMN with security annotations.
2. SecBPMN-Q, an extension of BPMN-Q for specifying security policies as queries.
3. An implemented framework for modeling in SecBPMN, specifying security policies in SecBPMN-Q, and running SecBPMN-Q queries against SecBPMN models.
4. We evaluate the applicability of our approach on a case study.

The rest of the paper is structured as follows. Section 2 describes our baseline. Sections 3 and 4 introduce SecBPMN and SecBPMN-Q, respectively. Section 5 discusses related work and shows how we applied our approach on a case study. Finally, Section 6 presents our conclusions and outlines future directions.

2 Baseline

In this section we briefly introduce the baseline of our research: the BPMN-Q language for querying business process models, and the RMIAS security reference model.

While BPMN is an effective means for expressing the interactions among the components in a complex system, it does not offer the possibility to verify whether certain critical properties of the model do hold. For example, when modeling the landing procedure in air traffic management, one cannot automatically verify with BPMN that pilots do confirm the landing trajectory of the plane.

Visual analysis of BPMN models works for small scenarios, but it is ineffective when many models exist, or when they are as large as hundreds of nodes. Moreover, when safety and security properties are concerned, relying on an informal analysis is not an option, due to the harmful effects of adopting a model that violates them. BPMN-Q is a diagrammatic query language which partially overcomes this limitation, by expressing properties concerning business process models through graphical queries that can be checked against a model [4]. These queries can be seen as patterns that a given BPMN model should comply with. BPMN-Q introduces a set of relations that are functional to define the queries, i.e. the concepts of path, negative path, negative flow as well as an extension of the "data object" element that enable characterizing the state of the object.

Figure 1 shows an example of a BPMN-Q query (taken from our SWIM ATM case study[1]). The query enables checking whether the flight plan (Reference Business Trajectory or simply RBT) is approved and if the landing documents are checked at least once. The query will match against all business processes where the first activity "Plane RBT generation service" generates the data object "RBT [Proposed]" (between brackets is indicated the state of the data object) and other two activities are executed: (i) "Control Tower communication service" generates the data object "RBT [Accepted]"; (ii) any activity ("@Y") reads the data object "Landing documents [Approved]".

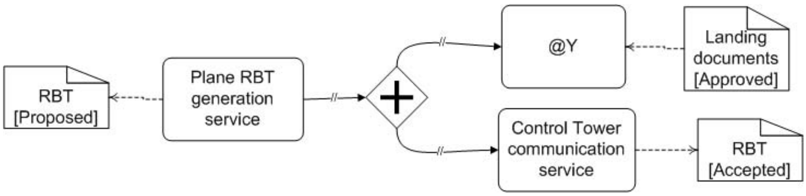


Fig. 1. Exmple of a BPMN-Q query

BPMN-Q enables expressing generic properties over BPMN elements, but does not support the specification of security properties. To overcome this limitation, the languages proposed in this paper extend BPMN and BPMN-Q with primitives for specifying and querying security properties, which we define based on a state-of-the-art reference model for information security. A prominent family of reference models extends the Confidentiality Integrity Availability (CIA) triad [24]. However, their adequacy has been questioned for they characterize a too limited set of properties of a system [25]. Later, more complete reference models were proposed, for example McCumber's cube [18], which analyzes system security from three different perspectives: information states, critical information characteristics and security measures. The Business Model for Information Security (BMIS) [1] focuses on business environments, and consists of four interconnected elements: organization design and strategy element, people element, process element and technology element. In our work, we choose the Reference Model on Information Assurance and Security (RMIAS) [7], as it is the result of an analysis and classification of security aspects proposed by the most known reference models on information assurance and security. As far as our knowledge goes, it proposes the most comprehensive set of security aspects, that are listed in Table 1.

[1] The System Wide Information Management (SWIM) [2] is a next-generation communication system which enables the secure interchange of information among ATM decision makers. We use it to evaluate the languages in Section 5.

Table 1. Security aspects in RMIAS [7]

Name	Definition
Accountability	An ability of a system to hold users responsible for their actions (e.g. misuse of information).
Auditability	An ability of a system to conduct persistent, non-by passable monitoring of all actions performed by humans or machines within the system.
Authenticity	An ability of a system to verify identity and establish trust in a third party and in information it provides.
Availability	A system should ensure that all system's components are available and operational when they are required by authorised users.
Confidentiality	A system should ensure that only authorised users access information.
Integrity	A system should ensure completeness, accuracy and absence of unauthorised modifications in all its components.
Non-Repudiation	The ability of a system to prove (with legal validity) occurrence/non-occurrence of an event or participation/non-participation of a party in an event.
Privacy	A system should obey privacy legislation and it should enable individuals to control, where feasible, their personal information (user-involvement).

3 SecBPMN: A Modeling Language for Secure Business Processes

We extend BPMN with security annotations covering each of the security aspects in the RMIAS reference model (see Table 1). Every annotation has a graphical syntax and is linked with an existing element of a BPMN model: an activity, a data object, or message flow. Moreover, annotations have attributes that security designers can use to specify detailed information on the security mechanisms[2] that enforce the policy. All attributes are optional but one: the BPMN element linked with the annotation.

Specifically, our language extends the subset of BPMN—that is supported by BPMN-Q—for specifying orchestrations, which enables expressing interactions among information system components: activities, gateways and data objects. Each security annotation is formalized in terms of one or more predicates, one for every type of BPMN element the annotation can be linked with.

Our graphical syntax was carefully designed according with Moody's guidelines for increasing the usability and comprehensibility of modeling languages [21]. The annotations share three common visual variables: they all have an orange fill color, a solid texture, and a circular shape; they differ in the icon in the middle of the circle. Every security annotation has a visual distance of three from non-security annotations, and a visual distance of one from other security annotations. We decide to use icons instead of abstract symbols because icons are easy to remember and faster to recognize [21]. Leitner et al. [14–16] conducted empirical studies to propose guidelines for representing a set of security aspects. We did not apply such suggestions because they conflict with the recommendation by the security experts that helped us define the security annotations

[2] They define the low level (software and hardware) functions that implement the controls imposed by the policy [31].

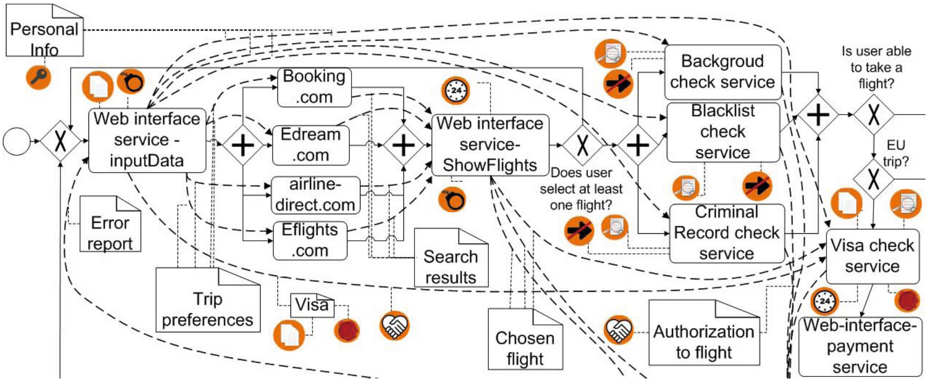


Fig. 2. Example of a SecBPMN business process model

and, moreover, the set of security aspects Leitner et al. took into account covers only partially the security aspects proposed in RMIAS.

Figure 2 shows an example of BPMN extended with security annotations, which shows part of a service composition, offered by different service providers, intended to enable flight tickets booking. There, the security annotations specify the security aspects that the implemented services will comply with. The annotations are defined in Table 2 and explained below.

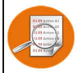
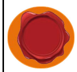
Accountability. It applies only to activities, and expresses the need of monitoring a set of users when executing the activity. Thus, there is only one corresponding predicate named AccountabilityAct. It has three parameters: a is the activity whose execution has to satisfy the security aspect corresponding to this type of annotation, enfBy is a set of security mechanisms used to enforce accountability for the activity, monitored is the set of users which are monitored.

If the activity is executed by a user that is not in monitored, the security aspect is satisfied without using the enforcement mechanism. This situation would typically occur with trusted users that do not need be monitored. Security designers can specify they keyword ALL in monitored, to indicate that all users are held for their actions.

Consider, for example, the predicate AccountabilityAct("Web interface service - inputData", {RBAC}, {customer}), which details one of the accountability security annotations in Figure 2. The first attribute contains the activity linked with the security annotation, the second one indicates that RBAC (Role-Based Access Control) [9] will be used to enforce accountability, while the third attribute specifies that only customers have to be monitored while executing that activity.

Auditability. It introduces three variants of security annotation, which are used to express that it should be possible to verify different aspects of the business process: (i) AuditabilityAct indicates that it should be possible to keep track of all the actions performed by the executor of the activity a when trying to execute that activity; (ii) AuditabilityDO indicates that it should be possible to keep track of all the actions that manage (e.g. write, read, store) the data object do; (iii) AuditabilityMF indicates that it should be

Table 2. Security annotations of SecBPMN: predicates and graphical syntax

AccountabilityAct (a: Activity, enfBy: {SecMechanisms}, monitored: {Users})	
AuditabilityAct (a: Activity, enfBy: {SecMechanisms}, frequency: Time) AuditabilityDO (do: DataObject, enfBy: {SecMechanisms}, frequency: Time) AuditabilityMF (mf: MessageFlow, enfBy: {SecMechanisms}, frequency: Time)	
AuthenticityAct (a: Activity, enfBy: {SecMechanisms}, ident: Bool, auth: Bool, trustValue: Float) AuthenticityDO (do: DataObject, enfBy: {SecMechanisms})	
AvailabilityAct (a: Activity, enfBy: {SecMechanisms}, level: Float) AvailabilityDO (do: DataObject, enfBy: {SecMechanisms}, authUsers: {Users}, level: Float) AvailabilityMF (mf: MessageFlow, enfBy: {SecMechanisms}, level: Float)	
ConfidentialityDO (do: DataObject, enfBy: {SecMechanisms}, readers: {Users}, writers: {Users}) ConfidentialityMF (mf: MessageFlow, enfBy: {SecMechanisms}, readers: {Users}, writers: {Users})	
IntegrityAct (a: Activity, enfBy: {SecMechanisms}, personnel: Bool, hardware: Bool, software: Bool) IntegrityDO (do: DataObject, enfBy: {SecMechanisms}) IntegrityMF (mf: MessageFlow, enfBy: {SecMechanisms})	
NonRepudAct (a: Activity, enfBy: {SecMechanisms}, execution: Bool) NonRepudMF (mf: MessageFlow, enfBy: {SecMechanisms}, execution: Bool)	
PrivacyAct (a: Activity, enfBy: {SecMechanisms}, sensitiveInfo: {Info}) PrivacyDO (do: DataObject, enfBy: {SecMechanisms}, sensitiveInfo: {Info})	

possible to keep track of all the actions executed to handle the communication (send and receive actions) within the message flow mf.

The three predicates share two parameters: enfBy to express a specific set of security mechanisms to be used, and frequency to specify how frequently the security checks are performed. If frequency is set to zero, the continuous verification is required.

For instance, consider the predicate AuditabilityAct("Background check service", {}, 10d), which formalizes one of the auditability annotations in Figure 2. It applies to activity Background check service, it does not require a specific technology for checking auditability, and it requires audits to be performed every 10 days.

Authenticity. It comes in two versions, depending on which BPMN elements the annotation applies to. AuthenticityAct imposes that identity and/or authenticity of users of activity a are verified. The attribute enfBy is the set of security mechanisms to be used while trustValue is the minimum level of trust [12] the executor of activity a must have. If attribute ident is true, anonymous users should not take part in the execution of the activity, while if auth is set to true, the identity of users should be verified. AuthenticityDo indicates that it should be possible to prove the data object do is genuine, i.e. it should be possible to prove that it was not modified by unauthorized parties, and to prove the identity of the entity who generated and/or modified it.

For example, consider the predicate AuthenticityDO (“Visa”, {TLS, X.509}), which formalizes an authenticity security annotation in Figure 2. The predicate specifies that the integrity of Visa data object should be guaranteed using TLS (Transport Security Layer) and X.509 security mechanisms.

Availability. It applies to three BPMN elements, hence we defined three different versions: (i) AvailabilityAct specifies that the activity a should be executed every time it's specified in the business process; (ii) AvailabilityDO specifies that the data object do should be available when required by the authorized users specified in authUsers attribute; (iii) AvaliabilityMF specifies that it is always possible to communicate through the message flow mf.

The predicates share two parameters: enfBy, described above, and level, i.e., the minimum time percentage that the resource (i.e., activity, data object or message flow, depending on the variant of availability annotation) should be available. In AvailabilityDO, security designers can specify that all users are authorized to request the data object, simply specifying the keyword ALL in the attribute authUsers.

For instance, the predicate AvailabilityAct(“Web interface service - ShowFlights”, { SAVE }, 99.5) specifies that Web interface service - ShowFlights has to process at least 99.5% of the total requests, using the SAVE (Source Address Validity Enforcement) protocol to prevent denial of service attacks.

Confidentiality. It has two variants: ConfidentialityDO which specifies the data object do can be accessed only by authorized users, and ConfidentialityMF which specifies that only authorized users can use (i.e send or receive) the message flow mf. Both predicates share three parameters: enfBy, already described; readers i.e. the set of users that are authorized to read the data object (or receive from the message flow); writers i.e. the set of users that are authorized to write the data object do (or send through the message flow). The attributes readers and writers allow the usage of the keyword ALL to specify that all the users are authorized.

For instance, consider the predicate ConfidentialityMF (mf(“Web interface service - inputData”,“Visa check service”), {TLS, RBAC}, {controlAuthority, VisaOwner}, {VisaOwner}), which details one of the confidentiality annotations in Figure 2. It specifies that only the users controlAuthority and VisaOwner can receive from the message flow between Web interface service - inputData and Visa check service, and only VisaOwner can send data objects through that channel. This security annotation must be enforced using both TLS and RBAC security mechanisms.

Integrity. It comes in three variants: (i) IntegrityAct specifies that the functionalities of activity a should be protected from intentional corruption. Attributes personnel, hardware and software determine if respectively the personnel, hardware or software, involved in the execution of the a, are protected from intentional corruption [10]; (ii) IntegrityDO specifies that the data object do should be protected from intentional corruption [10]; (iii) IntegrityMF specifies that every message exchanged through mf should be protected from intentional corruption. All the predicates share the attribute enfBy.

For instance, the predicate IntegrityAct("Visa check service", {} , false, true, true) specifies one of the integrity annotations in Figure 2. It indicates that software and hardware used to execute Visa check service will be protected from intentional corruption, e.g. unauthorized modifications of the software or hardware robbery.

Non-repudiation. It is defined as: NonRepudiationAct and NonRepudiationMF. The former indicates that the execution (or non-execution) of activity a should be provable, while the latter specifies that the usage (or non-usage) of the message flow mf should be verifiable. Both the predicates have in common two attributes: enfBy, already described before, and execution, which specifies if it's required a proof of execution (when it is set to true) or non-execution (when it is set to false) of activity a or message flow mf, in the latter case is required a proof of usage of the communication channel.

For example, the predicate NonRepudiationAct("Blacklist check service", {}, false) defines one of the non-repudiation annotations in Figure 2. It specifies that it should be possible to prove that Blacklist check service has never been executed. There are no constraints on the security mechanisms that have to be implemented because the parameter is an empty set.

Privacy. It has two variants: (i) privacyACT specifies that activity a should be compliant with privacy legislation, and it should let users to control their own data; (ii) privacyDO is similar to the former one, but is targeted to a specific data object, specified in do. Both predicates share two parameters: enfBy, already described before, and sensitiveInfo, i.e. the set of sensitive information that must be protected.

For example, consider the predicate PrivacyDO("Personal Info", {} , {name, surname, dateOfBirth, passportID}), which refines one of the privacy annotations in Figure 2. It specifies that the owner of name, surname, date of birth and passport id information contained in the data object Personal Info should be able to delete the data and, if the information are published, they should be anonymized as required by law, e.g. publish only partial information.

4 Modeling and Verifying Security Policies

We propose the SecBPMN-Q language, an extension of BPMN-Q query language, to model security policies using the security annotations in Table 2. Our query language permits to graphically model security policies, which is a useful feature to support the communication of the specified policies with other stakeholders.

Consider, for example, a textual policy such as "*The visa document must be authenticated and it must be sent through a secure channel which assures the information will not be sniffed or modified by third parties, implementing TLS and X.509 security mechanisms*". Figure 3 models this policy in SecBPMN-Q. Beside the two generic tasks and the path, that are elements of BPMN-Q, the BPMN-Q query is enriched with a message flow (represented as a dashed arrow) which exchange a data object called "Visa". When executed, this query will match any message flow between two activities which exchange the "Visa" data object. The confidentiality annotation linked to the message flow requires the communication channel to assure the data object will be received only by "Visa owners" and "Control authority". Moreover, the "Visa" data object has to be protected by unauthorized modifications, implementing the "MD5" security mechanism specified by integrity annotation, and its originality has to be provable using "TLS" and "X.509" security mechanisms, specified in the authenticity annotation. Some optional attributes are not specified, meaning that the security designer is imposing fewer constraints on the specific security mechanism. For example, in Figure 3, enfBy and writers parameters of ConfidentialityMF are not defined (see the underscore placeholder), hence the predicate will be satisfied, regardless the security mechanisms implemented or the set of users authorized to send data objects through the channel.

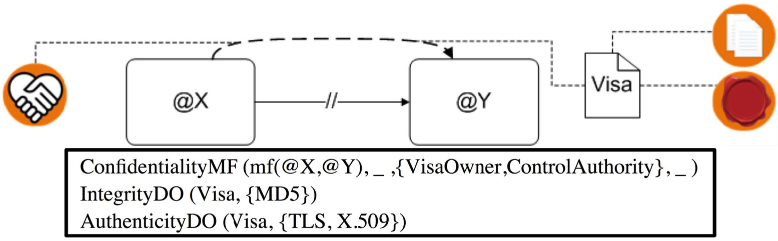


Fig. 3. Example of a security policy and predicates expressed with SecBPMN-Q

In order to verify if the security policies modeled with SecBPMN-Q are satisfied by a SecBPMN-Q business process, we extended the BPMN-Q engine with the implementation of Algorithm 1. The algorithm takes in input a SecBPMN business process and a SecBPMN-Q security policy, and it verifies if there exists a path in the business process that satisfies the security policy. For each path, the algorithm verifies if the security annotations of the business process are of the same type of those in security policy and if they are linked to the same SecBPMN element. If so, the security annotations of the security policy are verified against the security annotations in the business process.

Algorithm 1. Compliance check of a security policy

```
COMPLIANCE(SecBPMN bp, SecBPMN-Q secPolicy)
paths ← FINDPATH(bp, secPolicy)
if paths = ∅ then
    return false
for each path ∈ paths do
    satisfied ← true
    for each secAnnPolicy ∈ GETSECURITYANNOTATIONS(secPolicy) do
        for each secAnnPath ∈ GETSECURITYANNOTATIONS(path) do
            if secAnnPolicy.type = secAnnPath.type then
                if CHECKTARGET(secAnnPath, secAnnPolicy) then
                    satisfied ← SATISFIES(secAnnPath, secAnnPolicy) ∧ satisfied
    if satisfied then
        return true
return false
```

Algorithm 2. Pseudo-code of function “satisfies”

```
SATISFIES(SecurityAnnotation SecAnnPath, SecurityAnnotation SecAnnPolicy)
if (secAnnPolicy.enfBy ⊈ secAnnPath.enfBy) then
    return false
switch (SecAnnPolicy.type)
    case AccountabilityAct :
        return (SecAnnPolicy.monitored ⊆ SecAnnPath.monitored)
    case AuditabilityAct ∨ AuditabilityDO ∨ AuditabilityMF :
        return (SecAnnPolicy.frequency ≤ SecAnnPath.frequency)
    case AuthenticityAct :
        return ((SecAnnPolicy.ident → SecAnnPath.ident)∧
                (SecAnnPolicy.auth → SecAnnPath.auth)∧
                (SecAnnPolicy.trustValue ≤ SecAnnPath.trustValue))
    case AvailabilityAct ∨ AvailabilityDO ∨ AvailabilityMF :
        return (SecAnnPolicy.value ≤ SecAnnPath.value)
    case ConfidentialityDO ∨ ConfidentialityMF :
        return ((SecAnnPolicy.readers ⊆ SecAnnPath.readers)∧
                (SecAnnPolicy.writers ⊆ SecAnnPath.writers))
    case IntegrityAct :
        return ((SecAnnPolicy.personnel → SecAnnPath.personnel)∧
                (SecAnnPolicy.hardware → SecAnnPath.hardware)∧
                (SecAnnPolicy.software → SecAnnPath.software))
    case NonRepudiationAct ∨ NonRepudiationMF :
        return (SecAnnPolicy.exeution ↔ SecAnnPath.exeution)
    case privacyAct ∨ privacyMF :
        return (SecAnnPolicy.sensitiveInfo ⊆ SecAnnPath.sensitiveInfo)
```

A security annotation of a business process satisfies a security annotation of a security policy if all the attributes of the former are more restrictive of the attributes of the latter. The function satisfies, Algorithm 2, checks this property. As first step, Algorithm 2 checks if the security mechanisms specified in the security annotation of the policy are all specified in the security annotation of the business process; if not, it returns false, meaning that the security policy specifies at least a security mechanism that is not implemented in the business process. After that, depending on the type of annotation, the algorithm checks:

- *accountability*, if the monitored users specified in the policy are all monitored by the business process;
- *auditability*, if the frequency of the checks specified in the policy is less or equal than the one specified in the business process;
- *authenticity*, if ident attribute is true in the security annotation specified in the security policy (every user has to be identified) then the same attribute specified in the business process is true. The same criteria is used also for auth. The trustValue defined in the security annotation of the security policy has to be less or equal that the value defined in the one specified in the business process, since the security aspects corespondent to the security annotation is satisfied when the trust required is less than the trust offered by the executor of the activity;

- *availability*, if the value specified in the security annotation is less than the value specified in the business process;
- *confidentiality*, if the set of authorized users specified in the security annotation of the security policy is a subset of the authorized users specified in the business process;
- *integrity*, if the personnel attribute (for what concerns IntegrityAct) is true in the security policy then is true in the business process; the same criteria applies for hardware and software. The other two variants of integrity do not need special criteria because they are characterized only be the attribute enfBy, that is already checked in the first two lines of the algorithm;
- *non-repudiation*, if the attribute execution is the same in both the security annotations, since it specifies two different constraints;
- *privacy*, if the set of sensitive information specified in the security policy is included in the set specified in the business process.

The SecBPMN engine fixes a number of bugged functionalities and comes with a manual which explains the installation of all the required software packages [3].

When a SecBPMN-Q security policy is checked, the interface of SecBPMN-Q engine presents to the users all the business processes in the repository that have at least one path (graphically highlighted in the business process) that satisfies the security policy specified. Figure 4 shows the result of the SecBPMN-Q query shown in Figure 3 with the SecBPMN-Q process shown in Figure 2. The path highlighted in Figure 4 satisfies the security policy in Figure 3: (i) the first activity of the path, i.e. "Web interface service - inputData", is linked with a message flow to the last activity of the path, i.e., "Visa check service"; (ii) the message flow is used to exchange the data object "Visa" and it assures confidentiality of the transferred data object; (iii) integrity and authenticity of the "Visa" data object are preserved. Assuming the predicates that details the security annotations of the security policy are less restrictive of the predicates of the business process, the path, and consequently the business process, satisfies the security policy.

5 Discussion

The literature offers a number of graphical modeling languages for expressing security aspects in business process models. These languages support a predefined set of security policies that a designer can use; examples are SecureBPMN [6], other extensions of BPMN e.g. [19, 26, 29, 34], or UML profiles, such as UMLsec [13]. The advantage of these languages is that they are easy to learn and to use [21], thereby requiring a moderately low effort for security designers to specify a secure business process. The price to pay for using these modeling languages is in their limited expressiveness: these graphical modeling languages do not permit to define custom security policies, thereby preventing the creation of domain-specific variants. As such, existing verification engines (e.g., [28, 30, 32, 34, 35]) that enable the automated verification of these models do also support a fixed set of hard-coded security policies.

[3] The extended version of the engine and the manual can be found at `http://www.secbpmn.disi.unitn.it`.

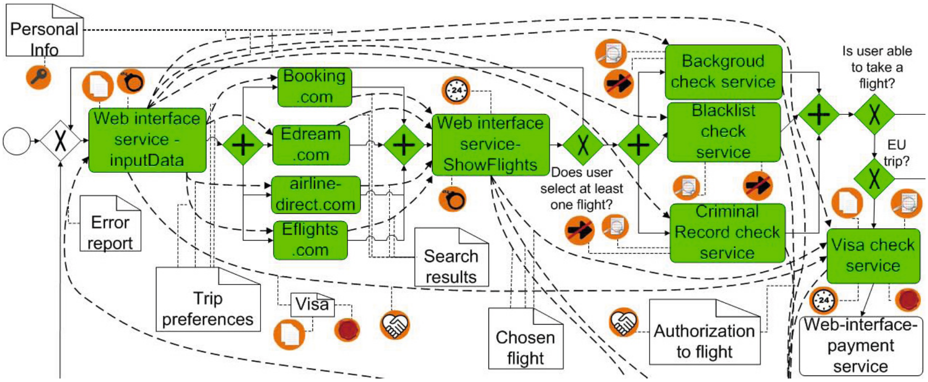


Fig. 4. Result of the query based on SecBPMN-Q policy in Figure 3 against the SecBPMN model in Figure 2

Other graphical languages have been proposed to check the compliance of a process with a query. For example, the Business Process – Query Language (BP-QL) [5] permits to create graphical queries that are checked against processes modeled using the Web Services Business Process Execution Language (WSBPEL) [22]. BP-QL permits to search paths that are compliant with patterns that are defined through the query language proposed; however, BPMN is not used as a basis for the query language. Similarly, the Business Process Query Language (BPQL) [8] permits to graphically define both queries and business process models using the same language. Unfortunately, BPQL is not based on BPMN, hence the learning process is likely to be less quick than that with by BPMN-Q, for this latter language is based on the well-known standard.

Other approaches build on formal languages (e.g., first-order logic, temporal logic, etc.). This trend of work is characterized by high expressiveness. For example, Liu et al. [17] propose a language and a framework which statically verifies a business process against a formally expressed regulatory requirements, while Rushby [27] proposes a language and a framework which checks if the code of a software system diverges from specified behaviors (i.e., policies). The main drawback of these approaches is their low usability, for they require a substantial effort for formalizing business processes and security policies. Moreover, they can hardly be used at runtime, for their verification requires more time, due to the use of a more expressive logics.

We applied SecBPMN and SecBPMN-Q on a case study about a SWIM [2] ATM system, that is part of the Aniketos[4] European project. The ATM system consists of a large number of autonomous and heterogeneous components, which interact with each other to enable air traffic management operations: pilots, airports personnel, national airspace managers, meteo services, radars, etc. In such a complex information system, ensuring security is critical, for security leaks may result in severe consequences on safety and confidentiality. Experts from the Aniketos project analyzed the security requirement document provided with the case study, and identified 27 active entities and

[4] www.aniketos.eu

more that 60 security policies. We studied these security policies and modeled them using SecBPMN-Q. After that, we examined the documentation about the case study and we defined four business processes, each containing a number of nodes (activities and gateways) between 28 and 58.

Being based on BPMN, we did not experience particular issues in modeling the processes described in the documentation using SecBPMN. SecBPMN-Q enabled us model all the security policies elicited by the experts but two cases:

- security policies concerning redundancy, which we could represent only at a high-level of abstraction, without managing to express if the fallback activities have to be performed by the same or a different executor. This limitation was inherited by BPMN-Q, which does not support BPMN swim-lanes and pools. To overcome of this limitation, we plan to introduce swim-lanes and pool elements in an extension of SecBPMN/SecBPMN-Q;
- security policies about the non-delegation of an activity, i.e., preventing that third parties execute one activity or parts of it. Even in this case, our future work includes introducing additional elements to the meta-model to support this type of policy.

This preliminary evaluation shows the applicability of the proposed languages for modeling security policies and security-annotated business processes in a non-trivial scenario. However, more extensive evaluation is required for our approach, including experimentation on other domains, assessing the scalability of our algorithms, and checking how well novices and business process experts learn our languages.

6 Conclusions and Future Work

This paper has introduced SecBPMN and SecBPMN-Q, a modeling language for modeling security-annotated business processes, and a query language for expressing security policies, respectively. Our languages are supported by a toolset that supports both modeling and the execution of queries. Moreover, we have applied our approach on a complex information system for air traffic management.

Our approach overcomes the deficiencies of existing approaches, which either suffer from a limited expressiveness—being graphical languages that support only a predefined set of security annotations—or from limited scalability—begin reliant on expressive temporal logics, thereby inhibiting efficient runtime verification.

Our approach is not yet complete, and opens the doors to several future directions: (1) apply the languages to different domains; (2) assess the learnability and usability of our languages; (3) create a catalogue of patterns representing common security policies; (4) determine the scalability of our automated reasoning mechanisms; (5) include our engine in a workflow system to support security policy-compliant runtime reconfiguration.

Acknowledgement. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no 257930 (Aniketos) .

References

1. An introduction to the business model for information security. Technical report, ISACA (2009), http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/An-Introduction-to-the-Business-Model-for-Information-Security.aspx
2. Federal Aviation Administration. SWIM ATM case study, http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/swim/ (last visited March 2014)
3. Awad, A.: BPMN-Q: A language to query business processes. In: EMISA, St. Goar, Germany. LNI, vol. P-119, pp. 115–128. GI (2007)
4. Awad, A.: A compliance management framework for business process models. PhD thesis (2010)
5. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes with BP-QL. Information Systems 33(6), 477–507 (2008)
6. Brucker, A.D., Hang, I., Lückemeyer, G., Ruparel, R.: SecureBPMN: Modeling and Enforcing Access Control Requirements in Business Processes. In: Proc. of SACMAT 2012, pp. 123–126 (2012)
7. Cherdantseva, Y., Hilton, J.: A reference model of information assurance and security. In: Eighth International Conference on ARES, pp. 546–555 (September 2013)
8. Deutch, D., Milo, T.: Querying structural and behavioral properties of business processes. In: Arenas, M., Schwartzbach, M.I. (eds.) DBPL 2007. LNCS, vol. 4797, pp. 169–185. Springer, Heidelberg (2007)
9. Ferraiolo, D.F., Cugini, J.A., Richard Kuhn, D.R.: Role-based access control (rbac): Features and motivations (1995)
10. Firesmith, D.: Specifying reusable security requirements. JOT 3(1), 61–75 (2004)
11. Ghose, A., Koliadis, G.: Auditing business process compliance. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 169–180. Springer, Heidelberg (2007)
12. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
13. Jürjens, J.: Umlsec: Extending uml for secure systems development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
14. Leitner, M., Miller, M., Rinderle-Ma, S.: An analysis and evaluation of security aspects in the business process model and notation. In: Proc. of ARES, pp. 262–267 (2013)
15. Leitner, M., Rinderle-Ma, S.: A systematic review on security in process-aware information systems - constitution, challenges, and future directions. Inf. Softw. Technol. 56(3), 273–293 (2014)
16. Leitner, M., Schefer-Wenzl, S., Rinderle-Ma, S., Strembeck, M.: An experimental study on the design and modeling of security concepts in business processes. In: Proc. of PoEM, pp. 236–250 (2013)
17. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Syst. J. 46(2), 335–361 (2007)
18. McCumber, J.: Information systems security: A comprehensive model. In: Proceeding of the 14th National Computer Security Conference, NIST Baltimore, MD (1991)
19. Menzel, M., Thomas, I., Meinel, C.: Security requirements specification in serviceoriented business process management. In: Proc of ARES 2009, pp. 41–48 (2009)

20. Monakova, G., Brucker, A.D., Schaad, A.: Security and safety of assets in business processes. In: Applied Computing, vol. 27, pp. 1667–1673. ACM, USA (2012)
21. Moody, D.: The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. 35, 756–779 (2009)
22. OASIS. Web Services Business Process Execution Language, `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html` (April 2007)
23. O.: BPMN 2.0, `http://www.omg.org/spec/BPMN/2.0` (January 2011)
24. Parker, D.: Our excessively simplistic information security model and how to fix it. ISSA Journal, 12–21 (2010)
25. Parker, D.B.: Fighting computer crime - a new framework for protecting information. Wiley (1998)
26. Rodríguez, A., Fernández-Medina, E., Piattini, M.: A BPMN extension for the modeling of security requirements in business processes. IEICE Trans. on Information and Systems 90(4), 745–752 (2007)
27. Rushby, J.: Using model checking to help discover mode confusions and other automation surprises. Reliability Engineering and System Safety 75, 167–177 (2002)
28. Sadiq, W., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 149–164. Springer, Heidelberg (2007)
29. Saleem, M., Jaafar, J., Hassan, M.: A domain- specific language for modelling security objectives in a business process models of soa applications. AISS 4(1), 353–362 (2012)
30. Salnitri, M., Dalpiaz, F., Giorgini, P.: Aligning service-oriented architectures with security requirements. In: Meersman, R., et al. (eds.) OTM 2012, Part I. LNCS, vol. 7565, pp. 232–249. Springer, Heidelberg (2012)
31. Samarati, P., di Vimercati, S.C.: Access control: Policies, models, and mechanisms. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, pp. 137–196. Springer, Heidelberg (2001)
32. Schmidt, R., Bartsch, C., Oberhauser, R.: Ontology-based representation of compliance requirements for service processes. In: Proc. of CEUR 2007 (2007)
33. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., Paige, R.: Large-scale complex it systems. Commun. ACM 55(7), 71–77 (2012)
34. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. JSA 55(4), 211–223 (2009)
35. Yip, F., Wong, A.K.Y., Parameswaran, N., Ray, P.: Rules and ontology in compliance management. In: In Proc. of EDOC, pp. 435–435 (2007)

Supervised vs. Unsupervised Learning for Intentional Process Model Discovery

Ghazaleh Khodabandelou, Charlotte Hug,
Rebecca Deneckère, and Camille Salinesi

Centre de Recherche en Informatique, Panthéon-Sorbonne University, France
{Ghazaleh.Khodabandelou}@malix.univ-paris1.fr,
{Charlotte.Hug,Rebecca.Deneckere,Camille.Salinesi}@univ-paris1.fr

Abstract. Learning humans' behavior from activity logs requires choosing an adequate machine learning technique regarding the situation at hand. This choice impacts significantly results reliability. In this paper, Hidden Markov Models (HMMs) are used to build intentional process models (Maps) from activity logs. Since HMMs parameters require to be learned, the main contribution of this paper is to compare supervised and unsupervised learning approaches of HMMs. After a theoretical comparison of both approaches, they are applied on two controlled experiments to compare the Maps thereby obtained. The results demonstrate using supervised learning leads to a poor performance because it imposes binding conditions in terms of data labeling, introduces inherent humans' biases, provides unreliable results in the absence of ground truth, etc. Instead, unsupervised learning obtains efficient Maps with a higher performance and lower humans' effort.

Keywords: Supervised Learning, Unsupervised Learning, Intentional Process Modeling, Hidden Markov Models.

1 Introduction

Fueled by the impressive growth of events logs in organizations, process mining field has emerged a few years ago as a key approach to design processes [1, 2]. Mining processes from logs can be useful for understanding how humans really work, analyzing how actual processes differ from the prescribed ones (conformance checking). This allows improving models, methods and products.

Whereas most process mining approaches specify behaviors in terms of sequences of tasks and branching [2], research on method engineering and guidance shows that an explicit use of intentions in process models structure could effectively mitigate the method engineering issues such as rigidity or lack of adaptation [3–7].

Intention-oriented process modeling emerged at the early 90s, as a driving paradigm. It allows supporting guidance [8], handling traceability matters [4], guiding requirements elicitation, surveying strategic alignment [9], defining actors and roles, specifying the outcome of business process models [10], describing intentional services [11], diagnosing use cases, analyzing users behavior, customizing methods or make them more flexible [3], etc. Defining strategies and

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 215–229, 2014.

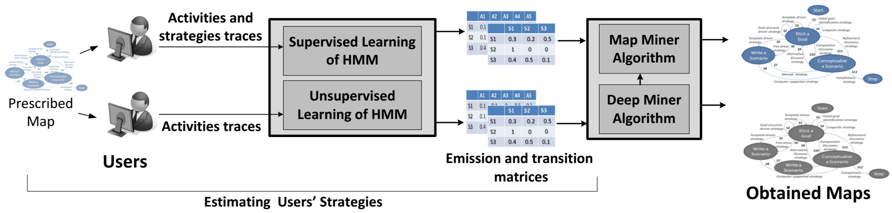


Fig. 1. The overview of Map Miner Method

intentions in process model structure has convinced as a robust mean to identify and analyze the relationships between processes, to understand the deep nature of processes, and to visualize any process (simple or complex) under a reduced and human-understandable form [5]. While intention-oriented process modeling has a longer tradition, it has largely neglected event logs so far. Map Miner Method (MMM) is a novel approach of process mining, which aims at constructing intentional process models from users' event logs. As a first step, the MMM framework uses Hidden Markov Models (HMMs) [12] to estimate users' strategies. Then, it generates intentional process models (Maps [5]) using Deep Miner and Map Miner algorithms [13]. This paper focuses on the first part of MMM: estimating users' strategies. These strategies can be estimated either with *supervised learning* or with *unsupervised learning*. While Supervised learning can be used when there is *a priori* knowledge about strategies in dataset, unsupervised learning can be used when there is no such knowledge available. Both learning approaches aim at characterizing the strategies that correspond the best to the users' activities in the event logs. These strategies will then be used to construct intentional process models. Thus, the choice of learning approach significantly impacts the discovered model accuracy. Hence, it is important to study limitations, advantages and conditions of use for these learning approaches.

The contribution of this paper is twofold: (i) first, in a theoretical context, it compares supervised and unsupervised learning of strategies in terms of convergence speed (complexity) and likelihood; and (ii) second, in an experimental context, it compares the intentional process models obtained with both learning approaches. The resulting Map process models provide a precious understanding of these approaches in terms of their performance as well as their conditions of use. Figure 1 depicts an overview of MMM framework, in which the focus of this paper is shown in the part of estimating users' strategies.

The remainder of this paper is organized as follows: in Section 2, we introduce the MMM and a brief definition of Map process models. In Section 3, supervised and unsupervised learning are described and then formally compared. In Section 4, both approaches are applied on two real datasets. Section 4.3 discusses the results of both approaches as well as the threats to validity. Related works are investigated in Section 5. Finally, Section 6 concludes this work and presents the perspectives.

2 Map Miner Method

MMM automatically constructs intentional process model from users' event logs. MMM consists of three phases: (1) it estimates users' strategies from activity logs using HMMs, (2) it constructs fine-grained Map process models from estimated strategies using Deep Miner algorithm (1), and (3) it constructs coarse-grained Map process models from fine-grained ones using Map Miner algorithm (2) (Figure 1). As mentioned earlier, this paper concentrates only on the first phase of MMM, *i.e.*, estimating users' strategies with the Map formalism. Among other intentional process models such as KAOS [14] or I* [15], we chose the Map formalism for several reasons: (a) it combines intentions and strategies at different abstraction levels, which allows handling large-scale and complex processes [5], (b) it supports process variability and flexibility by defining different strategies to fulfill a given intention, and (c) it has proven to be effective to specify business processes, user requirements, systems functionality, engineering methods, software engineering processes, etc [7]. Next parts explain briefly the Map metamodel and how HMMs can be adapted to it.

2.1 The Map Metamodel

The Map formalism [7] combines the concepts of *intention* and *strategy* with hierarchical abstraction levels and refinement links. In this framework, an intention is defined as a goal, an objective or a motivation to achieve with clear-cut criteria of satisfaction, which can be fulfilled by enacting a process [16]. The intentions are explicitly represented and form the high-level goals (*e.g.*, organizational goals). A Map process model (an instance of Map metamodel) specifies the multiple ways of working (*i.e.*, strategies) for fulfilling a set of intentions during the enactment of a process. For example, on Figure 2, one way to fulfill the intention *Specify an entity* is to select the strategy S_4: *By generalization*. In the next section, we explain how strategies can be linked to the activities logs.

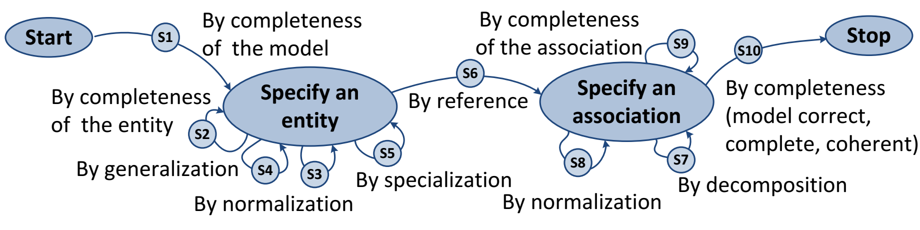


Fig. 2. Map process model for the construction of Entity/Relationship diagrams

2.2 Estimating Strategies from Activity Logs

Among the techniques to model different aspects of humans' behavior [17], HMMs have been proven to be appropriate for modeling the real world process, particularly unobservable cognitive states [18], such as underlying users' strategies. HMMs are stochastic Markov chains used for modeling a hidden sequence by a finite number of states. More precisely, HMMs consist of two complementary Markov processes: hidden and observed processes, such that the states of hidden process generate the symbol of observed process. It turns out that the topology of HMMs is particularly adapted to model the relation between strategies and activities in the Map formalism. To make it clear, let us consider an example for a Map process model enacted with 2 strategies and an HMM realized with 2 hidden states (see Figure 3). As shown this figure, strategies are used to move from one intention to another and are made of one or several activities. For instance, the strategy 1 allows moving from intention a to intention b and it is made of activities a_1, a_3 and a_4. The same structure can be found in an HMM, where hidden states generate observations. In other words, hidden state 1 generates the observations a_1, a_3 and a_4, or hidden state 2 generates the observations a_4 and a_7. This similar topology motivates using HMMs to model activity logs and users' strategies.

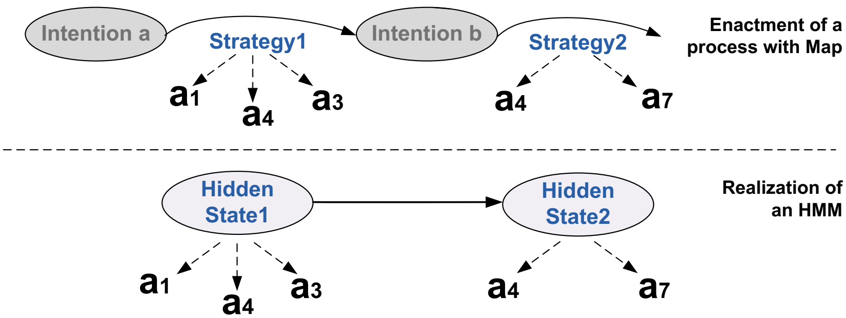


Fig. 3. An example for a Map process model enacted with 2 strategies (above) and an HMM realized with 2 hidden states (below)

Formally, the definition of an HMM is the tuple: $\mathcal{H} = \{\mathcal{S}, \mathcal{A}, \mathbf{E}, \mathbf{T}, \pi\}$, where $\mathcal{S}$ is the set of possible hidden states, $\mathcal{A}$ represents the set of possible observations, $\mathbf{T}$ is the states transition matrix, *i.e.*, the matrix which represents the probabilities of transition from one state to another, $\mathbf{E}$ is the observations emission matrix, *i.e.*, the matrix which represents the probabilities that a given observation appears in a given hidden state, and π is the vector made of the initial probabilities of hidden states. In the MMM framework, the users' strategies are modeled by the hidden process and the users' activities are modeled by the observed process.

Hidden Process: Users' Strategies. Let $\mathbf{s} = (s_1, \ldots, s_L) \in \mathcal{S}^L$ be a temporal sequence of users' strategies of length L. The hidden process of strategies is parametrized by the vector π, $\pi(u) = \Pr(s_1 = u)\ \forall u \in \mathcal{S}$ and the matrix $\mathbf{T}$ such that:

$$\mathbf{T}(u, v) = \Pr(s_\ell = v | s_{\ell-1} = u)\ \forall u, v \in \mathcal{S},\ \ell \in [2, L], \quad (1)$$

Observed Process: Users' Activities. Let $\mathbf{a} = (a_1, \ldots, a_L) \in \mathcal{A}^L$ be a temporal sequence of users' activities of length L. The emission probability of an observation $a \in \mathcal{A}$ for a given strategy $u \in \mathcal{S}$ is given by:

$$\mathbf{E}(a, u) = \Pr(a|u). \tag{2}$$

Assuming that $\mathcal{S}$, $\mathcal{A}$ and π are known, the HMM model is fully described by $\{\mathbf{E}, \mathbf{T}\}$, which represent the core information about the HMM behavior. These two matrices provide all the necessary information to characterize the strategies of the Map. Indeed $\mathbf{E}$ gives the relation between each strategy and the activities in the event logs, and $\mathbf{T}$ gives the transition probabilities between strategies. These two matrices have to be learned from the logs and the choice of the learning approach is crucial to ensure that the strategies are correctly characterized.

3 Learning Approaches to Estimate Users' Strategies

As discussed in section 2, the characterization of the strategies of the Map completely relies on the model parameters of the HMM, *i.e.*, the emission matrix $\mathbf{E}$ and the transition matrix $\mathbf{T}$. The learning problem is to find $\mathbf{E}$ and $\mathbf{T}$ that maximize the probability of generating the observed sequences of activities. There are two learning approaches for estimating these matrices: *Supervised* or *Unsupervised* learning. this section discusses in a theoretical context the necessary conditions for using both approaches as well as their respective performances.

3.1 Supervised Learning

Supervised learning aims at learning $\mathbf{E}$ and $\mathbf{T}$. However, the conditions under which it can be used are very restrictive and the results might be biased.

Conditions of Use. The application of this method requires the knowledge of: (a) the sets $\mathcal{A}$ and $\mathcal{S}$, and (b) some sequences of activities $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and their associated sequences of strategies $\mathbf{s}_1, \ldots, \mathbf{s}_N$.

While the knowledge of $\mathcal{A}$ and $\mathbf{a}_1, \ldots, \mathbf{a}_N$ is generally not an issue (the possible activities of a given process are usually known and are recorded in traces), the knowledge of $\mathcal{S}$ and $\mathbf{s}_1, \ldots, \mathbf{s}_N$ is more problematic. Indeed, since strategies are the cognitive operators, the usual way to obtain the set $\mathcal{S}$ is to refer to experts. Since humans' judgment is involved, the obtained set $\mathcal{S}$ can be biased. We argue that in a cognitive context such as a humans' strategy and intention, it is impossible to properly label the training data, because this information is not observable. Moreover, humans' bias [19] is unavoidably introduced into training data labeling, which significantly impacts the learning process and may produce incorrect or uninformative process models. Moreover, strategies are usually not recorded in traces [20]. Applying this learning method implies to conduct experiments specially designed to record traces of activities and traces of strategies. This condition highly restricts the range of use of this method in large-scale.

Performance. Given N sequences of activities $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and their associated N sequences of strategies $\mathbf{s}_1, \ldots, \mathbf{s}_N$, the aim of supervised learning is to find the couple $(\mathbf{E}^*, \mathbf{T}^*)$ which maximizes the likelihood of generating $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and $\mathbf{s}_1, \ldots, \mathbf{s}_N$:

$$(\mathbf{E}^*, \mathbf{T}^*) = \arg\max_{\mathbf{E},\mathbf{T}} \prod_{n=1}^{N} \Pr(\mathbf{a}_n | \mathbf{s}_n, \mathbf{E}, \mathbf{T}) \tag{3}$$

Obtaining the coefficient of $\mathbf{T}^*$ amounts in counting the number of transitions from one strategy to another and obtaining the coefficients of $\mathbf{E}^*$ amounts to count the number of occurrences of each activity during each strategy, as shown below:

$$\mathbf{T}^*(u, v) = \frac{\mathrm{Num}(u, v)}{\sum_{w \in \mathcal{S}} \mathrm{Num}(u, w)}, \ \forall (u, v) \in \mathcal{S}^2, \tag{4}$$

$$\mathbf{E}^*(u, a) = \frac{\mathrm{Num}(a|u)}{\mathrm{Num}(a)}, \ \forall u \in \mathcal{S}, \ \forall v \in \mathcal{A}, \tag{5}$$

where $\mathrm{Num}(u, v)$ denotes the number of transitions from strategy u to strategy v in the traces $\mathbf{s}_1, \ldots, \mathbf{s}_N$, $\mathrm{Num}(a)$ denotes the number of occurrences of activity a in $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and $\mathrm{Num}(a|u)$ denotes the number of occurrences of activity a while the strategy is u, in $\mathbf{s}_1, \ldots, \mathbf{s}_N$ and $\mathbf{a}_1, \ldots, \mathbf{a}_N$. The computation complexity of this method is very low since all the coefficients of $\mathbf{E}^*$ and $\mathbf{T}^*$ can be directly computed from the traces used for learning with (4) and (5).

The set of training sequences $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and $\mathbf{s}_1, \ldots, \mathbf{s}_N$, is extremely important for the accuracy of the estimation of $\mathbf{E}^*$ and $\mathbf{T}^*$. If the set contains few traces, or they are not fully representative of all the traces that can be produced by the process, the HMM model learned out of it might suffer underfitting issues. From a practical point of view, this issue is common since the conditions to get usable training traces are complex (resulting in few usable traces).

3.2 Unsupervised Learning

Unsupervised learning estimates the matrices $\mathbf{E}$ and $\mathbf{T}$ based only on traces of activities. Since there is almost no prior knowledge on the strategies set $\mathcal{S}$, this method is significantly less biased than supervised learning but the associated computational complexity is high.

Conditions of Use. For unsupervised learning, the required knowledge includes the set of activities $\mathcal{A}$, some traces of activities $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and the cardinality of the set $|\mathcal{S}|$, *i.e.* the number of possible strategies. Regarding strategies, neither the set $\mathcal{S}$ nor some traces of strategies $\mathbf{s}_1, \ldots, \mathbf{s}_N$ should be known, only the number of possible strategies is required. This parameter can be chosen by experts (*e.g.* as a way to set the level of complexity of the model) or can be set with techniques such as BIC [21], which makes a trade-off between the likelihood of the model and its complexity. Similarly to supervised learning, this choice introduces a bias, but given that only the number of strategies is set and not the strategies themselves, this bias is less important. The advantage of unsupervised learning is being applicable on datasets comprising only activities traces.

Performance. The Baum-Welch algorithm (BWA) [22] is the most commonly used in HMMs framework to estimate the model parameters $\mathbf{E}$ and $\mathbf{T}$. It uses the Expectation Maximization (EM) algorithm [23]. The aim of the EM algorithm is estimating the *maximum a posteriori* of the statistical model (with latent variable) parameters in an iterative way. Given a dataset made of N observed sequences of activities $\mathbf{a}_1, \ldots, \mathbf{a}_N$, the BWA finds the HMM matrices $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{T}}$ that locally maximize the probability of having these sequences generated by the HMM. More precisely, the BWA maximizes the likelihood of $\mathbf{E}$ and $\mathbf{T}$:

$$\left(\tilde{\mathbf{E}}, \tilde{\mathbf{T}}\right) = \arg\max_{\mathbf{E},\mathbf{T}} \prod_{n=1}^{N} \Pr\left(\mathbf{a}_n | \mathbf{E}, \mathbf{T}\right) \tag{6}$$

As we mentioned earlier, the number of strategies is required to know the dimensions of matrices $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{T}}$ since the BWA could not run without $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{T}}$ being initialized.

What is interesting to note here is the fact the likelihood is not maximized depending on some traces of strategies $\mathbf{s}_1, \ldots, \mathbf{s}_N$, as it was the case for supervised learning. It means that the space in which the likelihood is maximized is larger than the space for supervised learning. As a consequence,

$$\max_{\mathbf{E},\mathbf{T}} \prod_{n=1}^{N} \Pr\left(\mathbf{a}_n | \mathbf{E}, \mathbf{T}\right) \geqslant \max_{\mathbf{E},\mathbf{T}} \prod_{n=1}^{N} \Pr\left(\mathbf{a}_n | \mathbf{s}_n, \mathbf{E}, \mathbf{T}\right). \tag{7}$$

In other words, the maximum likelihood obtained by unsupervised learning is always higher than the maximum likelihood obtained by supervised learning since the latter comes from a constrained space. Unfortunately, the BWA cannot be guaranteed to converge to the global maximum likelihood since it is only proved to converge to a local optimum [12]. The limit of convergence depends on the initialization of the matrices $\mathbf{T}$ and $\mathbf{E}$ and it is verified by our experimental results (see section 4), that a simple initialization of $\mathbf{T}$ and $\mathbf{E}$ leads to a maximum likelihood of unsupervised learning higher than supervised learning.

Another difference with supervised learning is the computational complexity. While the complexity of supervised learning is very low, the BWA requires several iterations to converge to a local optimum. These iterations make unsupervised learning a more expensive method than supervised learning. The precise computation of both methods, applied on tow experiments, are given in section 4.3.

3.3 Summary of the Two Learning Approaches

In table 1, we present a theoretical comparison of the two learning approaches, based on the properties defined in sections 3.1 and 3.2. Regarding the conditions of use, unsupervised learning can be applied on any dataset comprising traces of activities, contrary to supervised learning which can be applied under more restrictive conditions. This makes unsupervised learning the most convenient method for a practical use. However, since unsupervised learning is only proved

Table 1. Theoretical comparison of supervised and unsupervised learning

	Traces for learning	**A-priori knowledge**	**Convergence speed (complexity)**	**Likelihood of the estimated parameters**
Supervised learning	Activities, Strategies	Set of activities, Set of strategies	Fast (one iteration)	Maximum over a restrained set
Unsupervised learning	Activities	Set of activities, Number of strategies	Slow (several iterations)	Local maximum

to converge to a local maximum, it is not guaranteed to provide an estimated model with a better likelihood than supervised learning. In order to investigate about this point, we compare both approaches on the same datasets in the following section.

4 Comparison of the Approaches in Experiments

To compare the supervised and unsupervised learning in a experimental context, we conducted two tailored experiments with the Master students of computer science of Sorbonne University: *Entity/relationship (E/R) diagrams* and *E-shopping*. Due to lack of space, we only show and analyze in details the strategies obtained for E/R diagrams in the current section. In section 4.3, the results of the two learning approaches for the two experiments are compared in terms of several comparison criteria, such as performance (indicated by the likelihood to generate the activities in the event logs), humans efforts, convergence speed, and computation complexity. Note that the traces used for this comparison have to be compatible with both learning approaches. Indeed, they have to comprise traces of activities and corresponding traces of strategies. Although strategies are generally not accompanied recorded logs, we intentionally asked the students participating in the experiments to label their traces of activities to indicate which strategies they followed. The traces of activities are recorded by a web-based tool.

• **E-shopping Experiment:** we guided students through a prescribed intentional process model to buy a present. We recorded 90 traces of activity produced by 90 students for which we know the sequence of selected strategies. The size of each trace varies between 6 and 40 activities. Note that with both learning approaches, all the traces are used for training.

• **The Entity/Relationship Diagrams Experiment:** according the intentional process model given by Figure 2, 66 students created E/R diagrams. Here again, all the traces are used for training for both learning approaches. Regarding to this model, students can select ten strategies to fulfill three intentions, *i.e.*, *Specify an entity*, *Specify an association* and *Stop*. From *Start*, it is possible to progress in the process by selecting strategies leading to the intentions but once the intention of *Stop* is achieved, the enactment of the process finishes. To fulfill an intention following a strategy, students have to carry out activities.

Table 2. A fragment of some event Log of E/R diagrams experiment

UserID	**TraceID**	**Timestamps**	**StrategyLabel**	Activities	...
45	7	31/10/12 14:54:00	1	Create entity	...
22	1	31/10/12 15:14:00	4	Create generalization link	...
12	8	31/10/12 14:54:00	7	Create association	...
45	7	23/10/12 09:41:00	2	Link attribute to entity	...
...	...	...	...	...	...

There are 12 unique activities related to the process. Table 3 gives the name of the strategies, related activities and the corresponding labels. The size of each trace varies between 3 and 68 performed activities. Table 2 illustrates a fragment of some event logs of the E/R diagrams experiment. Each row represents an activity, with its corresponding labeled strategy, its timestamps, the trace ID, and the ID of the user who performed the activity. A trace of activities is the ordered (by timestamps) sequence of activities that a user performed. In sections 4.1 and 4.2, the MMM is applied on this dataset with both supervised and unsupervised learning and strategies and associated Maps are discussed.

Table 3. Strategies and related activities

Labels	**Name of strategies**	**Related activities (activities labels)**
S_1	By completeness (model)	Create entity (a_1)
S_2	By completeness (entity)	Link attribute to entity (a_2)
S_3	By normalization	Delete Link attribute to entity (a_6), Delete entity (a_5), Define primary key (a_7)
S_4	By generalization	Create entity (a_1), Create generalization link (a_3)
S_5	By specialization	Create entity (a_1), Create specialization link (a_4)
S_6	By reference	{Delete link attribute to entity, Create entity, Create association, Link association to entity} (a_8), {Create association, Link association to entity} (a_9)
S_7	By decomposition	{Create association, Link association to entity} (a_9)
S_8	By normalization	{Delete association, Delete Link attribute to association}(a_{10})
S_9	By completeness (assoc.)	Link attribute to association (a_{11})
S_{10}	By completeness (final)	Check the model (a_{12})

4.1 Supervised Learning for MMM

First, we apply supervised learning on activities traces, to estimate the strategies, as explained in section 2. Since, we had the advantage of setting up the experiments, we were able to record students' strategies sequences in addition to their activity sequences (labeling the activities). This allows estimating the strategies by supervised learning. The inputs of the algorithm are the activities traces and their related strategies. The matrices **E** and **T** can then be computed, and provide the relation between strategies and activities and the transition probabilities between strategies, respectively. Since the learning was supervised, the relations between the strategies and the activities are in line with table 3. However, the transitions between strategies indicate that these latter have not been followed exactly as they were prescribed. By successively applying the Deep Miner algorithm and the Map Miner algorithm to the transition matrix **T**, we extract a Map which emphasizes this phenomenon. This Map is displayed in Figure 4.

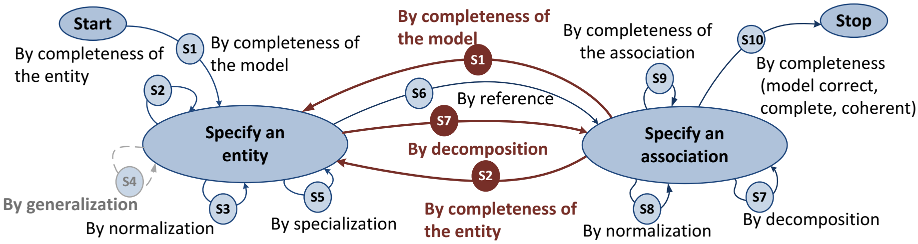


Fig. 4. Map process model obtained by supervised MMM

Therefore, it is possible to detect students' deviations from the prescribed Map. In particular, S_4 is never taken by students and S_7 is chosen in the wrong section. Regarding the strategies S_1 and S_2, they have selected by students from intention *Specify an association* to intention *Specify an entity*. This is not shown in the prescribed Map. However, these transitions are not surprising since they are allowed by the intrinsic semantic of the Map process model. Indeed, a given user can return to an intention already fulfilled to start another section.

4.2 Unsupervised Learning for MMM

By applying unsupervised learning on the activities traces, we discover a different set of strategies and a different Map process model. We recall that in this case, no strategies sequences are necessary as inputs to run the learning algorithm. Consequently, only the traces of activities are necessary used. the discovered strategies are detailed in table 4. In particular, from the emission matrix **E**, we obtain the relation between strategies and activities. Contrary to supervised learning, since no prior information about strategies is available, the names of strategies and intentions are not known. However, based on the names of activities, it is possible to discover the main topics of the strategies. Through a semantic analysis the strategies name can be inferred. As for supervised learning, a Map can be extracted from the estimated transition matrix **T**, it is displayed in Figure 5. Except *Start* and *Stop*, two intentions denoted by I'_2 and I'_3 are inferred. The comparison of this Map with the one obtained from supervised learning is made in the next section.

4.3 Discussion and Threats to Validity

It is interesting to discuss the results obtained in previous sections from both quantitative (models likelihood, algorithm convergence, complexity) and qualitative (models interpretation) points of view.

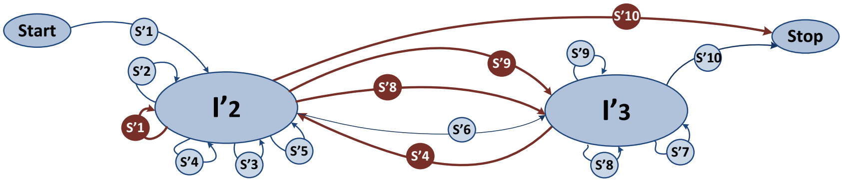


Fig. 5. Map process model obtained by unsupervised MMM

Table 4. Intentions, Strategies and Activities obtained by unsupervised learning for ER diagrams

Intentions	**Code**	**Activities**	**Strategies Topics obtained by MMM**
Start $\rightarrow I'_2$	S'_1	a_1 (0.94)	entity, creation, specify
I'_2	S'_1	a_1 (0.94)	entity, creation, specify
	S'_2	a_2 (0.88), a_1 (0.09)	attribute, entity, creation
	S'_3	a_2 (0.09), a_3 (0.13), a_4 (0.39), a_5 (0.40)	entity, delete, creation, specialize
	S'_4	a_1 (0.11), a_2 (0.54), a_8 (0.25)	delete, creation, attribute, entity, association
	S'_5	a_5 (0.1), a_6 (0.63), a_7 (0.28)	primary key, creation, entity
$I'_2 \rightarrow I'_3$	S'_6	a_1 (0.15), a_2 (0.79)	creation, entity, attribute, link
	S'_8	a_1 (0.09), a_2 (0.81), a_9 (0.08)	association, entity, link, attribute, creation
	S'_9	a_1 (0.37), a_9 (0.19), a_{11} (0.34)	creation, association, entity, attributes
$I'_3 \rightarrow I'_2$	S'_4	a_1 (0.11), a_2 (0.54), a_8 (0.25)	delete, creation, attribute, entity, association
I'_3	S'_7	a_9 (0.83), a_{10} (0.05), a_{11}(0.05)	link, creation, delete, entity, association
	S'_8	a_1 (0.09), a_2 (0.81), a_9 (0.08)	association, entity, link, attribute, creation
	S_9	a_1 (0.37), a_9 (0.19), a_{11}(0.34)	creation, association, entity, attributes
I'_3 $\rightarrow$ **Stop**	S'_{10}	a_9 (0.08), a_{12} (0.87)	check, model, coherent
I'_2 $\rightarrow$ **Stop**	S'_{10}	a_9 (0.08), a_{12} (0.87)	check, model, coherent

• Adopting a qualitative point of view, for the E/R diagrams experiment, although some strategies from the prescribed Map and the Map obtained by unsupervised learning are similar (S_1 and S'_1, S_2 and S'_2, S_{10} and S'_{10}, S_7 and S'_7), most strategies from unsupervised learning cannot be exactly identified to prescribed strategies. It is not due to a poor compliance of the Map obtained by unsupervised learning but due to the supervised learning assumption, *i.e.* the prescribed Map is actually followed by students. This assumption is not true. Indeed, during the enactment of the process, students may deliberately or accidentally not follow the prescribed Map. Consequently, assuming that the prescribed model is followed by students creates a bias in the definition of strategies and intentions. In addition, there is no ground truth for labeling the activities sequences. Consequently, the labeling could be flawed as it is a subjective process. Moreover, assigning the labels to the strategies and intentions constrains the discovered Map to a limited space which leads to poor performance of supervised learning. This phenomenon can be verified with the deviations of students detected by obtained Maps. Whereas the Map obtained by supervised learning detected only two deviations (S_4 and S_7), the Map obtained by unsupervised learning detected five deviations which are not the same as the supervised learning ones (S'_1, S'_4, S'_8, S'_9, S'_{10}).

• The log-likelihood of the strategies estimated by unsupervised learning in both experiments is higher than supervised learning (see Table 5). In other words, the strategies estimated by unsupervised learning have a higher chance to generate the activities observed in the event logs. It makes unsupervised learning more trustworthy than supervised learning. Note that this result is in line with the theoretical study performed in section 3.

Table 5. Practical comparison of supervised and unsupervised learning

Experiments	Learning	Traces for learning	A-priori knowledge	Convergence speed (complexity)	The estimated parameters Log-likelihood
E/R diagrams	**Supervised learning**	66 traces of activities, 66 traces of strategies	set of activities, set of strategies	1 iteration	$-2.54e^3$
	Unsupervised learning	66 traces of activities	set of activities, number of strategies	9, 986 iterations	$-2.36e^3$
E-Shopping	**Supervised learning**	90 traces of activities, 90 traces of strategies	set of activities, set of strategies	1 iteration	$-2.81e^3$
	Unsupervised learning	90 traces of activities	set of activities, number of strategies	4, 325 iterations	$-1.69e^3$

• From a cost-benefit and human-centric point of view, cognitive tasks are time-consuming and labor intensive. Since the methodology of labeling activities cannot be generalized to common event logs, this is one serious drawback for supervised learning. Thus, the cost of labeling the data for supervised learning approach is quite high as it involves the students' commitment to label and comment their activities at each step of the process. In comparison, the only humans' effort for unsupervised learning is to choose the number of strategies for the intentional process model. Nevertheless, the unsupervised learning requires a minimal humans' intervention and it allows obtaining intentional process models that match the actual enacted process. The drawbacks of unsupervised learning are a higher computation complexity and the need to automate the naming of obtained strategies and intentions.

• The BWA cannot be guaranteed to converge to the global maximum likelihood (see Section 3.2). The convergence depends on the initialization of the matrices **T** and **E** and it converges at 9, 986 learning iterations for E/R diagrams and at 4, 325 learning iterations for E-shopping. The supervised algorithm converges in the first iteration.

• While the complexity of the BWA is high due to its requirement to several iterations until the convergence to a local optimum, complexity of supervised learning is very low. Table 5 presents a summary of both learning approaches for tow experiments.

5 Related Work

To the best of our knowledge, there is no work comparing supervised and unsupervised learning for process model discovery from event logs. For this reason we position our work with respect to process mining techniques. Process mining approaches propose to model users' behaviors in terms of activities [2]. These techniques aim at recovering the original workflow from event logs, which containing the traces of processes enactments. Process mining approaches use different algorithms and techniques, such as classification and learning techniques to extract the information from event logs [24–27]. Some of these techniques are investigated hereafter.

Inference methods infer process models with a tradeoff between accuracy and noise robustness. Cook compares in [28] three inference algorithms of RNet [29], Ktail [30] and Markov models [31] for process discovery. The latter two are considered as the most promising approaches. Genetic algorithm [27] provides process models (Petri nets) built on causal matrix, *i.e.*, input and output dependencies for each activity. This technique tackles problems such as noise, incomplete data, non-free-choice constructs, hidden activities, concurrency, and duplicate activities. Nevertheless, it requires the configuration of many parameters to deal with noise and irrelevant data, which is a complex task. Directed acyclic graphs [32] proposes to transform the events into dependency graphs or workflow graphs using directed acyclic graph, representing events and their causal relations without loop. However, using this kind of graphs to model the processes is delicate as loops exist in process models. To tackle this challenge, this approach tries to count the tasks frequencies and then fold the graph. Nevertheless, the results are partially satisfying and the model does not completely fit the actual process. However, all these approaches neglect the underlying humans' cognitive operators such as users' strategies and intentions. In process mining field, HMMs are used in context of inductive workflow acquisition [26] to model a workflow or in [25] as a conformance checking technique. Each state of HMMs corresponds to a task node. The event logs can be observed and generated into workflow nets by inductive learning. This approach supports appearing the same tasks at several times in the model (duplicate tasks). It is similar to the approach of directed acyclic graphs [32] due to the presence of the splits and joins in the transformation step. However, the works using HMMs do not consider the hidden states as the cognitive states such as intentions/strategies.

6 Conclusions

In this paper, we have compared the supervised and unsupervised learning approaches, in both theoretical and practical contexts, to understand which approach allows discovering underlying users' strategies optimally. Applied on a real dataset to obtain Map process models, the results demonstrate that several issues hinder the application of supervised learning in modeling humans' cognitive process, such as considerable humans' involvement in terms of data labeling, introducing inherent humans' biases and lack of accurate ground truth. Therefore, we deduce from our study that unsupervised learning offers better results than supervised learning to discover intentional process models (Maps).

Although the MMM automatically discovers the topology of the intentional model, the names of strategies and intentions are still inferred semi-automatically. However, the logical relations between activities, strategies and intentions established in the intentional model could be exploited to build an ontology to fully automate the process of inferring the names of strategies and intentions. In addition, we are developing an ProM [33] plug-in which will allow modeling processes in intentional manner. This provides our community to have a vision of processes from an intentional angle.

References

1. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM TOSEM 7(3), 215–249 (1998)
2. van der Aalst, W.: Process mining: Discovery, conformance and enhancement of business processes. Springer, Heidelberg (2011)
3. Mirbel, I., Ralyté, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. Requirements Engineering 11(1), 58–78 (2006)
4. Jarke, M., Pohl, K.: Establishing visions in context: Towards a model of requirements processes. In: ICIS, pp. 23–34 (1993)
5. Rolland, C., Salinesi, C.: Modeling goals and reasoning with them. In: Engineering and Managing Software Requirements, pp. 189–217. Springer, Heidelberg (2005)
6. Plihon, V., Rolland, C.: Modelling ways-of-working. In: Iivari, J., Rossi, M., Lyytinen, K. (eds.) CAiSE 1995. LNCS, vol. 932, pp. 126–139. Springer, Heidelberg (1995)
7. Rolland, C., Prakash, N., Benjamen, A.: A multi-model view of process modelling. Requirements Engineering 4(4), 169–187 (1999)
8. Rolland, C.: Modeling the requirements engineering process. In: Information Modelling and Knowledge Bases, Budapest, Hungary, pp. 85–96 (May 1993)
9. Thevenet, L.-H., Salinesi, C.: Aligning is to organizations strategy: The instal method. In: Krogstie, J., Opdahl, A. L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 203–217. Springer, Heidelberg (2007)
10. Salinesi, C., Rolland, C.: Fitting business models to system functionality exploring the fitness relationship. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 647–664. Springer, Heidelberg (2003)
11. Rolland, C., Kirsch-Pinheiro, M., Souveyet, C.: An intentional approach to service engineering. IEEE Transactions on Services Computing 3(4), 292–305 (2010)
12. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proceedings of the IEEE, vol. 77(2), pp. 257–286 (1989)
13. Khodabandelou, G., Hug, C., Deneckere, R., Salinesi, C.: Unsupervised discovery of intentional process models from event logs. Submitted at: The 11th Working Conference on Mining Software Repositories, MSR 2014 (2014)

14. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1), 3–50 (1993)
15. Yu, E.: Modelling strategic relationships for process reengineering. Social Modeling for Requirements Engineering 11 (2011)
16. Soffer, P., Rolland, C.: Combining intention-oriented and state-based process modeling. In: Delcambre, L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 47–62. Springer, Heidelberg (2005)
17. Gray, W.D., John, B.E., Atwood, M.E.: The precis of project ernestine or an overview of a validation of goms. In: Human Factors in CS, ACM, pp. 307–312 (1992)
18. Hayashi, M.: Hidden markov models to identify pilot instrument scanning and attention patterns. IEEE Systems, Man and Cybernetics, 2889–2896 (2003)
19. Tversky, A., Kahneman, D.: Judgment under uncertainty: Heuristics and biases. Science 185(4157), 1124–1131 (1974)
20. Khodabandelou, G., Hug, C., Deneckere, R., Salinesi, C.: Supervised intentional process models discovery using hidden markov models. In: Proceed. of 7th Int. Conf. on Research Challenges in Information Science (2013)
21. Burnham, K.P., Anderson, D.R.: Model selection and multi-model inference: A practical information-theoretic approach. Springer, Heidelberg (2002)
22. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics 41(1), 164–171 (1970)
23. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. The Royal Statistical Society, 1–38 (1977)
24. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
25. Rozinat, A., Veloso, M., van der Aalst, W.M.: Evaluating the quality of discovered process models. In: 2nd WS on the Induction of Process Models, Citeseer (2008)
26. Herbst, J., Karagiannis, D.: Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In: Proceed. 9th Intl. Workshop on Database and Expert Systems Applications. IEEE (1998)
27. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Genetic process mining. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536,pp. 48–69. Springer, Heidelberg (2005)
28. Cook, J.E., Wolf, A.L.: Automating process discovery through event-data analysis. In: Software Engineering, ICSE 1995, pp. 73–73. IEEE (1995)
29. Das, S., Mozer, M.C.: A unified gradient-descent/clustering architecture for finite state machine induction. In: NIPS, pp. 19–26. Morgan Kaufmann (1994)
30. Biermann, A.W., Feldman, J.A.: On the synthesis of finite-state machines from samples of their behavior. IEEE Transactions on Computers, 592–597 (1972)
31. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. The Annals of Mathematical Statistics 37(6), 1554–1563 (1966)
32. 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. 467–483. Springer, Heidelberg (1998)
33. EUT: Prom, `http://www.processmining.org/prom/start` (November 2013)

Towards a Consistent Cross-Disciplinary Ontology for Business Process

Charles Crick[1] and Eng Chew[2]

[1] Faculty of Engineering and Information Technology
University of Technology, Sydney, Australia
charlescrick@computer.org

[2] Faculty of Engineering and Information Technology
University of Technology, Sydney, Australia
eng.chew@uts.edu.au

Abstract. This paper takes a cross-disciplinary view of the ontology of "business process": how the concept is treated in the IS research literature and how related concepts (with stronger human behavioural orientation) from organisational and management sciences can potentially inform this IS perspective. In particular, is there room for socio-technical concepts such as technology affordance, derived from the constructivist tradition, in improving our understanding of operational business processes?

The paper draws on the current research being pursued by the authors in developing a theoretical framework for understanding the role of IT in organisational agility. In this developing theoretical model, we are seeking to include the user-oriented socio-technical dimension that distinguishes the IT "as-used" from the IT "as-designed" in our use of business process as an organisational building block.

Keywords: Ontology, Business Process, Organisational Routine, Technology Affordance, Organisational Capability.

1 Introduction

Organisations, as the subject matter of research, have been approached from a number of perspectives ranging across the organisational sciences, management sciences and information systems (IS) disciplines. Over time and with the advance of technology, it could be said the research agendas across these disciplines have been brought closer together as understanding the role of technology in the modern organisation assumes a greater importance and focus. With research efforts "zeroing in" on the same types of organisational problems, particularly associated with the role of technology, it is timely to review how we might leverage and apply the various conceptual models that have been developed. Clearly each perspective offers valuable insight, albeit, within the constraints of the philosophical world view that each brings with it. In attempting to adopt a cross-disciplinary approach to the study of the organisation, the researcher is thus faced with competing philosophical positions, ontologies and epistemologies.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 230–241, 2014.

The idea of "business process" is a case in point. The concept is firmly entrenched in the IS literature, its use popularised by Hammer and Champy [1]. Weske's definition follows this lineage and is typical of the IS perspective "A business process consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations." [2]. This statement embodies four ingredients that characterise the ontology of business process from the IS point of view, namely: the notion of goal-directedness; process as a sequence of activities; organisational elements that perform the process; and technology that mediates the process. However, if one switches to the organisational sciences literature, one finds a parallel conceptualisation of "organisational building blocks" that are, on the one hand, analogous to the business process concept but at the same time different and based on different ontologies. Examples of these are organisational capabilities [3], organisational routines [4] and technology affordances [5].

In this paper we look in more detail at the various ways these organisational building blocks have been conceptualised in the literature across disciplines and ask these questions: How should these ontologies be related to each other? Can the traditional IS view of business processes be informed by the organisational theories developed in other disciplines? In particular, as we shall see, how should human agency be conceptualised as part of business processes? What are the questions this raises for the modelling and design of business processes in the modern organisation?

These questions reflect the motivation of our ongoing research into the role of IT in organisational agility. In our developing theoretical model, we are seeking to integrate the conventional concept of "business process" with the user-oriented, socio-technical dimension. Our goal is to achieve a greater understanding of IT "as-used" as distinct from the IT "as-designed within the organisation. In order to do this we have found some of the organisational terms such as capabilities, routines and affordances mentioned above as useful conceptualisations. The concepts of routines and affordances, in particular, highlight the human dimension as being integral to understanding the as-designed versus the as-used dichotomy, which has a bearing on business process design considerations. It remains to establish a basis for relating the ontologies that underpin these organisational building blocks in a consistent manner.

The remainder of the paper is organised as follows. Firstly we review the competing philosophical traditions from the intersecting research disciplines that have given rise to concepts such "business process", "organisational capability", "organisational routine", "technology affordance". We then examine how these organisational concepts can inform the ontology of business process. We conclude by positing a new conceptual model drawn from our own research that seeks to align these concepts, and suggesting the new model's implications for business process design.

2 Philosophical Traditions

One can recognize two underlying philosophical positions at play in organisational and IS research [6]. It is important to understand the influence these have had on how the "organisation" is conceived. On the one hand we have the *realist* world of objective reality where the researcher is free to observe, measure and develop testable theories about "real world" entities and their causal relationships. This is the realm of the deterministic physical and natural sciences that is largely responsible for modern scientific thought since The Enlightenment. In this paradigm, the organisation is conceived as an objective entity in which we can identify clearly delineated components such as resources, processes, humans and technologies. We can characterise relationships between them of the form "under condition A, B causes C" that can then inform our theories of how organisations work in a deterministic sense. The epistemology associated with the realist tradition is strong empiricism (positivism) in which objective phenomena are observed, measured and analysed [7].

The alternate *constructivist* paradigm, one that has developed in the social sciences, holds that the world, as observed by humans (including the researchers themselves), is a social construction such that any true "objective" reality cannot directly be perceived. In this world view, the organisation is conceived in terms of complex, messy social interactions in which causality in the scientific sense is difficult to determine [6]. Each person's perception of the world is coloured through the lens of the meanings they ascribe to it. The epistemology here is interpretivist, and researchers in this field attempt to interpret these "meanings" typically through rich and complex case study information [8].

Notwithstanding these ontological and epistemological differences, *both* the physical and social sciences are relevant in the IS discipline. In most if not all cases, the "information system" in question comprises human behaviour intersecting with material world entities such as technology.

Recent work [8, 9] has shown that the dichotomy could be bridged in some sense by the "critical realist" perspective, which admits a stronger form of causality into the interpretivist fold. Others have pointed to complexity theory as a unifying mechanism [6].

Gregor [10] argues that, in developing IS theory (be it descriptive, explanatory or predictive) the choice of underlying epistemology is not important per se. In other words valid IS theory can be built with any of those mentioned. We therefore remove ourselves from any further philosophical discussion by reiterating this view: that the theory itself once developed has a validity that is independent of the philosophical tradition under which it was developed. This means we do not have to particularly embrace a positivist or interpretivist position to leverage the theoretical organisational work that will be further discussed in the following section.

3 Organisational Building Blocks

3.1 Terminology

Besides the ontological debate, there are some more prosaic issues that confront the prospective cross-disciplinary organisational scholar. One of these is the characterisation of the organisational building blocks themselves. Over time some "generally understood" common definitions have emerged that have allowed strands of research to cross-fertilise and propagate. Similarly, however, inconsistency and confusion have also arisen in some of the concepts and terminology. Dosi et al. express it colourfully when referring to one of these: “The term ‘capabilities’ floats in the literature like an iceberg in a foggy Arctic sea, one iceberg among many, not easily recognized as different from several icebergs nearby” [3 p.3]. This could equally apply to the term “routine”. In fact capability and routine are the terms most frequently used in the organisational sciences to describe what it is the organisation does and how it does it, including how it uses IT. The ontology of these building blocks in particular and their relationship to the more familiar IS term, business process, is the subject of the discussion that follows. In this discussion we navigate the “icebergs” by representing well established perspectives and noting controversy where it exists.

3.2 Organisational Routines

Feldman and Pentland define an organisational routine as “a repetitive, recognizable pattern of interdependent actions, involving multiple actors” [11]. Those not familiar with the term would immediately notice the similarity with how we generally understand “business process”. Before positioning the routine against the business process, however, we look at a particular aspect that has been studied in the context of routines: *human agency*.

In the organisational sciences, the ontologies of human and non-human (e.g. material or technological) agency have been argued by scholars. Theoretical developments in this area have variously placed human intentionality at the centre and ignored technology as with structuration theory [12] or regarded the human and technology agencies as an inseparable duality, as with Actor-Network theory [13] or sociomateriality [14]. A common theme that arises is viewing social and technology elements in *relational* terms where behaviours are emergent from the interaction. This could be summarised as the technology-in-practice viewpoint [15]. This can be contrasted with the more orthodox IS perspective where roles, processes, technologies are characterised as stable, independent entities with simple unidirectional relationships [e.g. 16–18].

The specificity of the organisational routine to its context, as noted by Becker [19], influences its ability to be replicated and the inertia it generates within the organisation. In their influential work, Pentland and Feldman [11] distinguish the duality of “ostensive” and “performative” facets: the former representing the idealised, codified representation of the routine and the latter the routine-in-use, or what actually happens in practice. The implication here is that the routine may be performed differently each time it is repeated even if the ostensive aspect remains the same. This duality is represented schematically in **Fig. 1**. These authors note that the ostensive routine is *necessarily* an abstraction since it cannot fully specify all the detail required to perform the routine. Hence there is always an interpretative step required to get to the performative.

The idea of the performative routine is taken a step further into the socio-technical realm by the emerging concept of the technology affordance [5, 20–22]. An affordance represents the perception of what can be done with an item of technology by a user with a particular goal – i.e. the affordance is the *potentiality* for action of a

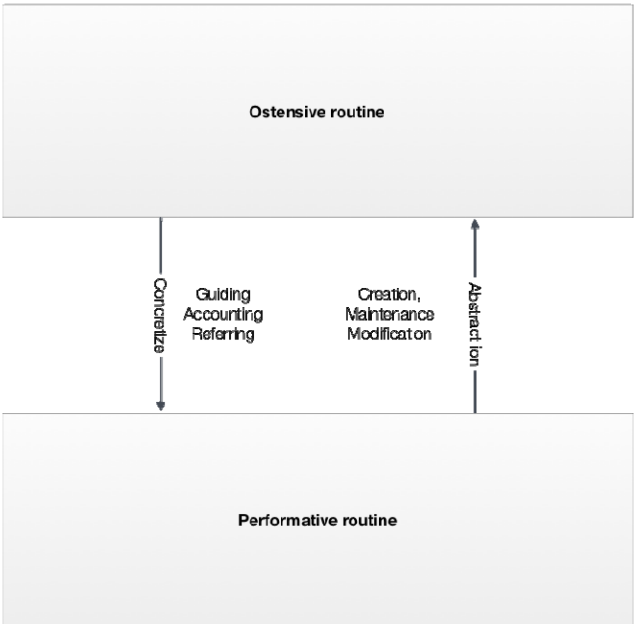


Fig. 1. Ostensive and Performative duality of routines (after Feldman and Pentland 2003)

technology feature, not necessarily how the feature was designed. According to Leonardi [21], the flexibility of organisational routines as well as technologies will determine how the affordance will be realised by virtue of the way the human and material agencies become "imbricated" or intertwined. In other words, the affordance (or constraint) posed by an item of technology may prompt a change to either the routine or the technology depending on its flexibility and on what has happened in the past.

3.3 Organisational Capabilities

Another well researched organisational building block is the organisational capability (OC). Winter defines an OC as "a high-level routine (or collection of routines) that together with its implementing input flows, confers upon an organization's management a set of decision options for producing significant outputs of a particular type" [23 p.991]. Dosi et al. [3] distinguish the *capability* from the *routine* on the basis of recognisable purpose. In their scheme, routines are repeatable units of activities, but a capability has a purpose or outcome that it is supposed to enable. Routines are thus the building blocks of capabilities, but would only satisfy the definition of being a capability themselves if the recognisable purpose is evident. Schreyogg et al. [24], in their definitional analysis, identify several higher order characteristics of OCs, such as representing collective organisational problem solving, combining explicit and tacit knowledge and being repeatable, reliable pattern of action. That these capabilities represent the product of organisational learning, and are what generates the value for the firm seem to be common across these viewpoints.

Like capability, the term "competency" (or "competence") has been used in a variety of ways both in specific technical and in more general senses. Core competency theory places them at a higher level of abstraction, being those valuable *capabilities* that specifically deliver *customer* benefit [25]. Consistent with this view is Volberda and Lewin [26], who place them at the top of a discrete three level hierarchy (with capabilities and routines occupying the other two levels) in their multi-level view of firm co-evolution. This view suggests that competencies are the "externally facing" view of the firm's capabilities – the ones that are important from a competitive marketplace point of view, whereas the other two are endogenous to the firm. McKelvey [27] uses the term collectively to include resources, capabilities and activities (per Porter's value chain). In the subsequent discussion, we similarly collapse the concept of competency into organisational capability.

A related concept is dynamic capability (DC). The initial definition of a DC was as the "firm's ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments" [28 p.516]. Helfat and Winter [29] distinguish DCs from OCs on the basis of the latter being associated with "earning a living" [23] using the current methods and techniques; whereas the DC being about changing the way things are done [30] . This places them at a meta-level, where DCs can modify OCs (e.g. through reconfiguration, as defined above), or in other words, they can change the way the firm earns its living when the environment requires it. This is consistent with Winter's [23] hierarchy of capability types. Interestingly, Trkman [31] characterises the continuous improvement of business processes (contingent on business environmental change) in terms of dynamic capabilities.

4 Implications for the Conceptualisation of Business Process

4.1 Routines

It would be a reasonable step to conceptually equate business processes to organisational routines on the basis that they share many of the same ontological elements, for example: sets of repeatable activities, the involvement of organisational participants and use of technology. However, we argue the distinction between the ostensive and performative aspects also has relevance to how business process should be conceptualised. It suggests that there has been a missing ingredient in the traditional IS orthodoxy when it comes to business processes [e.g. 1, 2, 32]. That is, we cannot treat a business process merely as an artefact that can be deterministically executed. Instead, as we argue, the non-determinacy of human agency must be factored in at two levels: firstly in the process-as-designed against the process-as-performed, noting that the latter aspect brings the human factors such as motivation, skills, tacit knowledge and experience which intrinsically means the process may not deliver what was "intended". Secondly, the way technology is used (by a human user) in the business process is similarly a function of the *potentiality* of the technology (for action) as perceived by the users, rather than just a set of pre-designed technology features.

For Business Process Management (BPM) systems [32], the ostensive/performative duality raises an issue. The automation of BPM is predicated on the notion of the "executability" of the process model, or in other words, "the model *is* the process" [33]. It would be tempting to equate the process model represented within a BPMS to the ostensive process definition, but this would be an oversimplification for all except fully automated processes. As noted by Feldman and Pentland [11], the ostensive includes any tacit procedural knowledge that may reside within the human actors, that contributes to how the routine is understood in the organisation. This means no matter how well we define and model the ostensive business process, there will be a performative dimension, one that requires (and delivers) "contextual flexibility", or in other words, flexibility that is bounded these contextual human agency factors.

4.2 Capabilities

The concept of organisational capability, as something that defines what an organisation does to earn its living, is a way of linking the business imperatives of the organisation (i.e. which capabilities it needs to deploy) to the work that is actually being done by the organisation (i.e. the routines or business processes). As such, we argue, it is a useful granular business alignment mechanism whereby the goals at the business process level can be aligned to the specific organisational capabilities they are enabling or delivering. This then allows the external competitive pressures being exerted on the organisation's capabilities to be linked to the work being done internally giving us two "fitness landscapes" that have to be balanced: the *fit* of the capabilities to the organisation's environment and the *fit* of the business processes to the capabilities. This follows the characterisation of external (evolutionary) fitness and internal (technical) fitness of capabilities by Helfat et al. [34]. We represent this idea in **Fig. 2** below.

For architectural context, the diagram also depicts the dynamic capabilities operating at a meta-level and across the layers in the model. These relationships correspond to the DC capacities of sensing, shaping and seizing opportunities offered by the rapidly changing environments [35] in terms of the "orchestration" of the organisational building blocks. Further treatment of DCs is outside the scope of this paper.

Of course, there will be many business processes that exist just to keep the organisational running rather than acting as competitive differentiators. These would include the "commoditised" back-office functions that all organisations possess. Nevertheless, the analysis of process goals and their alignment to organisational capabilities, this model suggests, is worthwhile for any business process.

We contrast this approach with that of Trkman [31], who provides a contingency theory-based approach to understanding the fit between the business processes and the needs of the business environment. This basically says understand your organisation's specific contingencies and align your BPM program to them. We argue that using organisational capabilities as a context provides a clearer, granular way of conceiving

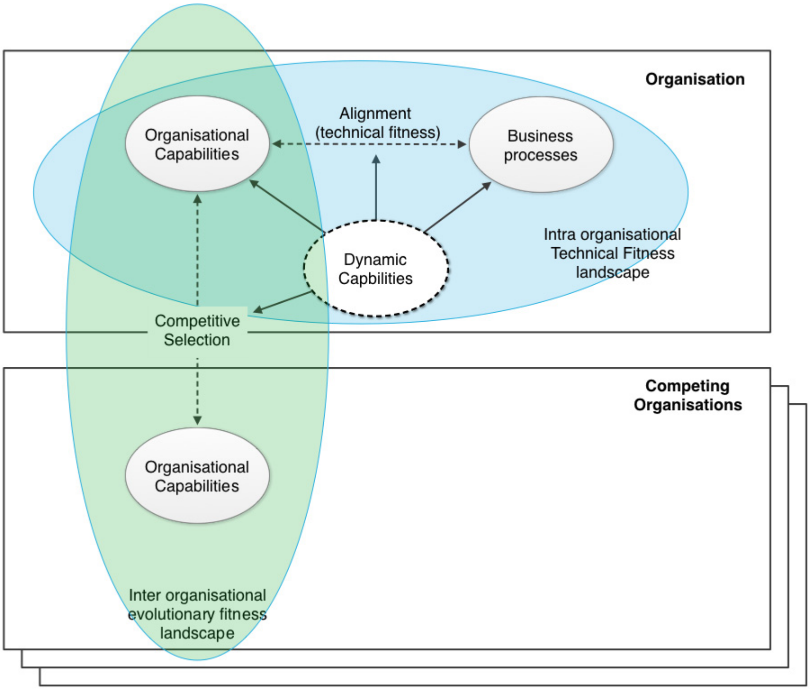


Fig. 2. Relationships between organisational building blocks

of the business' strategic requirements and thence how business processes should be aligned as building blocks of these capabilities.

5 Towards a Unified Model

The preceding discussion has drawn out two principles that are informing the ontological basis of our current research work. In this work we are seeking to develop a framework for understanding the role of business-IT alignment in the overall agility of the organisation, measured in terms of its ability to deploy capabilities in a dynamic business environment. These principles are: firstly, that the theories of organisational routines can be applied to how we conceptualise business process. In particular, that the business process "in-practice" is different from the business process "as-designed", due to the non-determinacy of the socio-material interactions. Secondly, that the relationship of an organisation's business processes to the core capabilities that define the organisation's competitive differentiators, is a key component of internal organisational alignment.

In **Fig. 3** we juxtapose the concepts discussed into our nascent conceptual model. The model builds on the fundamental relationships represented in **Fig. 2**. The numbered labels on the diagram are explained below.

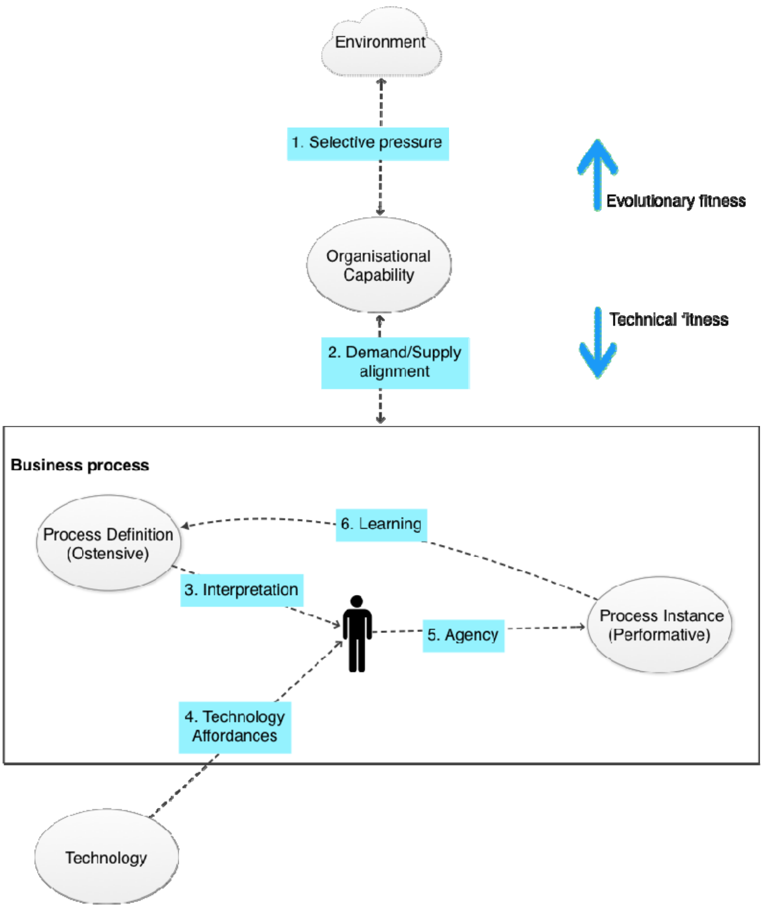


Fig. 3. Conceptual Model

1. Selective Pressure. This relationship represents the requirement for "evolutionary fitness" [34] on the part of the organisation's capabilities. The environment selects, through competitive pressure, which capabilities are required – and hence the "value" of each capability at a point in time. So, for example, a capability such as "manufacture cars" has an evolutionary fitness that is a function of the market demand and the competitors' products.
2. Demand/Supply Alignment. This is the central organisational alignment relationship whereby the external demand pressure for a given capability is met (or not) by the supply side: or in other words the ability of the organisation's business processes to deliver such a capability. This relationship determines the "technical fitness" of the capability. So for the "manufacture cars" capability, this represents the knowledge, skills, tacit knowledge, and resources bound up in the business processes that can deliver that capability as an outcome. Technical fitness is

decoupled[1] from evolutionary fitness in this model, reflecting an independent set of drivers that are endogenous to the organisation. For example, technical fitness could be measured in terms of efficiency or cost per unit output [34].

3. Interpretation. This is the interpretation required on the part of the human agent of the ostensive business process in order to actually perform it. Following Feldman and Pentland's application of this concept to organisational routines [4] this structural, idealised aspect of the business process is interpreted each time it is performed based on the context. This creates the opportunity for variation and allows contextual flexibility [4, 11].
4. Technology Affordances. This represents the socio-technical relationship whereby the features available in the technology are interpreted by the user into a set of "affordances" [5]. These affordances are the product of the user's particular goals, experience and skills providing a unique context for how the technology features (as designed) are perceived as part of the business process [20].
5. Agency. This recognises the human agency that actually causes the business processes to be performed. Agency in this sense is "something that produces an effect" [12] or in other words action.
6. Learning. This is a feedback loop whereby the ostensive aspect of the business process is realigned based on the performative experience. This is consistent with the Feldman and Pentland's original characterisation of routines [11] and their more recent work on modelling this experiential learning [36].

6 Conclusion and Further Work

In taking a cross-disciplinary approach to understanding the role of technology within the modern organisation, we have sought to use some theories of the organisation to inform a more traditional IS perspective. In particular, in this paper, we have presented a conceptual model that attempts to position related concepts such as business process, organisational capability and technology affordance into a new consistent ontological framework.

While this model, as depicted in **Fig. 2** and **Fig. 3**, is still under development, and research is ongoing, there is the potential for several contributions to business process theory. Firstly, it provides a richer ontology for the concept of business process by differentiating the process as-designed from the process as-performed and including the non-determinacy of the socio-technical elements. Secondly, it positions concept of business process within a broader organisational context that provides an opportunity to reason about alignment of processes to the overall strategic capability needs of the organisation. It presents a granular model of the organisation that draws a conceptual connection from the external organisational pressures to adapt, the alignment of business process to delivering the required capabilities, through to how technology is used within those processes.

[1] Potential interaction between evolutionary and technical fitness is an opportunity for further study.

The model has some implications for business process modelling and design. It suggests that the representation of the ostensive business process – or the formal business process model – is only part of the story. There is an adaptive loop at play in **Fig. 3** (links 3 and 6) whereby the ostensive process is interpreted as it is used and then updated as the organisation learns from experience. This invokes the Melão and Pidd [37] perspective of business processes as complex dynamic systems, rather than as static entities that are only designed once. Indeed, co-evolutionary theories of the organisation, whereby organisational building blocks are understood and modelled in terms of "species" in co-evolutionary relationships, present a rich paradigm with which to explore this area. They have been applied at the organisational level [26], the capability level [26, 27] and at the process level [33]. Accommodation of these theories into our overall conceptual model is one avenue of research we are pursuing, as we seek to understand organisational adaptation to change in terms of these granular concepts.

References

1. Hammer, M., Champy, J.: Reengineering the Corporation: Manifesto for Business Revolution. Harper Business (1993)
2. Weske, M.: Business Process Management - Concepts Languages Architectures. Springer, New York (2012)
3. Dosi, G., Nelson, R., Winter, S.: Introduction: The nature and dynamics of organizational capabilities. In: Dosi, G., Nelson, R., Winter, S. (eds.) The Nature and Dynamics of Organizational Capabilities, pp. 1–22. Oxford University Press, Oxford (2000)
4. Pentland, B., Feldman, M.: Organizational routines as a unit of analysis. Ind. Corp. Chang. 14, 793–815 (2005)
5. Zammuto, R.F., Griffith, T.L., Majchrzak, A., Dougherty, D.J., Faraj, S.: Information technology and the changing fabric of organization. Organ. Sci. 18, 749–762 (2007)
6. Boisot, M., McKelvey, B.: Integrating modernist and postmodernist perspectives on organizations: A complexity science bridge. Acad. Manag. Rev. 35, 415–433 (2010)
7. Dubé, L., Paré, G.: Rigor in information systems positivist case research: Current practices, trends, and recommendations. MIS Q. 27, 597–635 (2003)
8. Tsang, E.W.K.: Case studies and generalization in information systems research: A critical realist perspective. J. Strateg. Inf. Syst., 1–13 (2013)
9. Volkoff, O., Strong, D., Elmes, M.: Technological Embeddedness and Organizational Change. Organ. Sci. 18, 832–848 (2007)
10. Gregor, S.: The Nature of Theory in Information Systems. MIS Q. 30, 611–642 (2006)
11. Feldman, M., Pentland, B.: Reconceptualizing Organizational Routines as a Source of Flexibility and Change. Adm. Sci. Q. 48, 94–118 (2003)
12. Giddens, A.: The constitution of society. University of California Press, Berkeley (1984)
13. Latour, B.: Science in Action. Harvard University Press, Boston (1987)
14. Orlikowski, W., Scott, S.: Sociomateriality: Challenging the Separation of Technology, Work and Organization. Acad. Manag. Ann. 2, 433–474 (2008)
15. Feldman, M., Orlikowski, W.: Theorizing practice and practicing theory. Organ. Sci. 22, 1240–1253 (2011)
16. Tallon, P., Pinsonneault, A.: Competing perspectives on the link between strategic information technology alignment and organizational agility: Insights from a mediation model. MIS Q. 35, 463–486 (2011)

17. Melville, N., Kraemer, K., Tanniru, M.: Review: Information technology and organizational performance: An integrative model of IT business value. MIS Q. 28, 283–322 (2004)
18. Henderson, J., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Syst. J. 32, 4–16 (1993)
19. Becker, M.: Organizational routines: A review of the literature. Ind. Corp. Chang. 13, 643–678 (2004)
20. Majchrzak, A., Markus, M.: Technology affordances and constraints. In: Kessler, E. (ed.) Encyclopedia of Management Theory. Sage (2012)
21. Leonardi, P.: When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. MIS Q. 35, 147–167 (2011)
22. Yoo, Y., Boland, R.: Organizing for innovation in the digitized world. Organ. Sci. 23, 1398–1408 (2012)
23. Winter, S.: Understanding Dynamic Capabilities. Strateg. Manag. J. 24, 991–995 (2003)
24. Schreyogg, G., Kliesch-eberl, M.: How dynamic can organizational capabilities be? Towards a dual-process model of capability dynamization. Strateg. Manag. J. 28, 913–933 (2007)
25. Hafeez, K., Malak, N.: Core competence for sustainable competitive advantage: A structured methodology for identifying core competence. IEEE Trans. Eng. Manag. 49, 28–35 (2002)
26. Volberda, H., Lewin, A.: Co-evolutionary Dynamics Within and Between Firms: From Evolution to Co-evolution. J. Manag. Stud. 40, 2111–2135 (2003)
27. McKelvey, B.: Avoiding complexity catastrophe in coevolutionary pockets: Strategies for rugged landscapes. Organ. Sci. 10, 294–321 (1999)
28. Teece, D., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. Strateg. Manag. J. 18, 509–533 (1997)
29. Helfat, C., Winter, S.: Untangling dynamic and operational capabilities: Strategy for the (n)ever-changing world. Strateg. Manag. J. 32, 1243–1250 (2011)
30. Augier, M., Teece, D.: Dynamic Capabilities and the Role of Managers in Business Strategy and Economic Performance. Organ. Sci. 20, 410–421 (2009)
31. Trkman, P.: The critical success factors of business process management. Int. J. Inf. Manage. 30, 125–134 (2010)
32. 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., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
33. Vidgen, R., Wang, X.: From business process management to business process ecosystem. J. Inf. Technol. 21, 262–271 (2006)
34. Helfat, C., Finkelstein, S., Mitchell, W., Peteraf, M., Singh, H., Teece, D., Winter, S.: Dynamic capabilities: Understanding strategic change in organizations. Blackwell, Oxford (2007)
35. Teece, D.: Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. Strateg. Manag. J. 1350, 1319–1350 (2007)
36. Pentland, B., Feldman, M., Becker, M., Liu, P.: Dynamics of organizational routines: A generative model. J. Manag. Stud. 49, 1484–1508 (2012)
37. Melão, N., Pidd, M.: A conceptual framework for understanding business processes and business process modelling. Inf. Syst. J. 10, 105–129 (2000)

A Data-Centric Approach for Business Process Improvement Based on Decision Theory

Ekaterina Bazhenova and Mathias Weske

Hasso Plattner Institute at the University of Potsdam
{ekaterina.bazhenova,mathias.weske}@hpi.uni-potsdam.de

Abstract. An efficient business process redesign is an ambitious research and implementation challenge for both academia and industry. Traditional approaches for business process improvement are based on activity flows, not considering data of business processes. In this paper, we provide an approach to business process improvement, which is based on data and on combining data with decision theory. In particular, subprocesses are formalized as decision activities and analyzed according to techniques from decision theory. We demonstrate the applicability of our research with a use case, where meetings in an enterprise are scheduled.

Keywords: business process improvement, redesign, decision subprocess, process data.

1 Introduction

The prerequisite of successful existence of the enterprise of today is effective business process management. In consequence of technological progress in the last decades, organizations have received not only vast opportunities for the optimization of business processes, but also daunting challenges with regards to applying these innovations in real businesses. With that, the question of how to re-organize the business process in order to use new technologies, represents the challenge of business process redesign which "is often not approached in a systematic way, but rather considered as a purely creative activity" [4].

The majority of existing approaches to business process redesign are activity-centric and they do not consider process model data. However, data-centric approach to modeling business operations and processes "has been evidenced in both academic and industrial researches where it not only provides higher level of flexibility of workflow enactment and evolution, but also facilitates the process of business transformations" [10].

Other factors, which influence the application of business process management in enterprises, are the instability of markets and the necessity of making decisions under the conditions of risk and uncertainty. Even a simple business process, such as scheduling meetings at an enterprise, can have different execution outcomes depending on, for example, the time preferences of the customer. Due to technological development, centralized, calendar-oriented software for

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 242–256, 2014.

scheduling meetings is available, which can potentially improve the business process of time management [5]. However, the methodology of redesigning such a business process, considering both the internal structure of the process, and uncertainties of the external environment, does not exist.

The above mentioned factors served as the prerequisite for the development of a methodology for data-centric business process improvement based on the application of decision theory, which we present in this paper. Our fundamental contribution is a presentation of the integrated methodology for the identification of patterns for redesign in process models, redesign guidelines and introduction of process indicators which will allow the effectiveness of the redesigned models to be monitored.

The remainder of the paper is structured as follows. In Section 2 the notions of process models, data and the foundations of decision theory used in our approach are presented. In Section 3 we introduce a special kind of process model, a decision subprocess, which serves as a redesign pattern. Additionally, we present a transformation rule for improvement of such a process model. Section 4 demonstrates the applicability of the developed scheme with a use case, where meetings in an enterprise are scheduled. The related work is then provided in Section 5. Finally, the paper is concluded.

2 Preliminaries

The generic scheme of our approach for business process redesign is presented in Figure 1, the detailed version of which can be found in our previous paper [2].

The first step is to identify if the initial process model P contains patterns for redesign. The example of such a pattern, a decision subprocess, is presented in Section 3. If it is detected that the process model contains such patterns, the transformation of the process model is implemented as the second step of the redesign scheme, which will be explained in detail in Section 3. This transformation yields, as an outcome, an improved process model P'. To verify the effectiveness of the transformation, the third step of the redesign scheme simulates the execution of the improved process model P' with the usage of the key performance indicators, the development of which is planned for future work.

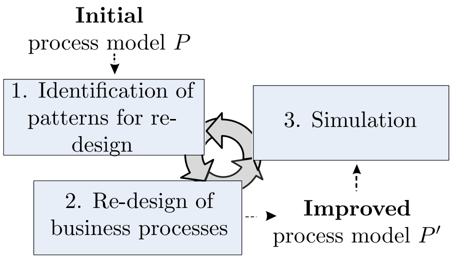


Fig. 1. Scheme for business process improvement

Depending on the simulation results, a conclusion is made, either to accept the improved process model P' and start using it in the enterprise, or to conduct further improvements of the process model. Such a decision can be done, for example, by a business analyst or higher management.

2.1 Process Model and Data

The input and output for our redesign scheme are process models, which can be viewed as blueprints for a set of process instances with a similar structure [15].

Definition 1 (Process model). $P = (N, E, D, F, R, \psi, \gamma)$ is a *process model* if it consists of a finite non-empty set N of nodes, and a finite set E of edges. Herewith, $N = N_A \cup N_E \cup N_G$ is a union of the mutually disjoint sets N_A (an nonempty set of activities), N_E (a set of events), and N_G (a set of gateways). With that, E is a set of directed edges between nodes, such that $E \subseteq N \times N$, representing control flow. Further, F is a set of edges representing data flow relations: $F \subseteq (N_A \times D) \cup (D \times N_A)$. R is a set of resources. $\psi : N_A \longrightarrow R$ is a function assigning to each activity a corresponding resource. $\gamma : N_G \longrightarrow \{xor, and\}$ is a function assigning to each gateway a corresponding control flow construct. ◇

In Definition 1, we take into account the resources which are involved in the execution of a business process. It is also assumed in the definition, that the activities of process models operate on an integrated set D of data nodes, which represent application data, created, modified, and deleted during the execution of a process model. The term data flow refers to data dependencies between process activities and data.

In our work we use the distinction of process data into data classes and data nodes (see Figure 2), which can be viewed as analogous to the object-oriented programming paradigm. *Data class*, used in a process model, serves as an abstract data type, which describes the properties of *data nodes*. The data nodes can be viewed as instances of the data classes at the modeling level. Data nodes are associated with exactly one data class in a process model, in a way that the particular values of data class properties are assigned to the data node associated with it.

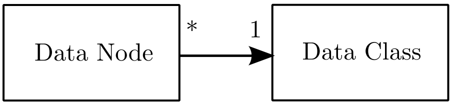


Fig. 2. Relations between data entitites

Definition 2 (Data class). *Data class* $D_c = (name, S, Q_c)$ is a tuple, where:

- $name$ is a constant which serves as a unique identifier for the data class D_c;
- S is a finite non-empty set of data states;
- Q_c is a finite set of attributes, which are properties representing data fields containing values of an arbitrary type. ◇

Definition 3 (Data node). Let D_c be a data class, used in a process model. A tuple $D_n = (name, s, \delta, \tau, \varphi, Q)$ is a *data node*, related to the corresponding data class D_c, with the following parameters:

- $name$ is a constant labeling data node D_n, which serves as a reference to the corresponding data class D_c;

- $s \in S$ is a variable reflecting the state assigned to D_n, where S is the set of data states of D_c;
- $\delta : D_n \longrightarrow \{singlinst, multinst\}$ is a function indicating if the data is a collection (singlinst) or not (multinst);
- $\tau : D_n \longrightarrow \{input; output; default\}$ is a function indicating if D_n is an input data node (existed before the start of the process), ouput data node (will exist after termination of the process) or none of these (default);
- $\varphi : D_n \longrightarrow R$ is a function indicating the resource allocated for D_n;
- $Q \in Q_c$ is a set of attributes assigned to D_n, where Q_c is the set of attributes of D_c. ◇

To be definite, we assume that the resource of a process model, allocated for the data node, is the same as the resource allocated to the activity, which accesses this data node. Thus, the value of the function ψ (from Definition 1), mapping the activity a to the resource R , is equal to the value of the function φ (from Definition 3), mapping the data node D_n, with which a is in a data flow relation, to the same resource R. More specifically, $\varphi(D_n) = \psi(a)$, where $a \in N_A$, and $(a, D_n) \in F \vee (D_n, a) \in F$. Also, as it can be seen from Definition 3, the set of attributes Q store the context data relevant to the business process, i.e. the particular characteristics of the data class.

2.2 Definitions from Decision Theory

As it was mentioned in the introduction, many business processes face the uncertainties of the business environment and decision theory is a tool which is focused on dealing with such challenges. Below we provide the notions used in our approach, with regards to the foundations of decision theory [9,11].

The core setting of decision theory is an occurrence of a subject *decision maker* whose aim is to make an optimal choice between a set of n *alternatives*: $X = \{x_i\}, i = 1, \ldots, n$, with a possible *outcome* event O. The main assumption is that any realization of the alternatives resulting from a decision can be compared, which is described by the *preference relations* of the decision makers, represented by the $\succcurlyeq$ sign.

Definition 4 (Preference relation). A *preference relation* $\succcurlyeq$ is a subset of the binary relation $X \times X$, that satisfies two principles :

1. *Completeness.* $\forall x_i, x_j \in X$: either $x_i \succcurlyeq x_j$, or $x_j \succcurlyeq x_i$, or both.
2. *Transitivity.* $\forall x_i, x_j, x_k \in X$: if $x_i \succcurlyeq x_j$ and $x_j \succcurlyeq x_k$ then $x_i \succcurlyeq x_k$. ◇

Definition 5 (Lottery). A *lottery* L is a finite vector $(p_1, \ldots, p_n)$, where p_i is the probability that the alternative x_i will be realized, such that $\sum_{i=1}^{n} p_i = 1$, $p_i \geq 0$. ◇

Another assumption of decision theory is that a decision maker is making a choice in a rational way, which is expressed by a *utility function* assigned to the decision maker.

Definition 6 (Utility Function). A *utility function* u is a function which assigns a real number to any given choice of the alternatives, $u : X \longrightarrow \mathbb{R}$ where $\mathbb{R}$ is a set of real numbers. A utility function u is said to represent a *preference relation* $\succcurlyeq$ if and only if $\forall x_i \in X, \forall x_j \in X, u(x_i) \geq u(x_j) \Leftrightarrow x_i \succcurlyeq x_j$ ◇

The value of the utility function is a *payoff*. For comparing the alternatives in a decision making process, a notion of *expected payoff* is used:

Definition 7 (Expected Payoff of the Lottery). An *expected payoff* E of the lottery is the average of payoffs which the decision maker gets from the assumed realization of the alternative, weighted by the probability of such a realization: $E(L) := \sum_{i=1}^{n} p_i u(x_i)$ ◇

In terms of the introduced definitions, the assumption of rational behavior is the following: the goal of each decision maker is to maximize the expected payoff of the lottery.

3 Redesign of the Decision Subprocess

Searching for ways to improve business processes led us to consider the typical challenges of the business environment, such as turbulence of markets and making decisions under conditions of risks and limited resources. In order to provide an effective mechanism for dealing with the uncertainties in business environment, in this section we provide a mapping between the decision theory and business process management, and devise how to use it for the business process redesign.

3.1 Process Model as a Decision Subprocess

The notions of decision theory, presented in Section 2, provides the premise for defining a special kind of business process models, which we refer to as *decision subprocesses.*

The generic structure of a decision subprocess is shown in Figure 3. The decision subprocess represents a process model, the internal logic of which is hidden inside the collapsed subprocess. As it can be seen from the figure, the set of alternatives in the decision subprocess is presented as the collection input data node D_n^i, and the final decision is presented as the collection output data node D_n^o, so that $\tau^i = \{input\}$, $\tau^o = \{output\}$. The decision subprocess should reflect the process of decision making, therefore it is assumed that the data represented by the output data

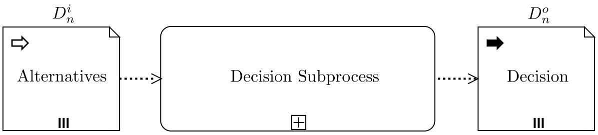


Fig. 3. Structure of a decision subprocess

node "Decision" is a subset of the data represented by the input data node "Alternatives".

Based on the above mentioned considerations, the decision subprocesses can be defined formally, based on following conditions. Let P be a process model, which consists of K data nodes, including the input data node D_n^i and the output data node D_n^o , which are bound to J data classes.

Condition 1. Set of alternatives is represented by the set of attributes Q^i of the input data node D_n^i.

Condition 2. Final decision is represented by the set of attributes Q^o of the output data node D_n^o. The set of attributes Q_c^o of data class D_c^o, which is assigned to the *output* data node D_n^o, is a subset of the set of attributes Q_c^i of data class D_c^i, which is assigned to the *input* data node D_n^i: $Q_c^o \subseteq Q_c^i$.

Condition 3. Decision makers are represented by a function φ, indicating resources allocated for data nodes D_n (see Definition 3).

Condition 4. Decision making process consists of decision makers choosing alternatives, so that each set of attributes Q of data class D_c assigned correspondingly to any data node D_n is a subset of the set of attributes Q_c^i of data class D_c^i assigned to the input data node D_n^i: $\forall Q_c : Q_c \subseteq Q_c^i$.

Definition 8 (Decision subprocess). If a given process model P satisfies conditions 1-4, then such a process model represents a *decision subprocess*. ◇

3.2 Scheme of Business Process Improvement

The introduction of the decision subprocess enables us to suggest an approach for the improvement of such a process model. Below we present the detailed approach, which consists of three consequent phases, corresponding to three stages of the scheme for business process improvement (see Figure 1):

S1 (a). Analysis of Business Process Model. The business process improvement scheme is launched when a business analyst of the enterprise decides that the current business process is not efficient.

S1 (b). Detection of Decision Subprocess. It is identified if the current process model P represents a decision subprocess, according to Definition 8.

S2 (a). Definition of Payoff Function. The improvement of the internal structure of the decision subprocess (i.e., the collapsed subprocess in Figure 3) can be done by the application of the decision theory methods. The persons, or other resources, involved in the execution of the decision subprocess, can be viewed as decision makers. Additionally, according to the assumption of rational behavior of decision makers, their goal is to maximize the *expected payoff* for the decision subprocess. Therefore, the assigned goal of this stage is to set the payoff function of the decision subprocess. The example of the payoff function could be the time saved by participants, to agree on the decision.

S2 (b). Optimization of Decision Subprocess. In such a way, we reduced the challenge of business process improvement to the task of maximizing the expected payoff for the decision subprocess. To solve this task, we propose the following transformation, which consists of two steps:

1. All the data classes D_c of the decision subprocess are consolidated into one data class $D_c^{'}$. Such transformation preserves the business context of the process model, as, according to Condition 4 of Definition 8, each set of attributes of any data class in a decision subprocess is a subset of the set of attributes of the data class assigned to the input data node.
2. The access management of resources is changed in such a way that within the decision subprocess, all the resources should have access to all the data nodes assigned to the consolidated data class $D_c^{'}$.

The output of such a transformation is a process model P', which is different from the initial process model P only in a way, that it contains a set of K data nodes $D_n^{'}$, all of which are assigned to one consolidated data class $D_c^{'} = (name', S', Q_c^{'})$, where

– $name'$ reflects the consolidated nature of the data class,. $name'$ can be assigned by a business analyst;
– $S' = \{S_j\}, j = 1, \dots, J$ is the set of states retrieved as a maximal subset of the sets of states of data classes $D_c^j, j = 1, \dots, J$ assigned to the initial process model P;
– $Q_c^{'}$ is the consolidated set of attributes retrieved as a maximal subset of the sets of attributes for all data classes in the initial process model P.

The data nodes $D_n^{'}$ of the transformed model P' are different from the corresponding data nodes D_n of the initial model P only in a way, that the value of the parameter $name'$ for each data node $D_n^{'}$ is equal to the value of the corresponding parameter of the consolidated data class $D_c^{'}$.

S3. Simulation of Redesigned Process Model. In order to assess the efficiency of the transformation, we plan to develop a set of indicators and conduct a simulation of the process model for estimating the values of these indicators. This is the final step of the improvement scheme. Depending on the results of the simulation, a conclusion is made, to either accept the improved process model P' and start using it in the enterprise, or to conduct further improvements of the process model. Such a decision can be done, for example, by a business analyst or higher management.

4 Use Case

In this section, we demonstrate the applicability of our approach to business process improvement with a use case, which incorporates the decision making process.

4.1 Setting of Use Case

Context Statement. *In the enterprise, there is a group of people* $\{person_j\}$, $j = 1, \ldots, N_{people}$ *for whom a meeting should be organized. The meeting should be held on a specific date, with a minimum number of people* N_{minp} *participating in it. A preliminary set of dates* $\{date_i\}, i = 1, \ldots, N_{dates}$ *is given, out of which each participant should choose one.*

We investigate two possible scenarios for the realization of this process:

1. The organization of the scheduling of meetings is being done by a secretary who writes a personal e-mail to every participant of the meeting, collects the responses, selects the date, for which the majority of participants have voted, and sends it back to participants for confirmation. If less than the required minimum N_{minp} of people confirm their participation, the process repeats. If more than N_{minp} people confirm this date, the secretary fixes it and sends a final e-mail to all the participants with the chosen date. This scenario represents the case of a so called closed scheduling system, where the participants make decisions without knowing each others choices.
2. The second scenario considers the scheduling of the meeting date with the help of a software platform, which serves as an agent, collecting the opinions of participants. An example of such a platform could be the online scheduling platform "Doodle" [1]. Such an approach represents the case of a so called open scheduling system, where the participants make choices, knowing each others choices.

4.2 Application of Scheme of Business Process Redesign

The application of our step-by-step approach for the improvement of business process for the use case, described in Section 3, is discussed next.

S1 (a). Analysis of Business Process Model. We assume, that a closed scheduling system is used in an enterprise (first scenario). After reviewing the context of this scenario, the business analyst comes to a conclusion that the scheduling of a meeting by a secretary involves a large number of created data artifacts (e-mails).

S1 (b). Detection of Decision Task. As the goal of the business process is to choose one final date for a meeting, it can therefore be considered as a decision subprocess. The formal mapping of the notions from the decision theory is presented in Table 1.

As shown in the table, the decision makers are the participants of the scheduling business process. With that, the set of alternatives is a set of dates, from which one date should be chosen as a final date for the meeting. Thus, the choice of alternatives can be represented by the following set of trials:

Table 1. Definitions from decision theory and corresponding elements of a business process in the scheduling use case

Definitions of Decision Theory	Corresponding Elements of the Use Case
Decision makers	Set of participants for the business process $\{person_j\}, j = 1, \ldots, N_{people}$
Set of alternatives	Set of dates $\{date_i\}, i = 1, \ldots, N_{dates}$
Choice of alternatives	Set of trials which represent the voting of participants $I = \{I_k\}, k = 1, ..., N_{people}$
Outcome	Event S_i / Event F_i - Success or Failure of $date_i$, $i = 1, ..., N_{dates}$
Utility function	$U(S_i) = 1$ and $U(F_i) = 0, i = 1, ..., N_{dates}$

Event $I_j \longrightarrow$ The first person from the group of participants $p_1 \in \{person_j\}, j = 1, ..., N_{people}$ **accepted** the date $date_j$

Event $\overline{I_j} \longrightarrow$ The first person from the group of participants $p_1 \in \{person_j\}, j = 1, ..., N_{people}$ **rejected** the date $date_j$

Furthermore, two outcomes for each alternative date are possible:

Event $S_i \longrightarrow$ "Success", each participant made a choice, and **not less** then N_{minp} voted for the $date_i, i = 1, ..., N_{dates}$

Event $F_i \longrightarrow$ "Failure", each participant made a choice, and **less** then N_{minp} voted for the $date_i, i = 1, ..., N_{dates}$

According to the logic of the business process, each participant will prefer at most that the meeting **will** take place, at any date. Therefore we assign the following values to the utility function: $U(S_i) = 1$ and $U(F_i) = 0, i = 1, ..., N_{dates}$, as presented in Table 1.

In such a way, the business analyst can come to the conclusion that the business process of scheduling the meeting at the enterprise, with the help of a secretary represents the decision subprocess.

S2 (a). Definition of Payoff Function. Recall, that at this stage of the scheme for business process improvement, the payoff function of the decision subprocess should be identified. From the second scenario it is known that the potential improvement of the scheduling business process can be provided by special software, which provides the participants with the possibility to view the choices of each other. Therefore, we propose to view the payoff function as an expected payoff of the choice of participants. Below we provide the comparison of the expected payoff of a choice in the general case and in both scenarios of closed and open scheduling systems.

General Formula for Expected Payoff of the Choice. According to Definition 7, the expected payoff for the participant from choosing the date is the following: $E(date) = P(S_i) * U(S_i) + P(F_i) * U(F_i)$. We showed in Table 1, that $U(S_i) = 0$ and $U(F_i) = 0$, therefore the equation for expected payoff is simplified as follows:

$$E = P(S) \tag{1}$$

When a participant chooses if he accepts a particular date, the answer is either "Yes" or "No". In such a way, we can view the sequence of the decisions made in the use case as a finite sequence of binary random variables with two possible outcomes: 0 or 1. Such process represents the *Bernoulli process* [3]:

Definition 9 (Bernoulli Process). The *Bernoulli process* is a sequence X_1, X_2, ... of independent random variables X_i, such that $P(X_i = 1) = P\{\text{success at the i-th trial}\} = p$, and $P(X_i = 0) = P\{\text{failure at the i-th trial}\} = 1 - p$, for each i. ◇

For the Bernoulli process, the formula of success in n trials, not less than k_1 times, and not more than k_2 times, is the following:

$$P\{k_1 \leq k \leq k_2\} = \sum_{k=k_1}^{k_2} C_n^k p^k q^{n-k} \tag{2}$$

Here, p and $q = 1 - p$ are the corresponding possibilities of success and failure of trials. It is assumed, that the choices of the people are random, so that the probability, that a participant will accept or reject the date, is equal: $P(I_1) = P(\overline{I_j}) = 0.5, j = 1, \ldots, N_{people}$. Therefore, according to Formula 2, $p = q = 0.5$. This formula is applicable in our use case for calculating the probability of the outcome $S_i, i = 1, ..., N_{dates}$.

Expected payoff of the choice in the scenario of a closed scheduling system. In this scenario, on any step of the decision subprocess, represented by trials $I = \{I_k\}, k = 1, ..., N_{people}$, the estimation by the participant of the probability of the success outcome for a particular date is always the same and can be calculated by Formula 2:

$$E(date) = P\{N_{minp} \leq k \leq N_{people}\} = \sum_{k=N_{minp}}^{N_{people}} C_{N_{people}}^k p^k q^{N_{people}-k} \tag{3}$$

Expected payoff of the choice in the scenario of an open scheduling system. Assume that only the first participant made a choice (trial I_1 or $\overline{I_1}$ was realized). As this participant does not know the preferences of others, he evaluates the probability of the final success event in the same way, as in the secretary scenario, using Formula 3. Now, when the second participant chooses a date (trial I_2 or $\overline{I_2}$), if he knows the choice of the first participant (trial I_1 or $\overline{I_1}$), his evaluation of the outcome can be estimated by considering the conditional probability of the final event in formula 2: $E(date) = (S|I_1)$. Thus, the required number of the participants for choosing a particular date, is less by 1 in formula 2:

$$E(date) = P\{N_{minp} - 1 \leq k \leq N_{people}\} = \sum_{k=N_{minp}-1}^{N_{people}} C_{N_{people}}^k p^k q^{N_{people}-k} \tag{4}$$

Formula 3 and Formula 4 are also applicable, if the participant is given a set of dates to choose: $\{date_i\}, i = 1, ..., N_{dates}$. In such a case, the participant evaluates all the conditional probabilities of the success of dates, taking into account the choices of the previous participants.

Comparison of expected payoffs of the choices in two scenarios. The comparison of Formula 3 and Formula 4 leads us to the conclusion that two options are possible. In the case that the event I_1 is realized, the expected payoff of the choice for the second participant in the second ("Doodle") scenario is greater, than in the first ("Secretary") scenario, by a positive summand added to the positive sum. If the event $\overline{I_1}$ is realized, the expected payoff is equal in both scenarios. According to the assumption of rational behavior of the decision maker, the second participant will make such a choice, which maximizes his expected payoff. Thus, he will more likely choose the date which was already chosen by other participants. This will increase the probability of a particular date to be chosen by the third participant and so on. The overall benefit for the whole process will be the decreasing in the time spent on decision making and, therefore, raising the effectiveness of the whole business process.

Numerical Example. Assume that 4 workers in the company are required to organize a meeting and they have to choose among three meeting dates, so that: $N_{people} = 4$; $\{date_i\}, i = 1, 2, 3$; $N_{minp} = 3$. The two scenarios can be presented as follows.

"Secretary" scenario. In this scenario, when the participants make the choice of dates, their estimation of the probability of the success for a date can be done using Formula 2:

$$P(S) = \sum_{k=3}^{4} C_4^k * 0.5^4 = C_4^3 * 0.5^4 + C_4^4 * 0.5^4 = 1/16 * (4 + 1) = 0.3125 \quad (5)$$

This probability stays the same at any step of choosing the dates by participants, because the participants do not receive any additional information which can influence their decision.

"Doodle" scenario. Assume that 2 people (Adam and Bob) have made choices according to Figure 4:

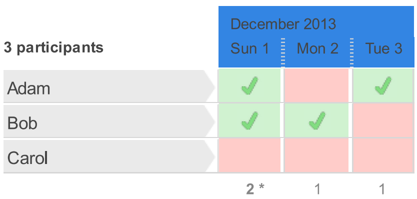


Fig. 4. Carol needs to make a choice

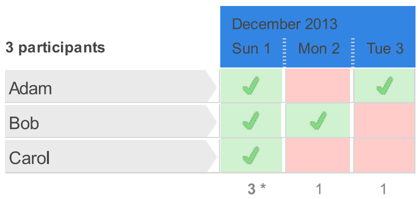


Fig. 5. Carol made a choice

Now assume that the third worker, Carol, needs to make a choice. For her, the probabilities of dates to be chosen, estimated by Formula 2, are the following:

$$\text{“Sun 1”} \longrightarrow P(S) = \sum_{k=1}^{4} C_4^k * 0.5^4 = 0.5^4 * (C_4^1 + C_4^2 + C_4^3 + C_4^4) = 0.9375 \quad (6)$$

$$\text{“Mon 2”} \longrightarrow P(S) = \sum_{k=2}^{4} C_4^k * 0.5^4 = 0.5^4 * (C_4^2 + C_4^3 + C_4^4) = 0.6875 \quad (7)$$

$$\text{“Tue 3”} \longrightarrow P(S) = \sum_{k=2}^{4} C_4^k * 0.5^4 = 0.5^4 * (C_4^2 + C_4^3 + C_4^4) = 0.6875 \quad (8)$$

We assume that Carol was initially hesitating between “Sun 1” and “Tue 3”. By looking at the Doodle poll (see Figure 4), she estimates that the possibility that the date “Sun 1” will be chosen is greater than for the date “Tue 3” (from Equations 6 and 8: $0.9375 > 0.6875$). She chooses the first date, and the meeting date is found, since three people have voted for the date (see Figure 5).

Simultaneously, if the system would be closed and Carol could only guess which decisions the other participants have made, the probability of success for all three dates from her point of view would be the same and its value could be calculated according to Equation 6. The difference in the expected payoffs of Equation 6 and Equation 5 is Carol's benefit for using the open scheduling system or *expected utility of knowing additional information* and its value is equal to $0.9375 - 0.3125 = 0.625$.

In the examples presented above we have demonstrated, with the help of decision theory, that for the scheduling decision subprocess, the open scheduling business process is more efficient than the closed scheduling business process. In the following subsection we present the possible transformation for the use case scenario of the closed scheduling system.

S2 (b). Optimization of the Decision Task. In order to implement the transformation rule, in this subsection we provide the simplified view at the process model in the case of a closed scheduling system, presented in Figure 6.

Thus, the process model consists of the following data classes with corresponding sets of parametres, as shown in Table 2. The presented use case satisfies the conditions of the decision subprocess with the following parametres:

Table 2. Data classes and corresponding sets of parametres in the use case

Data Class	**Corresponding Set of Parametres**
table with dates	$Q_1 := N_{minp}$, $Q_2 := listDates = \{date_i\}, i = 1, ..., N_{dates}$, $Q_3 := listPeople = \{people_j\}, j = 1, ..., N_{people}$;
confirmation request	$Q_1 := date$; $Q_2 := listPeople = \{people_j\}, j = 1, ..., N_{people}$;
response	$Q_1 := response$;
final date	$Q_1 := date$;

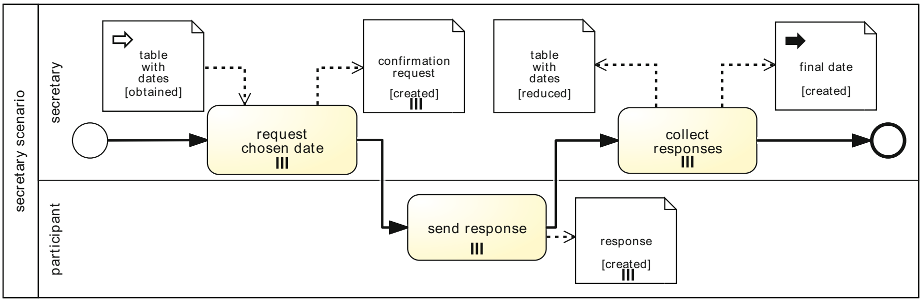


Fig. 6. Example of pattern for re-design of the use case, Secretary scenario

1. The set of alternatives is represented by the set of attributes of the input data node "table with dates": $listDates = \{date_i\}, i = 1, ..., N_{dates}$;
2. The final decision is represented by the set of attributes of the data class assigned to the output data node "final date" which is the subset of the set of attributes of the data class, assigned to the input data node $date \in \{date_i\}, i = 1, ..., N_{dates}$;
3. Decision makers are represented by initiating organizational unit "Secretary' as a resource R_1 and unit "Participant" as a resource R_2;
4. The set of attributes Q of each data class is a subset of the set of attributes of the data class assigned to the input data node: $\forall Q_c : Q_c \subseteq Q_c^i$.

As the scenario satisfies the conditions of a decision subprocess, a transformation rule to the initial process model P can be applied. All data nodes from the above mentioned scenario are replaced with one data node "Document" with different states. The transformed process model P' is presented in Figure 7. The set of attributes Q of the consolidated data class assigned to each data node in the process model P' is retrieved as a maximal subset of the sets of attributes for all data classes in the initial process model P: $Q_1 := N_{minp}$, and $Q_2 := listDates = \{date_i\}, i = 1, ..., N_{dates}$, and $Q_3 := listPeople = \{people_j\}, j = 1, ..., N_{people}$.

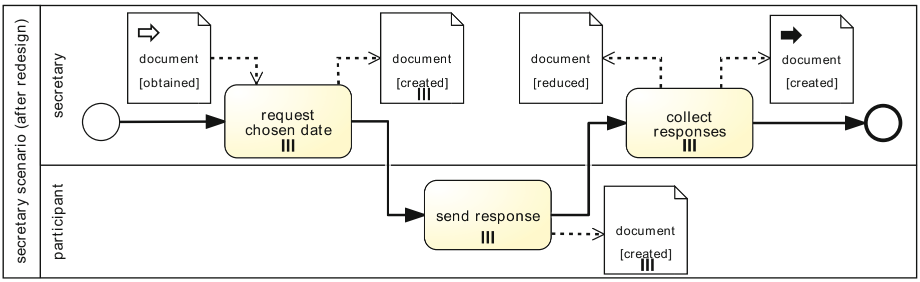


Fig. 7. Example of pattern after re-design of the use case

At the final phase of the investigation for the possibilities of business process improvement, the business analyst can decide how exactly to implement the redesigned data object “Document”. In our case the business analyst finds out that the Doodle poll can replace all the data nodes of the “Secretary” scenario.

5 Related Work

In contrast to the topic of process modeling, process redesign has not received so much attention from the scientific community [4]. A fundamental approach for business-process re-design based on best practices of successful redesign heuristics was presented in 2005 in [12]. In this paper the authors are introducing best practices, which can support the technical challenge of the business process re-design challenge in four dimensions: time, cost, quality and flexibility. This approach was applied, for example, in the healthcare domain for the reduction of throughput and service times of medical management processes, as described in [7]. As well, a number of different automation platforms supporting business process re-design were presented to the public, such as a framework based on Petri-nets [14] or, for example, software based on process mining techniques [8].

However, the above mentioned approaches are based on traditional activity flows and most of them do not consider data or business artifacts presented in the models. In our work we suggest an integrated approach which considers both activities, and the data of process models. Similar work was presented in IBM’s artifact-centric process modeling approach [6]. Also, the artifact-based approach was developed at Eindhoven University of Technology in cooperation with a Dutch consultancy company [13]. However, the above mentioned approaches provide company-specific redesign patterns. In contrast, in our work we provide a generic hybrid scheme for business process re-engineering, based on the application of techniques from decision theory.

6 Conclusion

In the paper we provided an approach for business process improvement, according to the scheme, consisting of the identification of specific patterns in process models and the redesigning of these models in order to increase its efficiency. We presented a decision subprocess, as such a redesign pattern, which incorporates the mapping of decision theory and the business process model at the modeling level. We introduced an approach for improving the internal structure of the decision subprocess by introducing and maximizing the payoff function. In future, we plan to present further redesign patterns for business process improvement.

We demonstrated the applicability of our research by improving a business process for the use case of scheduling meetings in an enterprise. In future, we plan to apply our approach to a broader class of business processes incorporating decision making. For instance, we could extend the use case used in this paper, to the integrated time management in the enterprise.

The limitation of our approach is that our scheme for business process improvement is bound to the dependencies between the data attributes of the data nodes at the modeling level. Nevertheless, in future we plan to enhance the approach with the data execution semantics.

References

1. Doodle AG, `http://www.doodle.com` (accessed: August 18, 2013)
2. Bazhenova, E.: Support of decision tasks in business process management. In: Proceedings of the 2014 Central European Workshop on Services and their Composition, ZEUS 2014 (2014) (manuscript accepted for publication)
3. Bertsekas, D.P., Tsitsiklis, J.N.: Introduction To Probability. Athena Scientific books. Athena Scientific (2002)
4. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Heidelberg (2013)
5. Ephrati, E., Zlotkin, G., Rosenschein, J.S.: Meet your destiny: A non-manipulable meeting scheduler. In: Smith, J.B., Smith, F.D., Malone, T.W. (eds.) CSCW, pp. 359–371. ACM (1994)
6. Hull, R.: Artifact-centric business process models: Brief survey of research results and challenges. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1152–1163. Springer, Heidelberg (2008)
7. Jansen-Vullers, M.H., Reijers, H.A.: Business process redesign in healthcare: Towards a structured approach. INFOR: Information Systems and Operational Research 43(4), 321–339 (2005)
8. Essam, M.M., Mansar, S.L.: Towards a software framework for automatic business process redesign. ACEEE International Journal on Communication 2(1), 6 (2011)
9. Neumann, J.V., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1944)
10. Ngamakeur, K., Yongchareon, S., Liu, C.: A framework for realizing artifact-centric business processes in service-oriented architecture. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 63–78. Springer, Heidelberg (2012)
11. Pratt, J.W., Raiffa, H., Schlaifer, R.: Introduction to statistical decision theory. MIT Press, Cambridge (1995)
12. Reijers, H.A., Liman Mansar, S.: Best practices in business process redesign: An overview and qualitative evaluation of successful redesign heuristics. Omega 33(4), 283–306 (2005)
13. Reijers, H.A.: Design and Control of Workflow Processes. LNCS, vol. 2617. Springer, New York (2003)
14. Van der Aalst, W.M.P., van Hee, K.M.: Framework for business process redesign. In: Proceedings of the 4th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE 1995), p. 36. IEEE Computer Society, Washington, DC (1995)
15. Weske, M.: Business Process Management - Concepts, Languages, Architectures, 2nd edn. Springer, Heidelberg (2012)

A Criteria Catalogue for Evaluating Business Process Pattern Approaches

Michael Becker and Stephan Klingner

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

Abstract. Process models are an important element of business process management. Modelling and management of these models can be supported by business process patterns. In recent years, various approaches for defining such patterns were introduced. The aim of this paper is to promote the precise classification of these approaches by presenting a catalogue consisting of several criteria developed by means of a systematic literature review. A first evaluation of this catalogue is conducted by classifying ten pattern approaches.

Keywords: business process patterns, process modelling, classification.

1 Introduction

Process models are of particular importance for designing, implementing, and evaluating information systems. Furthermore, they are used for multiple other purposes like supporting organisational communication, project documentation, and employee training [20]. Due to this fact, organisations already have modelled a wide variety of business processes and are continuously improving them.

Patterns have long proven to be effective concerning their ability to preserve existing knowledge, to abstract from concrete problems, and to foster communication between participitants [14]. While the usage of patterns has a long tradition in fields like software design, e.g. [19], patterns in the context of business process models (business process patterns, BPP) still constitute a rather unstructured research area. Despite several proposed approaches so far, the field still lacks a common terminology and general criteria on how to compare different pattern variants.

This work aims at increasing the understanding about BPP by presenting a catalogue of criteria for classifying different pattern approaches. In addition to this aim, the work presented here is embedded in a broader research programme concerning the configuration of complex services. Questions in this area are how to assemble a service model based on smaller BPP. Furthermore, we want to analyse how service configuration can be supported by BPP approaches. A first evaluation of service configuration approaches can be found in [7].

In this paper, we present and discuss the criteria catalogue. To evaluate the applicability of the criteria, we exemplarily analyse ten existing BPP approaches

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 257–271, 2014.

using the catalogue. For that reason, the remainder of this paper is organised as follows. In the next section, we present the theoretical background of BPP and give a brief overview of how we identified BPP approaches. The criteria used to compare BPP with each other are presented in Section 3 and applied to existing approaches in Section 4. The paper is concluded by discussing limitations and future research steps in section 5.

2 Theoretical Background

To increase the understanding about BPP, we give some additional theoretical background in this section. First, the concept of BPP is elaborated in more detail. In addition, we present our methodology for establishing the criteria catalogue and identifiying existing BPP approaches.

2.1 Business Process Patterns

According to [29], patterns are a means to establish an "abstraction from a concrete form" that occurs frequently "in specific non-arbitrary contexts". Patterns have two distinct application areas. Whereas in *forward engineering* patterns are used to create new models, during *reverse engineering* existing processes can be analysed regarding the existence of predefined patterns [18].

These two application areas coincide with different advantages from using BPP mentioned in literature. For example, BPP in forward engineering are a way to increase efficency and effectivity of process modelling by reusing existing business functions [37]. In reverse engineering, BPP can be used to identify improvement possibilities of existing processes [5] and to check the adherence to previously defined organisational or legal compliance rules [38]. On a more abstract level, it is possible to use BPP for comparing process modelling languages with each other [1].

Even though several specific approaches for specifying BPP exist, it is possible to identify various common attributes that are necessary for every pattern description [16,18]. Table 1 presents these attributes in condensed form together with a short description of each attribute.

2.2 Research Methodology

For identifying existing BPP approaches and establishing the criteria catalogue, we are currently conducting a systematic literature review based on the methodology presented in [23]. The review is structured according to the following four steps.

1. *Establish a research question*: The main goal of this paper is to establish and discuss the critera catalogue. This is supported by identifying existing approaches for specifying BPP, i.e. we deal with the question how BPP can be described. This question is embedded in a broader research programme as presented above.

Table 1. Common Attributes for describing BPP

Attribute	Description
Name, Description	General criteria for identifying a BPP. Particularly in large collection of patterns, it is necessary to provide a self-explanatory name for each pattern.
Problem	A detailed statement about the problem that is addressed by a BPP. The problem can be stated in various ways, e.g. goal-oriented by defining a desired outcome [2] or by indicating constraints a process model needs to adhere to [3].
Context	The context describes requirements that need to be satisfied for applying a given BPP. Several levels of abstraction are conceivable to define a context, ranging from a broad point of view (e.g. the structure of a company) to necessary process states.
Solution	The solution section is the core of a BPP description and defines the necessary steps to apply a pattern. Based on the formality of the pattern representation, it is possible to include graphical representations like BPMN or UML activity diagrams.
Effects	In this section, the results of applying a BPP are described. This can be achieved by a purely informal description of the context. Furthermore, it is possible to identify performance indicators that are influenced by a specific BPP [16]. Though most approaches focus on defining positive effects, it is also necessary to keep side effects in mind.

2. *Develop a search strategy for identifying relevant contributions*: We started the literature survey by reviewing publications of main conferences and journals in the BPM area, searching publication titles for *pattern*, *template*, and *Muster* (German for pattern). To extend these first results, we searched for the terms *process pattern*, *process template*, and *Prozessmuster* (German for process pattern) in the general literature databases ACM DL, IEEE XPlore, ScienceDirect, and SpringerLink. To conclude the survey, a forward-backward-reference search based on the found results is currently conducted.
3. *Establish inclusion and exclusion criteria*: We include academic and practical approaches dealing with BPP, e.g. papers presenting a pattern catalogue or general approaches on how to specify patterns. Furthermore, we consolidate contributions describing equal approaches.
4. *Analyse obtained results*: Since the focus of this paper is to present the criteria catalogue and to foster discussions about its applicability, the literature review is still in progress. We only use a small selection of identified literature for evaluating our catalogue. A rigorously and soundly evaluated criteria catalogue is an important requirement for comparing BPP approaches with each other.

The search strategy applied in step 2 is a result of the detailed classification of our review according to the taxonomy presented by [12]: The *focus* of our review is on identifying research outcomes and practical applications regarding

BPP. We conduct the review with the *goal* to integrate existing approaches by generalising and summing up central statements. In doing so, a consistent terminology can be established and used for building linguistic bridges between different BPP approaches. Furthermore, we compare existing approaches based on a given criteria catalogue. In conducting the review, we present approaches from a neutral *perspective*. In the ongoing review, we want to analyse recent literature as completely as possibly and, thus, seek an exhaustive *coverage*. However, in this work, we only present selected approaches to evaluate the criteria catalogue. Since we focus on abstract ideas of process model patterns, we *organise* the literature review conceptually. Finally, the intended *audience* of our review consists of scholars specialised in BPM.

3 Criteria Catalogue

We developed the subsequently presented criteria catalogue for comparing different BPP approaches with each other. Every criterion is either obtained from literature about classification of processes or established inductively during the literature review (depicted using the letter *i* in Tables 2, 3, and 4). To distinguish between different types of criteria, we divided the catalogue into the three classes *general* criteria, *representational* criteria, and criteria regarding the *features* of pattern approaches.

3.1 General Criteria

The criteria for a general description of BPP are presented in Table 2. Every pattern approach is classified according to a specific *type*. This criterion was established inductively during the literature review. The type is used to group approaches that are based on similar fundamental ideas and allows for an identification of the wide variety on how BPP are applied in science and practice.

- *Metamodel* The most generic approaches present BPP metamodels, i.e. they define the structure that a BPP catalogue or BPPs need to conform to [26]. These contributions are valuable, since they lay the foundation for specifying pattern catalogues. While a large collection of BPP is of great value for practice, the academic world is usually interested in justified metamodels.
- *Design Patterns* Similar to the well-known software design patterns [19], design patterns for processes are used to support modelling new processes. It is possible to use these patterns for combining predefined modelling elements at high levels of abstraction [4]. Furthermore, using design patterns may support process maintenance similar to effects found in software engineering [22].
- *Anti Pattern* Anti patterns define situations that must not or should not occur in process models. Patterns that must not occur usually violate predefined constraints that may evolve from legal or organisational requirements. Furthermore, it is possible to identify situtations that reduce the performance of a process and, thus, should be avoided. Based on the degree of

formalisation of the pattern representation, it is possible to automatically identify process parts with anti patterns. However, knowledge about anti patterns can also support creating better process models and to adhere to business process modelling guidelines [6].

- *Compliance Pattern* This type of pattern can be seen as the positive counterpart to anti patterns, since compliance patterns describe situations that process models need to adhere to. They are usually related to business rules which can, for example, be represented using the ECA paradigm (event, condition, action) [24]. Similar to anti patterns, compliance patterns might be triggered by legal or organisational requirements. Furthermore, it is possible to use compliance patterns as design patterns to foster the development of valid process models.
- *Mining Patterns* Unlike the aforementioned pattern types, mining patterns are the result of process mining activities in existing event logs. Thus, they represent situations that frequently occur in workflows. These patterns can be used to increase the understanding of a specific domain. For example, it is possible to identify co-occurring activities or order relations between activities [35]. Based on these data, tools for process modelling can be enhanced by recommendations [25]. Since mining patterns are more fundamental compared to the other pattern types, they can serve as an empirical basis for derive design patterns.

The *origin* describes the author of a pattern approach. It is possible to distinguish between patterns from research and patterns from industry. While scientific approaches are usually more complex and founded on a rigorous theoretical underpinning, approaches from practice are mostly tailored to specific challenges of companies and more lightweight. This criterion was adopted from [17].

The *scope* of a BPP determines its application area. Patterns can be tailored for a specific industry. In doing so, it is possible to compile a best practice catalogue. Contrary, there also exist pattern approaches that are not focused on one domain but provide a general method for the specification of BPP. The criterion was derived from the criterion domain used in [17]. However, the specific domains used as values are established inductively during literature review.

Access describes the availability of BPP. Organisations may have approaches to model BPP and pattern catalogues that are not publicly available due to various restrictions. In contrast, scientific approaches are often available for the public audience. Somewhere in between are BPP offered via limited access, e.g. by purchasing from third party providers. This criterion was adopted from [17].

In terms of analysing existing research approaches, the *pattern origin* is a valuable criterion. It is possible to deduce BPP by conducting case studies in different industries. In doing so, existing processes of companies are either manually or automatically analysed for the existence of patterns. A more academic approach is to review existing literature about processes and to identify commonalities. Finally, it is possible to (semi)automatically extract patterns by mining processes from event logs.

Table 2. General Criteria for Comparing Business Process Patterns

Criteria	Source	Values				
Type	i	Metamodel	Design	Anti	Compliance	Mining
Origin	[17]	Research	Industry			
Scope	i	Domain Specific	General			
Access	[17]	Closed	Limited	Open		
Pattern Origin	i	Case Study	Literature Review	Process Mining		

3.2 Representation of Business Process Patterns

The following critera address the representation of BPP and are summarised in Table 3. In general, every pattern needs to be defined in a specific *notation*. This can be done by using an existing notation, e.g. BPMN or UML. Furthermore, it is possible to extend an existing notation with necessary elements for representing BPP. On the one hand, these extensions can be facilitated by the used modelling notation. For example, UML provides capabilities to establish so-called UML profiles, an extension of the language w.r.t. the metamodel [28]. On the other hand, it is possible to extend the metamodel and to establish new notational elements. Besides using and extending existing notations, it is also possible to develop a new notation for representing BPP. This criterion was inspired by [17] where the criterion modelling language is used.

BPP can be represented using different degrees of *formalisation*. First of all, it is possible to describe BPP without any formalisation. This is often the case when patterns are described in natural language as a best practice catalogue for an organisation. Due to the lack of formality, these patterns can only be used as a starting point for modelling, since it is not possible to use them directly as modelling elements. Contrary to this, the syntax and semantics of BPP can be defined formally. Thus, the usage of patterns (formal syntax) and their meaning (formal semantics) is clearly defined. While informal description of patterns might lead to ambiguities and misunderstandings [36], formally defined patterns might be too restrictive. Since it is sometimes not necessary or not possible at reasonable expense to define formal semantics for every notational element, semiformal approaches exist. This criterion was adopted from [9, p. 59].

Similar to the formalisation degree, the *representation* of a BPP depends on the used notation. Patterns can either be represented textual or graphical. While textual representation may be based on natural language or formal logics, graphical representations use elements like rectangles and arrows to describe BPP. This criterion was adopted from [27]. Though existing research partly argues for using graphical representations to increase efficiency [39], it is susceptible to debate whether it is possible to transfer these finding to the BPP area.

To establish a catalogue of BPP, it is sometimes necessary to define *structural relations* between patterns. A rather simple approach is to indicate related patterns, e.g. patterns that solve similar problems or can be used in similar contexts. A more advanced approach for structuring a catalogue of predefined patterns is to define hierarchic restrictions between these patterns. In doing so, it is possible to describe specification and generalisation relations.

Table 3. Representational Criteria for Comparing Business Process Patterns

Criteria	Source	Values		
Notation	[17]	Existing	Extension	New
Formalisation	[9]	Formal	Semi-Formal	Informal
Representation	[27]		Graphical	Textual
Structural Relation	i		Related	Hier. Restriction
Compositional Relation	i	Sequential	Hier. Composition	Notation Dependent
Level of Abstraction	i		L-0	L-1

Besides structuring the pattern catalogue, it is possible to define *compositional relations* to specify how patterns can be combined with each other. In a solely sequential way, patterns can be used as consecutive modelling elements. Furthermore, it is possible to compose complex patterns from more simple ones, i.e. patterns are organised in a hierarchical way. If patterns are presented in an existing process modelling language, it is also possible to use the patterns in combination with other modelling elements, i.e. the relations are notation dependent. The values of this criterion were established inductively during the literature review.

The *level of abstraction* on which BPP are presented directly affects the way patterns are applied during modelling. With a L-0-representation, patterns are presented on the same level of abstraction as process modelling elements. It is necessary to note that this does not directly correspond to the usage of an existing notation. Instead, patterns might be presented language independent for being applicable in different notations. If BPP are presented in a more abstract way than processes, we call this a L-1-representation. This criterion was inspired by existing literature about metamodelling, e.g. [11]. In this sense, L-0 approaches present BPP as models and L-1 approaches are metamodels for concrete models. This criterion might be susceptible to discussion, since notations for modelling business processes have different abstraction levels of their own. However, we present this criterion as it seems important for describing a pattern approach.

3.3 Features of Business Process Patterns

The last group of criteria describes features that are supported by approaches for defining BPP; it is summarised in Table 4. Existing notations for modelling business processes allow for modelling different *views*. For a holistic representation of patterns, it is necessary to cover not only one view. We analyse pattern approaches based on the support of these views. This criterion was established inductively during research. However, it was inspired by the separation of views according to [40]. In addition to the known views control flow, data flow, and resource, we add two new views. BPP supporting the message view allow for describing the interaction between different process participants. Approaches with an abstract view are not focused on a specific view but rather provide general descriptions of BPP.

Table 4. Feature Criteria for Comparing Business Process Patterns

Criteria	Source	Values				
Views	i, [40]	Abstract	Control	Data	Resource	Message
Adaptability	i	Static	Design Choic.		Configuration Pts.	Formal
Guidelines	i		Yes		No	
Tool Support	i		Yes		No	
Predefined Patterns	i			Number		

The *adaptability* of BPP defines the degree to which the patterns can be customised for a specific use case. On the one hand, static BPP can be used to create new or evaluate existing models. However, there is no predefined way on how to adapt them for specific needs. On the other hand, there exist pattern approaches that define how patterns can be configured. This can be achieved on several ways, e.g. by giving modellers various design choices at hand, by defining fixed configuration points, or by using a formalised configuration approach. This criterion was established inductively during the literature review.

To increase usability of BPP, it is often necessary to lead modellers by giving them *guidelines* on how to use and combine patterns in different phases of the BPM life cylce, e.g. a handbook describing the application of patterns during process modelling. We analyse the pattern approach by means of existence of such guidelines. For the sake of brevity, we present this criterion based on a simple yes-no-distinction, since comparing guidelines is a separate research topic.

While a collection of patterns or a metamodel providing general pattern attributes contributes to the academic discussion about BPP, *tool support* is necessary for making pattern approaches applicable in practice. Depending on the type of the approach, a conceivable tool might be an implemented collection of reusable patterns for existing process editors. Furthermore, it is possible to develop tools for defining process models adhering to a specific metamodel. In this work, we do not detail the tool type but restrict the values to yes and no.

The last criterion we use is the amount of *predefined patterns* an approach presents. This criterion ranges from no predefined patterns to exemplary descriptions (e.g. in terms of use cases) to a given catalogue of patterns. Though the amount of existing patterns is no functional characteristic of an approach, it might indicate approaches that require additional evaluation.

4 Results

In this section, a first evaluation of the criteria catalogue described above is conducted by comparing ten BPP approaches from science with each other. It is necessary to note, that the number of BPP approaches presented does not raise the claim of a comprehensive survey. Since the focus in this stage of our research is to complete and evaluate the criteria catalogue, completeness is not required up to now. In the following, we present intial findings according to the different types general criteria, representational criteria, and feature criteria. We have selected the presented BPP approaches to point out a wide variety of different strategies.

4.1 General Criteria

Table 5 presents the evaluation of the analysed approaches regarding the general criteria of the catalogue. The most important criterion in this class is the type of a pattern. As stated above, it was developed inductively during the review process. Therefore, it was possible to classify every identified approach.

Table 5. Evaluation of general criteria

No.	Source	Type	Origin	Scope	Access	Pattern Origin
01	[16]	Metamodel	Science	Process Improvement	Open	LR
02	[18]	Metamodel	Science	General	Open	ISO standard
03	[4]	Design	Science	Configurable Processes	Open	LR
04	[8]	Design	Science	Social Processes	Open	LR, CS
05	[37]	Design	Science	General	Open	CS
06	[31]	Design	Science	General	Open	?
07	[34]	Design	Science	Change Management	Open	CS
08	[35]	Mining	Science	General	Open	PM
09	[5]	Anti	Science	General	Open	CS
10	[38]	Compliance	Science	General	Open	CS

For this paper, the review was restricted to academic contributions. Therefore, every pattern approach originates from science and is, thus, open to the public. This allows for discussing the approaches and comparing them with each other. However, the restriction to academia is a severe limitation, too. It is reasonable to assume that a multitude of pattern approaches exist in organisational practice. Particularly considering the fact that companies maintain process repositories of hundreds or even thousands of process models [15], it would be naive to assume that practice is waiting for academic pattern catalogues. However, academia can foster the pattern discussion in practice by providing new methods for identifying and describing BPP.

The interrelationship between practice and academia can be seen in the evaluation of the pattern origin criterion, too. Most of the BPP approaches presented here are based on case studies (indicated by CS in Table 5) and on literature reviews (indicated by LR). However, there is also an approach describing BPP identified via process mining (indicated by PM) and one approach that established BPP according to an existing ISO standard. By utilising case studies, process mining, and ISO standards, it is possible to develop BPP that are found in practice. Contrary, scientifically grounded patterns might be found by literature reviews.

4.2 Representation of Business Process Patterns

In Table 6, the evaluation of the representational criteria is summarised. The abbreviations in this table need to be interpreted as follows. The second column notation contains shortcuts for natural language (NL), UML Activity Diagrams

(UML AD), Event Driven Process Chains (EPC), and Semantic Business Process Modeling Language (SBPML). The values in the third column (degree of formalisation) are either formal (F) or semiformal (SF). Column 4 depicts whether a BPP approach is based on graphical (G) or textual (T) representation. The structural and compositional relations between BPP are represented as HR (hierarchic restriction), RP (related patterns), ND (notational dependent), and HC (hierarchic composition). The last column represents the level of abstraction.

Table 6. Evaluation of representational criteria

No.	Notation	Form.	Repres.	Struct. Relations	Comp. Relations	Abs. Lev.
01	NL,UML	SF	T,G	RP	ND	L-1
02	NL,UML AD	SF	T,G	none	ND	L-1
03	Abstract	SF	T,G	none	ND	L-1
04	BPMN Extension	SF	G	HR	ND	L-0
05	UML AD	SF	G	RP	ND	L-0
06	Petri Nets	F	G	HR	HC,ND	L-0
07	NL,EPC	SF	G	none	ND	L-1
08	Formal Logic	F	T	RP	n/a	L-1
09	SBPML	SF	G	none	ND	L-0
10	Abstract	SF	T	none	ND	L-1

It is noticeable that all approaches present BPP at least semiformally defined. This is due to the fact that the description of BPP is usually not restricted to natural language but rather supported by graphical representations using an existing process modelling language. This method has two benefits. First, the natural language definition allows for a detailed description of the problem and context factors addressed by a specific BPP apporach. Second, the graphical representation can be used as a starting point for using a BPP either for creating new models or for searching for patterns in existing models.

The evaluation of the criteria unveils a correlation between used notation and compositional relations. Of course, BPP of approaches based on an existing notation can be combined according to the rules of this notation, i.e. their compositional relations are notationally dependent. The same holds for BPP approaches that are not tied to a specific notation but use an abstract representation. On the one hand, this can be achieved by giving guidelines for implementation of a BPP in different languages (e.g. [4] presents implementations for Configurable EPCs [30] and for Provop [21]). On the other hand, formal logics can be used to specify restrictions processes need to adhere to [38].

An interesting result regarding the compositional relations criterion is revealed by the mining approach number 08. The criterion is not applicable for mining patterns, since it cannot be said in which form BPP are mined from existing process logs. In [35], formal logics is used to specify the mined BPP. Thus, compositional relations between these patterns are at least conceivable.

4.3 Features of Business Process Patterns

The results concerning the evaluation of feature criteria are presented in Table 7. In the second column presenting views of the BPP approaches, shortcuts for control flow (CF) and message flow (MF) are used. The criteria guidelines and tool support are presented solely based on a yes-or-no evaluation. However, a no in these columns does not automatically indicate that there is no support for these BPP approaches. In particular, for approaches based on existing process modelling languages, it is not necessary to develop distinct software tools. Instead, it is possible to reuse existing tools, possibly enhanced by pattern repositories. The same applies for guidelines that exist for process modelling languages, too.

Table 7. Evaluation of feature criteria

No.	Views	Adaptability	Guidelines	Tool Support	Predefined Patterns
01	Abstract	Static	No	No	2
02	CF	Static	Yes	No	1
03	CF	Design Choices	No	No	10
04	CF,MF	Static	Yes	No	7
05	CF,MF	Design Choices	Yes	Yes	7
06	CF	Static	Yes	No	43
07	CF	Config. Points	No	No	14
08	CF	n/a	No	No	n/a
09	CF	Static	No	No	18
10	CF	Static	Yes	No	16

The prevalence of approaches focussing on the control flow must not be considered as an indicator for evaluating a broader amount of BPP approaches. Instead of this, it is entirely based on the subjective selection of presented approaches. Particularly, the workflow patterns community has published several techniques for defining other views, too. The interested reader is referenced to the seminal works about data flow [33] and resource flow patterns [32].

Contrary to this, the prevalence of static BPP approaches can be seen as more representative. This is attributable to the used process modelling languages, since most of them do not support process configuration [30]. To overcome this shortcoming, BPP approaches present different design choices for several BPP. For example, approach 05 presents at least two variants of every BPP resulting in different UML Activity Diagrams. Approach 07 pursues another strategy. Instead of defining configurable BPP, they define BPP elements that can be combined according to predefined configuration rules.

5 Conclusion

During the evaluation of the criteria catalogue, several questions arose which should be discussed in this section. A major challenge we had to deal with, is the lack of a rigorous BPP definition resulting in discussions about what counts as a BPP and what not. Though in general it is clear what is meant by the word pattern, this might be controversial for concrete approaches.

As stated above, the definition given by [29] includes that patterns need to be an “abstraction from a concrete form”. However, this might not be applicable for BPP that are defined in an existing process modelling language and, above this, for BPP in abstraction level L-0. Since these BPP can be directly used as modelling elements, one might argue that these are not patterns but process parts. Though the decision whether an approach describes BPP does not severly influence the criteria catalogue, it needs to be considered during literature review and empirical evaluation of the catalogue.

During classification of BPP approaches according to the criteria, it was sometimes difficult to assign a type to a specific BPP approach. It has been shown that the types design, anti, compliance, and mining pattern might not be mutually exclusive. This is due to the fact that this criterion is based on the usage of a BPP. However, it is possible to use a specific BPP in more than one way, e.g. using compliance patterns as design patterns. Furthermore, transformations between anti patterns and compliance patterns are conceivable. However, we still argue for this criterion from a practical point of view, since it allows for a simple classification of BPP approaches.

In this paper, we propose an approach for establishing a unified BPP terminology and first steps for integrating existing BPP approaches. In doing so, we have identified the two criteria structural relations and compositional relations that seem of special importance for future research. It can be expected that BPP approaches allowing for the definition of relations between BPP can be combined with approaches by other authors more easily. This is due to the fact that these relations can be used to identify commonalities between different BPP.

Currently, the criteria catalogue is limited by two shortcomings that need to be overcome in future research. Though we conducted a first evaluation of the criteria, we cannot ensure consistency of the classification as of yet. Instead, we present the catalogue as a basis for discussion to increase its rigour. Using the results of the literature survey, the catalogue can be further strengthened by evaluating inter-rater reliability and, if necessary, adjust criteria.

The second shortcoming is a result of the criteria used so far. Currently, the majority of them can only be applied to classify existing BPP approaches according to several characteristics. In doing so, it is possible to identify BPP approaches that meet specific requirements. For example, a process modelling project for automated processes needs to adhere to other requirements than modelling highly collaborative human processes. While the first might lay its focus on the control flow perspective, the latter needs distinguished message flow support. Chances are that it is possible to automate this step based on a catalogue of requirements that are linked with specific BPP characteristics. However, the catalogue currently does not contain quality criteria like soundness or robustness. For example, completeness of BPP descriptions can be evaluated based on the structure presented in Table 1.

In the long run, the integration of BPP approaches should increase process modelling efficiency and effectivity by supporting modellers. Using BPP, it is possible to reduce errors that often occur during modelling [13] and to simplify business process improvement [16]. By using a unified terminology, existing tools for process modelling can be enhanced by pattern catalogues that are not limited to a single approach.

Our next research step ist to extend the criteria catalogue based on the feedback of the scientific community. The final outcome of this step should be an extensive catalogue consisting of both descriptive and discriminative criteria. The catalogue is continuously evaluated by means of the BPP approaches identified during the literature review. This should have a twofold effect: besides classification, the catalogue is further strengthened. Based on the evaluation of existing BPP approaches, our research aims at identifying use cases for applying BPP to model complex business services. Since services need to be modelled according to different views [10], it is of special importance to combine different BPP approaches with each other.

Acknowledgements. We thank the anonymous reviewers for their valuable feedback. Parts of this work were funded by grants of the German Ministry of Education and Research (BMBF) in the context of the joint research project "IPS" (01IS12013B) under the supervision of the PT-DLR.

References

1. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow Patterns. Distributed and Parallel Databases 14, 5–51 (2003)
2. Andersson, B., Bider, I., Johannesson, P., Perjons, E.: Towards a formal definition of goal-oriented business process patterns. Business Process Management Journal 11(6), 650–662 (2005)
3. Awad, A., Goré, R., Thomson, J., Weidlich, M.: An Iterative Approach for Business Process Template Synthesis from Compliance Rules. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 406–421. Springer, Heidelberg (2011)
4. Ayora, C., Torres, V., Weber, B., Reichert, M., Pelechano, V.: Enhancing Modeling and Change Support for Process Families through Change Patterns. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 246–260. Springer, Heidelberg (2013)
5. Becker, J., Bergener, P., Rckers, M., Wei, B., Winkelmann, A.: Pattern-Based Semi-Automatic Analysis of Weaknesses in Semantic Business Process Models in the Banking Sector. In: ECIS 2010 Proceedings (2010)
6. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–262. Springer, Heidelberg (2000)
7. Becker, M., Klingner, S., Sonnenberg, M., Ritter, J.: A comparative survey of service configuration approaches. In: RESER 2013: Finding Growth Through Service Activities in Barren Times (2013)

8. Brambilla, M., Fraternali, P., Vaca, C.: BPMN and Design Patterns for Engineering Social BPM Solutions. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 219–230. Springer, Heidelberg (2012)
9. Bttcher, M.: Architektur integrierter Dienstleistungssysteme. Ph.D. thesis. Universitt Leipzig (2009)
10. Bttcher, M., Fhnrich, K.P.: Service Systems Modeling. In: Alt, R., Fhnrich, K.P., Franczyk, B. (eds.) Proceedings First International Symposium on Services Science, Logos, Leipzig, Germany (March 2009)
11. Bzivin, J.: On the unification power of models. Software and Systems Modeling 4(2), 171–188 (2005)
12. Cooper, H.: Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge, Technology & Policy 1(1), 104–126 (1988)
13. Delfmann, P., Herwig, S., Lis, L., Stein, A., Tent, K., Becker, J.: Pattern Specification and Matching in Conceptual Models - A Generic Approach Based on Set Operations. Enterprise Modelling and Information Systems Architectures 5(3), 24–43 (2010)
14. Dittmann, T., Gruhn, V., Hagen, M.: Improved Support for the Description and Usage of Process Patterns. In: 1st Workshop on Process Patterns, 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications, Seattle, pp. 37–48 (2002)
15. Dumas, M., García-Bañuelos, L., Dijkman, R.: Similarity Search of Business Process Models. IEEE Data Engineering Bulletin 32(3), 23–28 (2009)
16. Falk, T., Griesberger, P., Johannsen, F., Leist, S.: Patterns For Business Process Improvement - A First Approach. In: ECIS 2013 Completed Research (2013)
17. Fettke, P., Loos, P., Zwicker, J.: Business Process Reference Models: Survey and Classification. In: Bussler, C.J., Haller, A. (eds.) BPM 2005 Workshops. LNCS, vol. 3812, pp. 469–483. Springer, Heidelberg (2006)
18. Förster, A., Engels, G.: Quality Ensuring Development of Software Processes. In: Oquendo, F. (ed.) EWSPT 2003. LNCS, vol. 2786, pp. 62–73. Springer, Heidelberg (2003)
19. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: elements of reusable object-oriented software. Pearson Education (1994)
20. Gulla, J.A., Brasethvik, T.: On the Challenges of Business Modeling in Large-Scale Reengineering Projects. In: International Conf. on Requirements Engineering (2000)
21. Hallerbach, A., Bauer, T., Reichert, M.: Capturing Variability in Business Process Models: The Provop Approach. J. Softw. Maint. Evol. 22(6,7), 519–546 (2010)
22. Jeanmart, S., Gueheneuc, Y.G., Sahraoui, H., Habra, N.: Impact of the Visitor Pattern on Program Comprehension and Maintenance. In: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement ESEM 2009, pp. 69–78. IEEE Computer Society, Washington, DC (2009)
23. Keele, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Tech. rep., EBSE Technical Report EBSE-2007-01 (2007)
24. Knolmayer, G., Endl, R., Pfahrer, M.: Modeling Processes and Workflows by Business Rules. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 16–245. Springer, Heidelberg (2000)
25. Koschmider, A., Hornung, T., Oberweis, A.: Recommendation-based editor for business process modeling. Data & Knowledge Engineering 70(6), 483–503 (2011)

26. Mu, L., Gjøsacter, T., Prinz, A., Tveit, M.S.: Specification of modelling languages in a flexible meta-model architecture. In: Proceedings of the Fourth European Conference on Software Architecture: Companion ECSA 2010, pp. 302–308. ACM, New York (2010)
27. Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C.: Making sense of business process descriptions: An experimental comparison of graphical and textual notations. Journal of Systems and Software 85(3), 596–606 (2012)
28. Pardillo, J.: A Systematic Review on the Definition of UML Profiles. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part I. LNCS, vol. 6394, pp. 407–422. Springer, Heidelberg (2010)
29. Riehle, D., Zllighoven, H.: Understanding and using patterns in software development. Theory and Practice of Object Systems 2(1), 3–13 (1996)
30. Rosemann, M., van der Aalst, W.: A configurable reference modelling language. Information Systems 32(1), 1–23 (2007)
31. Russell, N., ter Hofstede, A., van der Aalst, W., Mulyar., N.: Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMCenter.org (2006)
32. 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)
33. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns: Identification, Representation and Tool Support. In: Delcambre, L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 353–368. Springer, Heidelberg (2005)
34. Schaefer, T., Fettke, P., Loos, P.: Control Patterns - Bridging The Gap Between Is Controls And BPM. In: ECIS 2013 Completed Research (2013)
35. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action patterns in business process model repositories. Computers in Industry 63(2), 98–111 (2012)
36. Taibi, T., Ngo, D.C.L.: Formal Specification of Design Patterns - A Balanced Approach. Journal of Object Technology 2(4), 127–140 (2003)
37. Thom, L., Reichert, M., Iochpe, C.: Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence. International Journal of Business Process Integration and Management (IJBPIM) 4(2), 93–110 (2009)
38. Turetken, O., Elgammal, A., van den Heuvel, W.J., Papazoglou, M.: Enforcing Complicance on Business Processes through the use of Patterns. In: ECIS 2011 Proceedings (2011)
39. Whitley, K.: Visual Programming Languages and the Empirical Evidence For and Against. Journal of Visual Languages & Computing 8(1), 109–142 (1997)
40. Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede, A., Russell, N.: Pattern-based Analysis of BPMN An extensive evaluation of the Control-flow, the Data and the Resource Perspectives (revised version). BPM Center Report BPM-06-17, BPM Center (2006)

Understanding the Factors That Influence the Adoption of BPM in Two Brazilian Public Organizations

Carina Alves[1], George Valença[2], and André Felipe Santana[1]

[1]Centro de Informática, Universidade Federal de Pernambuco, Brazil
[2]Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Brazil
{cfa,afls2}@cin.ufpe.br, georgevalenca@deinfo.ufrpe.br

Abstract. While the increasing interest in BPM by private and public organizations confirm the relevance of process-centric philosophy, it also increases the expectations and uncertainties on how to introduce and evolve a BPM initiative. This paper investigates how BPM practices are adopted by Brazilian public organizations. We conducted case studies with two Brazilian public organizations to investigate how the interaction of barriers and facilitators influence the evolution of their BPM initiatives. A System Dynamics approach is proposed as a diagnosis tool to analyze the current performance of BPM initiatives. Systemic archetypes were created to represent specific combinations of virtuous reinforcement and balancing cycles among barriers and facilitators. We identified that support from top management and lack of team skills and competencies in BPM are key factors influencing the evolution of BPM initiatives. The implications for practice lies in the fact that systemic archetypes are generic structures repeatable in different contexts. Due to their predictable behavior, the recognition of archetypes can inspire effective action strategies to handle problematic situations that may occur in BPM initiatives facing similar situations.

Keywords: Business Process Management, Public Sector, Barriers and facilitators, System Dynamics Analysis.

1 Introduction

Business Process Management (BPM) has emerged as a holistic management approach. While the increasing interest in BPM by private and public organizations confirms the significance of process-centric approach [2], it also increases the expectations and uncertainties of how to initiate and evolve a BPM initiative. BPM is often associated with new technologies aimed at modeling and automating business processes. However, recent research suggests that the adoption of BPM philosophy involves complex cultural and organizational changes [6]. In recent years, we have observed an increasing adoption of BPM by Brazilian public sector. Two main reasons motivate public organizations to pursue a process-centric perspective. The first reason relates to the demand from citizens to increase the quality of public services. The second reason is the need to adopt digital technologies to create new service delivery channels. Brazilian public organizations face continuous pressure for

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 272–286, 2014.

accountability and transparency of their activities. Successful examples of e-government initiatives are online submission of tax returns and electronic voting. Besides serving the public interest, governmental organizations have other distinctive characteristics compared to private organizations, such as: machinery of government changes, low flexibility and innovation, stiffness of a hierarchical structure and influence of political factors. A number of studies have highlighted the growing interest of BPM by the public sector [1,2,3]. However, low attention has been paid to the evolution and overall success of BPM initiatives. Motivated by the previous scenario, this research investigates how BPM practices are adopted by Brazilian public organizations. In particular, we aim to explore the following research questions:

RQ1: What are the facilitators and barriers faced by BPM initiatives in Brazilian public organizations?
RQ2: How the interaction of facilitators and barriers influence the evolution of BPM initiatives in Brazilian public organizations?

In this paper, we report on results from two case studies conducted with Brazilian public organizations. To explore the barriers and facilitators faced by studied organizations, we designed and performed a System Dynamics Analysis approach based on the Systems Thinking discipline proposed by Senge [4]. This approach treats barriers and facilitators as factors that can interact with each other to create patterns of dysfunctional systemic behaviors, which may slowdown the success of BPM initiatives. This paper is structured as follows: Section 2 presents the research background. Section 3 describes the research method. Section 4 describes research results. Section 5 presents a discussion of findings and limitations of this study. Finally, Section 6 concludes the paper and provides directions for future research.

2 Background

2.1 Maturity of BPM Initiatives

The introduction of BPM in organizational environment aims to promote increased agility, efficiency and innovation in operation [1]. However, organizations still struggle to realize a comprehensive adoption of BPM [10]. This challenge is mainly due to the fact that BPM initiatives are affected by contextual characteristics of each organization. Therefore, the effective adoption of BPM approaches needs to be carefully instantiated to the specific needs and characteristics of each organization. Rosemann and Bruin [5] propose a comprehensive BPM maturity model. These factors were further refined by Rosemann and vom Brocke [7] to build a framework for BPM. The model describes six core factors to BPM success, which are:

- Strategic alignment – BPM initiatives must be aligned with strategic goals of the organization through a bidirectional link. Business process improvement efforts have to be defined according to strategic priorities.

- Governance – provides a reference framework to guide organizational units to ensure responsibility and accountability. BPM governance can be considered the lead of the BPM initiative.
- Methods – a set of methodologies, techniques and tools supporting the different phases of process lifecycle. BPM CBOK, Balanced Scorecard and Six Sigma are examples of such approaches.
- Information Technology – refers to hardware, software and IT solutions that support modeling, automation and improvement of business processes. Several solutions are available from workflow-based systems and process mining tools to complement BPM suites.
- People – consists on stakeholders of the organization. Education and communication of BPM principles are key strategies to disseminate a process-driven culture.
- Culture – is perceived as a key driver for the success or failure of BPM [6]. Cultural values supporting the BPM initiative include customer orientation, readiness for change, understanding of process concepts, tendency for collaboration and influential leadership.

Our research aims to investigate current strengths and weaknesses influencing BPM evolution. With this goal in mind, we adopted the former six factors to investigate how the related barriers and facilitators interact with each other and affect the performance of the BPM initiative.

2.2 System Dynamics

According to Sterman [8], System Dynamics discipline helps people to (i) learn about the structure and dynamics of the complex systems in which we are embedded, (ii) design high-leverage policies for sustained improvement, and (iii) catalyze successful implementation and change. Systems archetypes are known patterns of system behavior representing specific combination of virtuous reinforcement and balancing cycles formed by its component variables [10]. They describe or predict the behavior of a system by drawing related causal loops of variables from this scenario. There are 13 generic archetypes, according to Senge [4]. Each archetype has a script that guides the interpretation of the investigated context. The selection of an archetype depends on how the related script appropriately describes the phenomena identified. This is accomplished by recognizing variables in the context holding cause and effect relations that fit the archetype script. The use of system archetypes is a rich technique for either examining a past situation or forecasting specific scenarios by identifying potential traps and mitigating risks of occurrence. It is worth noting that the effectiveness of System Dynamics approach depends on the capacity of the actors involved to reflect on their reality. They should go beyond gathering superficial factors that translate the functioning of the studied context.

3 Research Method

This study is part of a larger research project [1, 3] that aims at (i) identifying the most relevant factors influencing the evolution of BPM and (ii) proposing strategies to increase the maturity of BPM initiatives in Brazilian public organizations. The cases were purposively selected based on expectations about their information content. Moreover, since we had access to organizations A and B, we could follow their initiatives for a prolonged time (i.e. a period of three years). In this paper, we present two case studies conducted with Brazilian public organizations. We developed a single research protocol describing data collection and analysis procedures [9]. The case studies were structured in three phases, which are following described.

Phase 1 – Semi-Structure Interviews and Focus Groups

In this phase, we conducted semi-structured interviews with two BPM leaders in each organization. The interviews consisted of two parts: (i) general questions regarding demographic and contextual aspects of the organization; and (ii) specific questions addressing goals, barriers and facilitators of their initiatives. In particular, the elicitation of barriers and facilitators was inspired by the six core elements critical to BPM success presented in Section 2.1. We conducted several in-depth interviews with two leaders from organization A and one leader from organization B during one year. All interviews were registered using a voice recorder and later transcribed to spreadsheets. We also organized two focus groups with four BPM leaders from public organizations participating in the research project, where leaders from both studied organizations have participated. The goal of these focus groups was to discuss common practices, lessons learned and challenges faced by organizations.

Phase 2 – System Dynamics Analysis

A System Dynamics Analysis was performed at this phase. During meetings with BPM leaders, we obtained an exhaustive set of barriers and facilitators. These factors were prioritized according to their impact on the initiative, and subsequently selected based on the following division: 2/3 of barriers and 1/3 of facilitators. We adopted this approach to emphasize the barriers, which are the negative aspects that must be mitigated. To avoid a complex matrix with a heavy number of crossings, we attempted not to exceed a total of 15 factors. The final set was neutralized (removal of verbs and adjectives) to derive variables and simplify the analysis of causal relations, avoiding inappropriate logical comparisons. For example, the barrier *lack of BPM roles and responsibilities* was modified to *BPM roles and responsibilities*. This set of variables was represented in lines and columns of a causal matrix. Each variable in a line was analyzed to identify its potential influence on other variables listed in the columns. Relations were determined by crossing lines with columns and received a code “d” or “i”. It indicates that the variable in the line affects the variable in the column in a directly (“d”) or inversely (“i”) proportional form. The values ‘3’ and ‘1’ were then assigned to these codes, representing standard weights related to the intensity of causal relations, where ‘3’ represents a high intensity and ‘1’ means a low intensity. Cells in the matrix with no code state that no relation was identified between two variables.

Based on the results from the interviews, we constructed causal matrixes for both organizations. Individual discussion meetings were held with BPM leaders where we explained the complete matrixes as a starting point for discussion. Then, they were asked to indicate whether the relations and weights of the factors were appropriate. After this procedure, the variables in the resultant matrix were reordered by values in the columns ‘Sum weight of causes’ and ‘Sum weight of effects’ (Figure 1). These sums inform variables’ systemic power. They are useful to identify potential leverage factors to the performance of the investigated BPM initiatives. Finally, causal relations were examined to identify systems archetypes. The archetypes represent the performance of BPM initiatives concerning barriers and facilitators. While constructing such archetypes, we included specific factors that contribute to the dynamics observed. In some cases, when interviewees did not explicitly mention the factors, we were able to infer the factors due to our familiarity with both initiatives.

#	VARIABLES	Team motivation	Support from top management	Speed of team learning	Financial resources	Compliance with the payment schedule	Vertical structure culture	Roles and responsibilities definition	Concurrence with non-BPM activities	BPM maturity	Proper operation of the BAM tool	Availability of adequate IT infrastructure	Process owner abilities	Priority to implement systems integration	BPM team turnover	Delay in implementation of modeled processes	Total of CAUSES	SUM of Weights for being CAUSE	SUM of Weights for being EFFECT
1	Team motivation		p3	p3				p1		p3			p1		i1		6	12	25
2	Support from top management	p3		p1	p3	p3	i1	p3	i3	p3	p1	p3	p3	p3		i1	13	31	8
3	Speed of team learning	p3	p1				i1	p1		p3	p3		p3		i1	i1	9	17	15
4	Financial resources	p3	p1			p3	i1	p1	p1		p1	p3			i1	i1	10	16	4
5	Compliance with the payment schedule	p1								p1					i3	i1	4	6	8
6	Vertical structure culture					i1				i3			i3				3	7	13
7	Roles and responsibilities definition	p1		p1					i1	p3			p3			i1	6	10	14
8	Concurrence with non-BPM activities	i3		i3		i1		i1		i3	i1	i1	i3	i1	p1	p1	11	19	8
9	BPM maturity	p3	p1	p3	p1		i3	p3	i1		p1	p1	p3		i1	i3	12	24	31
10	Proper operation of the BAM tool	p1	p1				i1			p1			p1				5	5	12
11	Availability of adequate IT infrastructure	p1		p3						p1	p3						4	8	8
12	Process owner abilities	p1					i3	p3		p3	p1					i1	6	12	21
13	Priority to implement systems integration	p1							i1	p1							3	3	14
14	BPM team turnover	i1		i1					p1	i3	i1		i1			p1	7	9	8
15	Delay in implementation of modeled processes	i3	i1				p3	i1		i3							5	11	11

Fig. 1. Causal Relations Matrix for Organization A

Phase 3 – Validation Meetings
A final validation meeting was undertaken with BPM leaders from both organizations to present the archetypes created. The main goals of this phase were (i) discussing our findings to identify necessary adjustments in the archetypes and obtain concluding remarks and, particularly, to carefully validate our personal inferences; and (ii) providing leaders of organizations with a rich diagnosis of BPM initiatives, highlighting its utility as a learning tool to direct suitable actions towards initiative's evolution.

4 Results

4.1 Context

Organization A is responsible for the public administration of Recife, capital of Pernambuco State. The execution of business process modeling and improvement activities started in 2006 with the conduction of several pilot projects. In 2010, a formal BPM initiative was established. Since then, they have modeled and automated major business processes. However, processes are monitored in an ad-hoc fashion. This organization has not yet established a Business Process Management Office (BPMO). Leaders recognize that this limitation may threaten the evolution of their initiative.

Organization B is responsible for auditing the accounts of Pernambuco State and its municipalities. The definition of a BPM initiative started in the beginning of 2012, although informal process modeling efforts had been previously conducted. A BPMO has been established in 2013. Table 1 gives an overview on how each organization is handling the six core factors critical to BPM maturity, as presented in Section 2.1.

4.2 Case Study A

As a result of the interviews with BPM leaders from organization A, we elaborated a list of barriers and facilitators, in which an exhaustive group of 38 variables was obtained. After few interactions, we refined the initial list into a set of 15 prioritized variables, with 5 facilitators and 10 barriers. This list is represented in the causal matrix in Figure 1. The matrix presents barriers and facilitators (in red and blue, respectively) and establishes relations among them. The BPM team was responsible to indicate the existing relations and weights among the variables.

In order to identify potential leverage factors to the performance of the BPM initiative, variables in the matrix were reordered considering the values in the columns 'Sum weight of causes' and 'Sum weight of effects'. Table 2 presents the variables reordered by their systemic power. The next step consisted of analyzing the causal relations identified and frame them in a systemic archetype. The structure selected is known as *growth and underinvestment* archetype, which is presented in Figure 2. It intends to represent situations where the performance of a system evolves during a certain time, and then it starts to halt due to a lack of investment in factors that could leverage its accomplishment.

Table 1. Contextual factors in Organizations A and B

Factor	Organization A	Organization B
Strategic Alignment	The initiative begun with the goal of monitoring KPIs. Then, it evolved to focus on the execution of process modeling and automation. The initiative successfully evolved over the years. However, it lacks an explicitly alignment with the corporate strategy.	The BPM initiative belongs to the organizational planning area, which ensures its alignment with strategic goals. According to the organization strategic planning (2012-2018), BPM initiative is formally a strategic action. The president and directors actively sponsor the initiative.
Governance	Governance is not a relevant concern. Therefore, no governance model was identified. The organization does not plan to adopt one in the short term.	Corporate governance is a main concern for the organization due to its role as public accounts auditor. The organization shall build a BPM governance model in next months.
Methods	No formal BPM methodology is adopted. However, the organization received extensive support from external consultants.	A BPM methodology is currently under construction by the internal team and external consultants.
IT	Intensive use of BPM systems, such as Bizagi and Agiles. However, the organization lacks an appropriate technical infrastructure (i.e. computers, network facilities, etc.).	Bizagi is adopted for process modeling. No BPMS is currently in use, but the organization has plans to acquire a BPM suite in the short term.
People	Stakeholders have not receive appropriate training on BPM concepts. In addition, the limited size of the BPM team restricts the evolution of the initiative.	Internal staff and external consultants conduct the BPM initiative. An intensive training program is in course to ensure that knowledge is satisfactorily transferred to the BPM team.
Culture	Strong hierarchical structure may challenge a BPM vision. BPM leaders aim to achieve individual goals without coordination with other areas, because there is a low integration among areas.	Hierarchical structure. The organization is attempting the build a project-driven culture. Corporative education and communication channels are well defined. Staff has a strong resistance to change.

Table 2. Variables in Organization A reordered by their systemic power. factors labeled with (*) should be preceded by "lack of", as reported by interviewees.

#	Variable	Facilitator or Barrier	Sum Weight of Causes	Sum Weight of Effects
1	Support from top management	F	31	8
2	BPM maturity (*)	B	24	31
3	Concurrence with non-BPM activities	B	19	8
4	Speed of team learning	F	17	15
5	Financial resources	F	16	4
6	Team motivation	F	12	25
7	Process owner abilities (*)	B	12	21
8	Delay in implementation of modeled processes	B	11	11
9	Roles and responsibilities definition (*)	B	10	14
10	BPM team turnover	B	9	8
11	Availability of adequate IT infrastructure (*)	B	8	8
12	Vertical structure culture	B	7	13
13	Compliance with the payment schedule	F	6	8
14	Proper operation of BAM tool (*)	B	5	12
15	Priority to implement systems integration (*)	B	3	4

The archetype in Figure 2 encompasses three main loops, detailed as follows.

R – This is a ***virtuous reinforcement loop*** representing a dynamic structure that led the BPM initiative to perform effectively. The central variable *Results of the BPM initiative* was inferred by us considering our deep understanding of the studied organization. It indicates the efforts carried out to promote BPM and the positive results already obtained by the initiative. This variable reinforces the initiative evolution and consequently fosters the *Support from top management*. An increased sponsorship contributes to the availability of *Financial resources*, which leads to *Compliance with the payment schedule* for external consultants. In the long term this reinforces the *Results of the BPM initiative*, which strengthens *Team motivation*. A more active BPM team promotes the *Support from top management* and equally contributes to augment the *Speed of team learning*, which in turn reinforces *Team motivation*.
B1 – This ***balancing loop*** is mainly formed by variables that were pointed out as barriers, and whose interactions tend to slow down and break the performance of the virtuous reinforcement loops in ***R***. This occurs when *Results of the BPM initiative* generates *Concurrence with non-BPM activities*. It means that people involved in the initiative started to accumulate new responsibilities besides the BPM activities. As a consequence, it is possible to remark the absence of *Roles and responsibilities definition*. The later variable decreases the *BPM maturity* and strengthens the shortage of members with *Process owner abilities*. In turn, it reinforces a *Vertical structure culture,* which is also caused by the low level of *BPM maturity*. This highlights an organizational structure with areas not properly integrated and mainly pursuing its individual goals. The complete loop results in less *Results of the BPM initiative*.

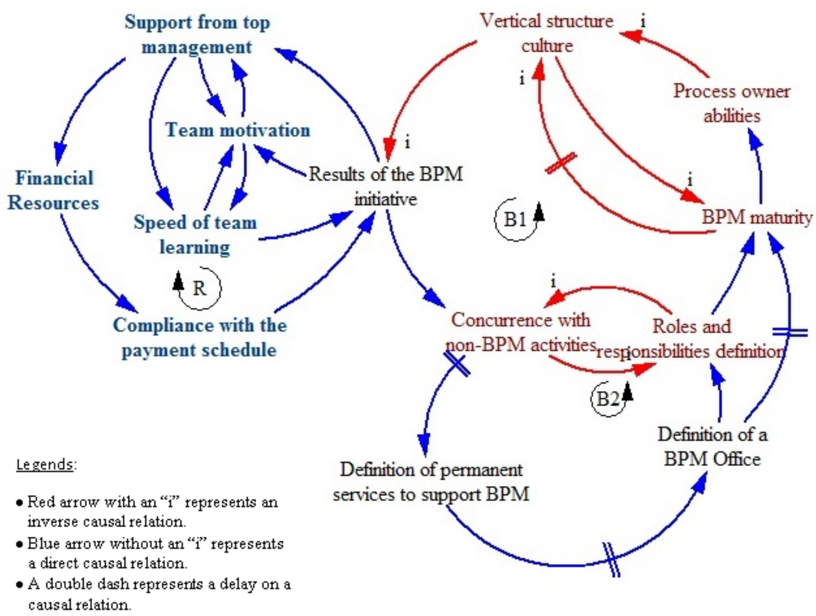


Fig. 2. Growth and underinvestment archetype for the BPM initiative in Organization A

B2 – This represents a ***corrective balancing loop*** to describe the underinvestment structure present in the BPM initiative. We can perceive that the *Concurrence with non-BPM activities* leads to the need of *Definition of permanent services to support BPM*. This promotes the relevance of the variable *Definition of a BPM Office*, considering that a specific organizational unit ideally should provide BPM services. The establishment of a BPM Office fosters the *Roles and responsibilities definition*. As a consequence, the organization achieves greater *BPM maturity*. Loop ***B2*** shall invert the slowdown effect of balancing loop in ***B1*** and consequently contributes to the sustenance of the initial performance growth.

In the second round of interviews conducted to follow the initiative evolution, we observed that Organization A had profound changes in the majority of managerial positions due to new elections. However, the BPM initiative did not have significant evolution. The BPM Office was still not fully established and the supporting infrastructure to process automation continued deficient. Hence, there was not a proper infrastructure support for new BPM projects. We conclude that the typical effect of growth and underinvestment archetype is strongly characterized in Organization A.

4.3 Case Study B

In Organization B, our initial analysis of barriers and facilitators generated a list of 21 variables. After prioritizing this list, 15 variables were obtained, from which 5 were facilitators and 10 were barriers. The causal matrix is presented in Figure 3.

#	VARIABLES	Support from top management	BPM pilot projects	Availability of an area responsible for disseminating BPM culture	Clarity of the BPM initiative objectives	Prerogative of organizational development by the unity responsible for the BPM initiative	Concurrence with non-BPM activities	Discontinuity of BPM initiatives	Resistance to change	Integration of organizational areas	Managers fearing power loss	Focus on strategic goals	Availability of resources	Internal and external communication strategies of the BPM initiative	Team skills and competencies in BPM	Availability of a BPMS	Total of CAUSES	SUM of Weights for being CAUSE	SUM of Weights for being EFFECT
1	Support from top management		d3	d3	d1	d1	i1	i3	i3	d1			d3			d1	10	20	20
2	BPM pilot projects	d3					d3	i1	i3	d3	d1		i3		d1		8	18	32
3	Availability of an area responsible for disseminating BPM culture		d3		d3			i3	i3	d3		d3	d1	d1	d3		9	23	25
4	Clarity of the BPM initiative objectives	d3	d3	d1				i3	i1					d3	d1		7	15	7
5	Prerogative of organizational development by the unity responsible for the BPM initiative			d3									i1	d1			3	5	12
6	Concurrence with non-BPM activities		i3	i3				d1	d3	d1			i1				6	12	13
7	Discontinuity of BPM initiatives	i3	i3	i3					d1								4	10	31
8	Resistance to change	i3	i3	i1		i3		d3		i3			i1		i3		8	20	22
9	Integration of organizational areas	d1	d3	d3		d1	i3	i3				d3				d1	8	18	23
10	Managers fearing power loss		i1	i1		i3	d3	d3	d3	i3							7	17	2
11	Focus on strategic goals	d1					i1	i1	i1	d3							5	7	11
12	Availability of resources		d3	d1		d1		i1						d1	d3	d3	7	13	11
13	Internal and external communication strategies of the BPM initiative	d3	d3	d3			i1	i3	i3	d3	i1						8	20	31
14	Team skills and competencies in BPM		d3	d3	d3	d3	i1	i3	i1			d1	d1	d3		d3	11	25	11
15	Availability of a BPMS	d3	d1					i3		d3		d1					5	11	8

Fig. 3. Causal Relations Matrix for Organization B

It was possible to classify the variables according to their systemic power by examining the columns 'Sum weight of causes' and 'Sum weight of effects' in the resulting causal matrix. Table 3 displays the reordered matrix, classifying the 15 analyzed factors. By analyzing relations in the causal matrix and concerns manifested by the BPM leaders, we identified the archetype structure *Growth and Underinvestment*, similarly to the case of Organization A. We discuss the causal relations and loops that compose the archetype shown in Figure 4.

R1 and **R2** – These two cycles compose a ***virtuous reinforcement loop*** where *Availability of an area responsible for disseminating BPM culture* enables *Clarity of the BPM initiative objectives*. Therefore, it is possible to establish *BPM pilot projects* as a central goal that triggers business process improvement efforts along the organization. A clear view of the initiative objectives also fosters the *Support from top management*. As a whole, these two later factors reinforce the *Availability of an area responsible for disseminating BPM culture*. These loops form a dynamic structure that initially leverages the performance of the BPM initiative in Organization B.

B1 – This represents a ***balancing loop*** that in the long term shall inhibit the positive influence of the virtuous cycles ***R1*** and ***R2***. In this loop, by increasing the number of *BPM pilot projects* the organization reduces the *Availability of resources*, since the

Table 3. Variables in Organization B reordered by their systemic power. Factors labeled with (*) should be preceded by "lack of", as reported by interviewees.

#	Variable	Facilitator or Barrier	Sum Weight of Causes	Sum Weight of Effects
1	Team skills and competencies in BPM (*)	B	25	11
2	Availability of an area responsible for disseminating BPM culture	F	23	25
3	Support from top management	F	20	20
4	Internal and external communication strategies of the BPM initiative (*)	B	20	31
5	Resistance to change	B	20	22
6	BPM pilot projects	F	18	32
7	Integration of organizational areas (*)	B	18	23
8	Managers fearing power loss	B	17	2
9	Clarity of the BPM initiative objectives	F	15	7
10	Availability of resources (*)	B	13	11
11	Concurrence with non-BPM activities	F	12	13
12	Availability of a BPMS (*)	B	11	8
13	Discontinuity of BPM initiatives	B	10	31
14	Focus on strategic goals (*)	B	7	11
15	Prerogative of organizational development by the unity responsible for the BPM initiative	F	5	12

BPM team shall be allocated in several parallel projects. In the long term it contributes to the *Discontinuity of BPM initiatives* and hampers the conduction of *BPM pilot projects*, which is a central variable of the virtuous loops ***R1*** and ***R2***. This is a paradoxical effect in the dynamics of this case: by increasing the number of *BPM pilot projects* in the short term, the initiative shall block these projects in long term, due to the low *Availability of resources*.

B2 – Similarly to ***B1***, this represents a ***balancing loop*** that in the long term tends to inhibit the virtuous cycles ***R1*** and ***R2***. Within this cycle, *BPM pilot projects* shall increase *Managers fearing power loss*. It means that managers may understand that these projects are affecting their own areas and threatening their control. This intensifies *Resistance to change*, contributing to *Discontinuity of BPM initiatives* and reducing the number of *BPM pilot projects* in the long term.

B3 – This is another ***balancing loop*** that tends to inhibit the virtuous cycles ***R1*** and ***R2*** in the long term. In this cycle, *BPM pilot projects* increase the *Concurrence with non-BPM activities*. It means that BPM activities initially do not have a high priority, as they will be undertaken in parallel with daily activities by technical and management staff. This intensifies *Resistance to change*, considering that participants of the pilot project may perceive process improvement efforts as fruitless activities or as

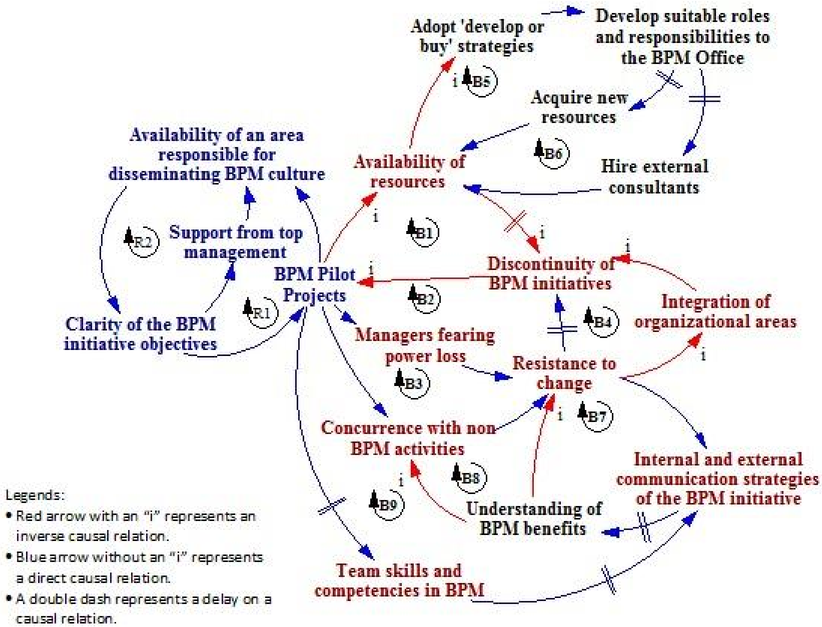


Fig. 4. Archetype **Growth and Underinvestment** for BPM initiative in Organization B

something that will bring extra work. As a consequence, the later variable contributes to the *Discontinuity of BPM initiatives*, which shall hamper the execution of new *BPM pilot projects* in the long term.

B4 – This ***balancing loop*** tends to negatively influence the virtuous cycles ***R1*** and ***R2*** in the long term. *BPM pilot projects* shall generate *Concurrence with non-BPM activities*, which will increase *Resistance to change*. As a result, there will be less *Integration of organizational areas*, promoting the *Discontinuity of BPM initiatives*. This cycle in the long term may reduce the number of *BPM pilot projects*.

B5 and **B6** – These two cycles act as balancing loops, but differently from ***B1***-***B4***, they compose a wide ***corrective balancing loop***. The low *Availability of resources* causes the organization to *Adopt 'develop or buy' strategies*. This may lead the organization to *Develop suitable roles and responsibilities* to the BPM Office to ensure a clear view of current needs for BPM. Therefore, the organization can either *Acquire new human resources* or *Hire external consultants*. These variables increase the *Availability of resources*. We must highlight that part of the corrective actions in the archetype are already in course in Organization B. They have recently hired external consultants to support the establishment of a BPM Office.

B7 and **B8** – These cycles are similar to ***B5*** and ***B6*** and compose a ***corrective balancing loop***. To reduce the *Resistance to change* it is necessary to develop *Internal and external communication strategies* of the BPM initiative. These strategies shall disseminate information about the BPM key concepts, while communicating the results of the initiative. As a consequence, there should be an increased *Understanding of*

BPM benefits. This tends to reduce *Resistance to change* and *Concurrence with non-BPM activities*.
B9 – in the long term *BPM pilot projects* may help to develop *Team skills and competencies in BPM*, which also in the long term would support the development of *Internal and external communication strategies* of the BPM initiative. After a certain period, these strategies shall increase the *Understanding of BPM benefits* and consequently lead to less *Resistance to change*. Finally, this should reduce risk of *Discontinuity of BPM initiatives* and increase the number of *BPM pilot projects*. Hence, **B9** is a loop with long term effects.

The delay in perceiving the need to take corrective actions is a common trap in *growth and underinvestment* situations. Additionally, a contradictory characteristic of these contexts involves actions that in the short term serve to leverage a desired performance. These actions also tend to block this scenario if nothing is done to establish an appropriate infrastructure in the long term.

5 Discussion

System dynamics analysis enabled us to identify factors affecting the current situation of the studied organizations. We observed that in Organization A *Support from top management* was the main cause of several patterns detected in the initiative evolution. It means that the initiative is based on a robust support from the executive management, which empowers the BPM team to take strategic decisions and engage the whole organization to effectively adopt BPM. On the other hand, increasing *BPM maturity* was perceived as the direct result of the interaction among variables. In Organization B, not having a skilled BPM team was evidenced as the central cause of interaction among variables. The lack of *Team skills and competencies in BPM* in the long term may threaten the initiative evolution and even affect its discontinuity. On the other hand, *BPM pilot projects* appeared as the main consequence of the dynamics among several variables. Pilot projects are facilitated by an active sponsorship and existence of an area responsible for the initiative. Concerning the resultant archetypes, the following similarities were observed through the comparative analysis of the two organizations dynamics:

- The ***growth and underinvestment*** archetype is a valid systemic pattern for both initiatives. This occurs since both organizations did not properly invest in a supportive infrastructure for the initiatives, which may hamper BPM evolution. Establishing major BPM roles and responsibilities, and formalizing a BPMO are strategies to overcome this infrastructure deficiency and enable the initiative to thrive.
- Our analysis identified common variables for both cases. They play a similar role in the initiatives: *Support from top management* integrated the virtuous reinforcement cycle and enabled the growth of the initiatives. While *Concurrence with non-BPM activities* was part of the cycle that inhibits the success of the BPM initiative.

We also identified key differences between the dynamics observed in the initiatives:

- In Organization A the ***growth and underinvestment*** archetype represents the current reality of the BPM initiative. The archetype was considered as representative for the studied situation, since its typical effect of having the evolution threatened by a lack of investment in infrastructure is strongly characterized in this organization. The initiative experienced a growing period, but it currently presents signs of stagnation due to its deficient infrastructure. This became a serious limitation for the automation of new business processes, for instance.
- Organization B is slightly different because its initiative has recently started. Thereby, the archetype represents a forecasted scenario, and the organization increases its chances to act preventively against undesired predictions. It is important to mention that we obtained richer data from this organization, which enabled us to develop a more detailed archetype.

The interpretation of the archetypes suggests that organizations may perceive a tradeoff between expanding the BPM initiative due to its initial success and investing sufficient resources to guarantee its sustainable growth over time. In Organization A, it was evidenced that the poor infrastructure is mainly related to the absence of a formal BPMO. This situation may change if an office is established so that staff turnover is reduced. One appropriate decision here would be to stop the automation of new processes until this infrastructure is at least satisfactory. The analysis of BPM initiative dynamics in Organization B revealed that *Team skills and competencies in BPM* and *Internal and external communication strategy for BPM* are relevant leverage points. These are actual barriers with a heavy systemic impact, but they do not demand a challenging action. They represent factors that the organization should carefully treat to promote a corrective balancing cycle and foster the initiative evolution.

Comparing our results with the factors proposed Rosemann and vom Brocke [7], we observed that the common variable *Support from top management* is associated with the ***strategic alignment*** factor. The variable *Concurrence with non-BPM activities* represents an initial resistance to adopt BPM practices. This is related to the factors ***people*** and ***culture***, reinforcing the relevance of BPM education and communication to establish a process-oriented culture. The intention of both initiatives to establish major BPM roles and responsibilities, together with a formal BPMO indicates their concern with ***governance*** and ***methods*** factors. In both organizations, we did not observe an explicit relevance of ***IT*** factor.

6 Conclusion

This paper presents the use of systemic archetypes to explore the cause-effect interaction of barriers and facilitators in BPM initiatives of two Brazilian public organizations. We investigated patterns in the relations between barriers and facilitators to recognize archetypes representing systemic behaviors in the studied initiatives. Due to their predictable behavior, the recognition of archetypes can inspire effective action strategies to handle problematic situations that may occur in BPM initiatives facing similar situations. It is worth mentioning that the interpretation of a particular reality in terms of archetypes depends upon the following conditions: (i) researcher experience in the general structure of known archetypes; (ii) identification in the studied reality of features and variables that fit a particular archetype structure; (iii) validation of the created archetypes with participants of the studied reality. We plan to perform new case studies to increase the understanding on how systemic archetypes can help BPM teams to reflect upon their own actions and conduct informed decisions during the evolution of BPM initiatives.

References

1. Valença, G., Alves, C., Santana, A.F., Oliveira, J., Monteiro, H.: Understanding the Adoption of BPM Governance in Brazilian Public Sector. In: Proceedings of 21st European Conference on Information Systems, Utrecht, Netherlands (2013)
2. Tregear, R., Jenkins, T.: Government Process Management: A review of key differences between the public and private sectors and their influence on the achievement of public sector process management. In: Proceedings of 1st Int'l Workshop on Management of Business Processes in Government, Brisbane, Australia (2007)
3. Santana, A.F.L., Alves, C.F., Santos, H.M., Felix, A.L.C.: BPM Governance: An Exploratory Study in Public Organizations. In: Proceedings of the 12th Int'l Conf.on Business Process Modeling, Development and Support, London, pp. 46–60 (2011)
4. Senge, P.M.: The Fifth Discipline: The Art & Practice of the Learning Organization. Crown Publishing Group, New York (2006)
5. Rosemann, M., Bruin, T.: Towards a Business Process Management Maturity Model. In: Proceedings of 13th European Conference on Information Systems, Regensburg, Germany, May 26-28 (2005)
6. Tumbas, S., Schmiedel, T., Bringmann, M.: vom Brocke, J.: Developing a BPM-Supportive Organizational Culture: on the Importance of Contextual Factors. In: Proceedings of 21st European Conference on Information Systems, Utrecht, June 6-8 (2013)
7. Rosemann, M., vom Brocke, J.: The Six Core Elements of Business Process Management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, vol. 1, pp. 107–122. Springer, Heildelberg (2010)
8. Sterman, J.D.: Business Dynamics: Systems thinking and modeling for a complex world. McGraw-Hill (2000)
9. Yin, R.: Case Study Research: Design and Methods, 5th edn., vol. 4. Sage Publications (2009)
10. Wolstenholme, E.: Using generic system archetypes to support thinking and modelling. Syst. Dyn. Rev. 20, 341–356 (2004)

Using Fractal Process-Asset Model to Design the Process Architecture of an Enterprise: Experience Report

Mturi Elias, Ilia Bider, and Paul Johannesson

Department of Computer and Systems Science (DSV),
Stockholm University (SU), Stockholm, Sweden
{mturi,ilia,pajo}@dsv.su.se

Abstract. The Fractal Process-Asset (FPA) model has been proposed as an approach for identifying business processes and defining relationships between them in an enterprise. This paper reports on a project of applying the model in a Higher Education Institution enterprise. The goal of this project is twofold. One is to design a process architecture that provides a holistic view on the major business processes and their interconnections in the department to be used for business planning and development. Second is to test whether the FPA model is suitable for creating a holistic view on the major business processes and their interconnections in an enterprise. The FPA model has been applied and evaluated by business domain experts in a frame of a real organization—department of Computer and System Sciences, Stockholm University. The results show that the FPA model is understandable and suitable for creating a holistic view on the major business processes in an enterprise and their interconnections. The educational processes architecture produced is understandable and can be used for business planning and development. Though the study has been conducted only in one organization, there is a likelihood that the results achieved are of general nature.

1 Introduction

A business process is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers [1]. The processes interact with each other throughout an enterprise, for example, via outputs from one process forming the inputs for another process. Each process is, therefore, part of a larger whole, and the enterprise can be seen as complex networks of interconnected processes. Nevertheless, in many organizations, business processes are still considered and designed in isolation, which results in creating potential gaps, inefficiencies and hindrances to enterprise performance. To address these issues, the enterprise needs to get a holistic view on the enterprise processes that shows how they are interconnected [2]. A process architecture is a schematic that shows the ways in which the business processes of an enterprise are grouped and interlinked [3, 4].

This paper reports on the project of building educational process architecture that provides a holistic view on the major business processes related to education in the

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 287–301, 2014.

department of Computer and System Sciences, Stockholm University. In building this view, the Fractal Process-Asset (FPA) model from [5] has been used to depict relationships between different processes that are not directly connected between each other through input-output relationships. The FPA model has been developed for “unwinding” all processes in an organization starting from the main one. The model consists of two types of archetypes (patterns): the process-assets archetype and the asset-processes archetype. Unwinding is done by recursively applying the two archetypes. The process-assets archetype is used to find all assets that the organization need to repeatedly run a given process, for example employees, or appropriate infrastructure. The asset-processes archetype is used to find all processes aimed at maintaining a given assets, for example, hiring and training people. By applying these archetypes, one can design the process architecture of an enterprise starting from the main process and going downwards via repeating pattern "a main process->its assets->processes for each assets->assets for each process->...". An asset is anything tangible or intangible such as materials, staff, or other resources that are needed for successful running process instances of a certain type [5, 6].

The Fractal Process-Asset (FPA) model from [5] is relatively new. Though it received positive responses from the experts in management consulting, it was not sufficiently tested in practice so far. Testing it was considered as a scientific/research goal for the current project.

Summarizing the above the goal of the project reported in this paper is twofold:

1. To create a holistic view on the major business processes in the department related to teaching and learning and their interconnections to be used for business planning and development.
2. To test the appropriateness of the FPA model for this task, more specifically to investigate:
    a. Whether a FPA model could be applied to build a process architecture in practice in resource efficient way
    b. Whether it can reveal or explicate important facts about the business
    c. Whether it could be understood and appreciated by domain specialists.

To reach the goals above, a number of interviews were conducted with the members of staffs of the department responsible for different parts of the business, including unit directors, head of academic units, teachers, and administrative staff. Based on these interviews, the educational process architecture has been built and presented to the domain specialists for validation. After demonstration, the domain specialists answered a number of questions that allowed us to validate the holistic view on processes in the department, and appropriateness of the fractal approach from [5] for this end. The results show that the Fractal Process-Asset (FPA) model is suitable for creating a holistic view on the major business processes in an enterprise and their interconnections. The educational process architecture produced is understandable and can be used for business planning and development. Though the study has been conducted only in one organization, there is a likelihood that the results achieved are of general nature.

The sections below present the details of the project, including lessons learned. In Section 2, we describe the context of the project by presenting a short description of the organization, business process under investigation (teaching and learning process), and project team. Section 3, provides the overview of the fractal process-asset (FPA) model we have used in the project. Section 4, describes the planning and execution of the project. Section 5 presents and discusses the educational process architecture built during the project. Section 6 discusses the results achieved and lessons learned. Section 7 contains concluding remarks and plans for the future.

2 Context of the Project

In this section, we present the specific context in which the project has been completed.

2.1 The Organization. The project has been completed in the department of Computer and System Sciences (DSV). DSV belongs to the Faculty of Social Sciences at Stockholm University (SU) and carries all types of academic activities: undergraduate, postgraduate and research with about 5700 students. It runs bachelor, master, and doctoral programs in the fields of Computer Science and Information Systems. It has about 280 staff members including teachers and administrative personnel.

2.2 The Process View. The primary business of DSV is teaching and learning, research and consultancy. The focus of this study is on the teaching and learning as core business process performed in the department. Teaching and learning, as a main business process, involve all processes linked to delivering knowledge to students. They include teaching, examining and graduation. We intended to investigate and map all business processes, which are vital to the successful execution of the teaching and learning for knowledge delivery as the main business process.

2.3 The Team. The project involved senior staff from both teaching and administrative units; this group will be referred to as the *business domain expert*. The business domain experts included director of studies, director of finance and administration, head of academic units, IT director and coordinators of some specific academic programs. The major team consisted of two teachers and one PhD student (all authors of this paper); this group will be referred to as the *enterprise modeling experts*. The two teachers had long experience of teaching and research in the field of enterprise modeling. The student represented the "learning" stakeholders.

3 The Fractal Process-Asset (FPA) Model

In this section, we describe the Fractal Process-Asset (FPA) model we have applied in the project to design the educational process architecture of DSV. The model was developed, in our previous research [5], to help organizations to easily identify and map their business processes and the interconnection between them. The model is based on the process-assets and asset-processes archetypes [5]. *Process-assets archetypes* help to find out what assets are needed for a particular process, especially for a main process from which we start unwinding. *Assets-processes archetypes* help to find out supporting processes that are needed to have each type of assets ready available for deployment.

In the following sub-sections, we describe the two concepts and we show how they can be applied to design the process architecture of an enterprise.

3.1 The Process-Assets Archetype for Main Processes

We consider an enterprise to be any organization whose operational activities are financed by external stakeholders. For example, an enterprise could be a private company that gets money for its operational activities from the customers, or a public institution that gets money from the taxpayers. We also consider a main process to be any process that creates value to the enterprise's external stakeholders, which they are willing to pay for. Our definition of the term main process may not be the same as those of others [7, 8]. For example, we consider as main processes neither sales and marketing processes, nor product development processes in a product manufacturing company. However, our definition of the main process does cover processes of producing and delivering products and services for external stakeholders, which is in line with other definitions of main processes [7, 8].

Main processes are the vehicles of generating money for operational activities. To get a constant cash flow, an enterprise must ensure that new business process instances (BPIs) of main processes are started with some frequency. To ensure that each started BPI can be successfully finished, the enterprise needs to have assets ready to be employed so that the new BPI gets enough of them when started. We consider that any main process requires the following six types of assets [5] (see also Figure 1 and Figure **2**):

1. *Paying stakeholders*. Examples: customers of a private enterprise, members of an interest organization, local or central government paying for services provided for the public.
2. *Business Process Templates* (BPTs). Examples are as follows. For a production process in a manufacturing company, BPT includes product design and design of a technological line to produce the product. For a software development company that provides customer-built software, BPT includes a software methodology (project template) according to which their systems development is conducted. For a service provider, BPT is a template for service delivery.
3. *Workforce* – people trained and qualified for employment in the main process. Examples: workers at the conveyor belt, physicians, researchers.
4. *Partners*. Examples: suppliers of parts in a manufacturing process, a lab that complete medical tests on behalf of a hospital. Partners can be other enterprises or individuals, e.g., retired workers that can be hired in case there is temporal lack of skilled workforce to be engaged in a particular process instance.
5. *Technical and Informational Infrastructure* – equipment required for running the main process. Examples: production lines, computers, communication lines, buildings, software systems etc.
6. *Organizational Infrastructure*. Examples: management, departments, teams, policies regulating areas of responsibilities and behavior.

More clarifications about the above assets can be found here [5]. The type of processes (main) together with types of assets required for running it constitute a process-assets archetype for main processes. An arrow from the asset to the process shows the needs to have these types of asset in order to successful run process instances of the given type. A label on an arrow shows the type of assets. **Figure 2** is an example of such instantiation for teaching and learning process.

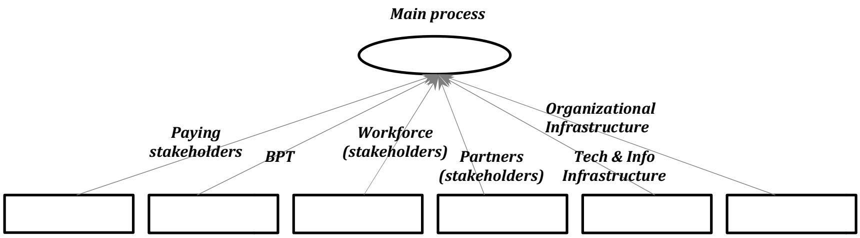


Fig. 1. The process-assets archetype for main process

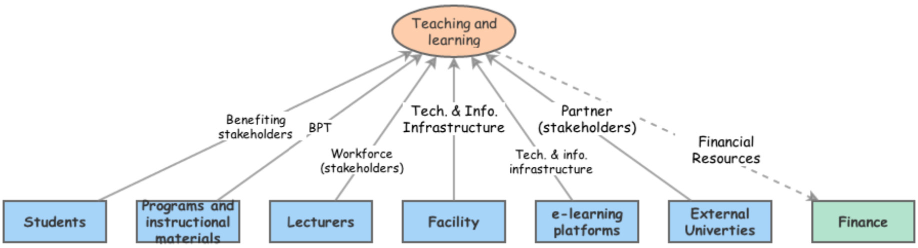


Fig. 2. An example of instantiation of the process-assets archetype for teaching and learning as the main process

3.2 The Asset-Processes Archetype

In the previous section, we have introduced six types of assets that are needed to ensure that BPIs of a main process run smoothly and with required frequency. Each asset type requires a package of supporting processes to ensure that it is in the condition ready to be employed in BPIs of the main process. We present this package as consisting of three types of processes connected to the life-cycle of each individual asset (see also an example in Figure 3):

1. *Acquire* – processes that result in the enterprise acquiring a new asset of a given type. The essence of this process depends on the type of asset, the type of the main process and the type of the enterprise. For a product-oriented enterprise, *acquiring* new customers (paying stakeholders) is done through marketing and sales processes. *Acquiring* skilled work force is a task completed inside a recruiting process. *Acquiring* a new BPT for a product-oriented enterprise is a task of new product and new technological process development. Creating a new BPT also results in introducing a new process in the enterprise.

2. *Maintain* – processes that help to keep existing assets in the right shape to be employable in the BPIs of a given type. For customers, it could be Customer Relationship Management (CRM) processes. For workforce, it could be training. For BPT, it could be product and process improvement. For technical infrastructure, it could be service.
3. *Retire* – processes that phase out assets that no longer can be used in the main process. For customers, it could be discontinuing serving a customer that is no longer profitable. For BPTs, it could be phasing out a product that no longer satisfies the customer needs. For workforce, it could be actual retirement.

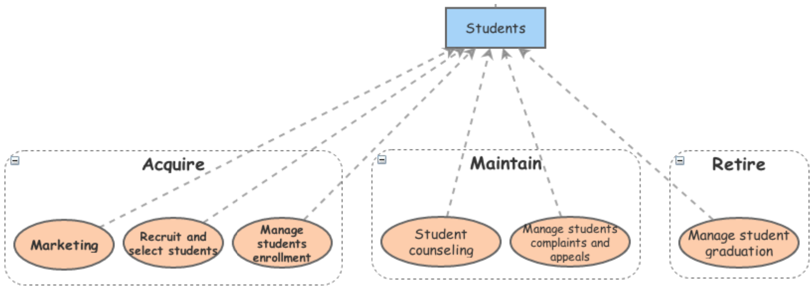


Fig. 3. An example of instantiation of the asset-processes archetypes for "student" asset

The asset-processes archetype can be graphically presented in the form of Figure 3. In it, the asset type is represented by a rectangle, and a process type - by an oval. An arrow from the process to an asset shows that this process is aimed at managing assets of the given type. The label on the arrow shows the type of the process – *acquire*, *maintain*, or *retire*. Instantiation of the archetype is done by inserting labels inside the rectangle and ovals. Actually, Figure 3 is an example of such instantiation for the *student* asset in a higher education institution.

3.3 Archetypes for Supporting Processes

Types of assets that are needed for a supporting process can be divided into two categories: general and specific asset types. General types are the same as for the main process, except that a supporting process does not need paying stakeholders. The other five types of assets needed for a main process: BPT, workforce, partners, technical and informational infrastructure, organizational infrastructure, might be needed for a supporting process as well.

Note also that some supporting processes, e.g., servicing an infrastructure, can be totally outsourced to a partner. In this case, only the partner's rectangle will be filled when instantiating the archetype for such a process.

Additionally to the five types of assets listed above, other types of assets can be added to a specific category of supporting processes. We have identifying two additional assets for supporting processes of acquiring an asset that belongs to the category of stakeholders, e.g., paying stakeholders, workforce, and partners:

- *Value proposition*, for example, description of products and/or services delivered to the customer, or salary and other benefits that an employee gets.
- *Reputation*, for example, of being reliable vendor, or being a great place of work.

Adding the above two asset types to the five already discussed, gives us a new process-assets archetype, i.e., the archetype for the acquiring stakeholders. An example of instantiation of such an archetype is presented in Figure 4.

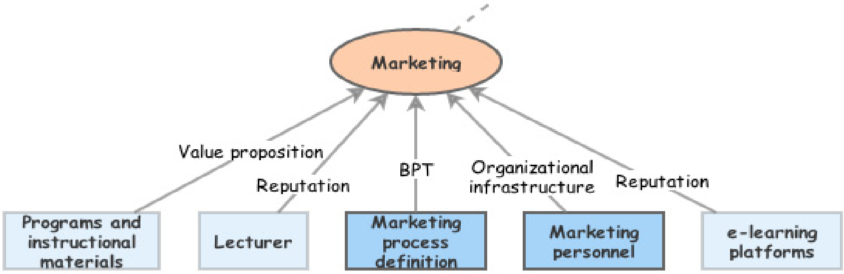


Fig. 4. An example of instantiation of the process-assets archetype for acquiring students

4 Project Execution Plan

Activities of the project presented in this paper consisted of three phases as shown in Figure 5. (a) Investigation of business processes, (b) modeling of the educational processes architecture of the department, (c) evaluation of the results.

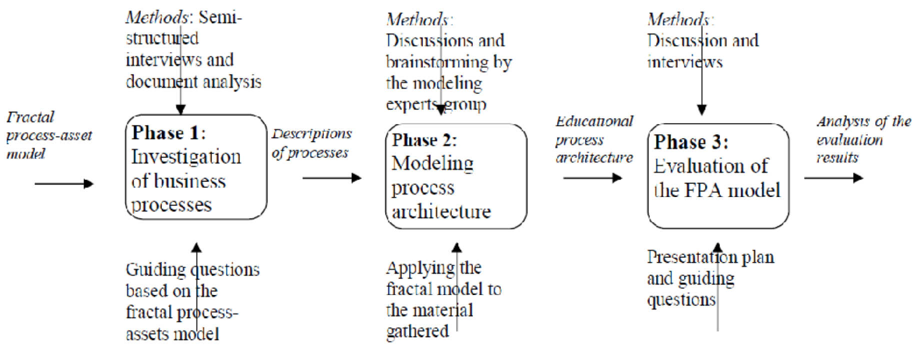


Fig. 5. The Project Execution Plan

4.1 Phase 1. Investigation of Business Processes

In this phase, *enterprise modeling experts* investigated business processes performed in the department to enable the modeling of the educational process architecture. Our aim was to find out what business processes are involved, and what is needed for running the teaching and learning process at the department. Two methods were employed to investigate the business processes.

The first method was through semi-structured interviews [9]. They enabled us to get an in-depth understanding of business processes involved based on the issues brought up during the interview with the *business domain experts* (interviewees). We interviewed nine *business domain experts*, starting with the director of studies who is the main senior personnel responsible of ensuring the successful execution of the teaching and learning process. The list of interviewees was later extended to the operational staff. Getting the input from the operational staff who performs the actual activities in different business processes increased our understanding of the details of various processes related to teaching and learning.

The second method used to investigate the business processes was the document analysis [9]. Document analysis focuses on information from formal documents or records. In this project document analysis was useful to complement the information obtained from the interviews. Therefore during the interviews we asked and received supporting documents, from which some of the business processes were identified.

The interviews protocols and documents were then analyzed and used to build the educational process architecture described in the next section.

4.2 Phase 2. Modeling of the Educational Process Architecture

In this phase, the educational process architecture was designed by applying the Fractal Process-Asset (FPA) model and the data collected in phase 1. It is during this phase that the first goal of the project was achieved. The design was done by *modeling experts* using Insightmaker [10] as a modeling tool. The choice was based on our knowledge of using the tool. The design process consisted of a number of iterations that in a simplified form can be presented as a sequence of the following steps:

- **Step 1**: Identification of assets that are utilized by the department to ensure successful execution of the teaching and learning process. This was achieved by applying the process-assets archetype. The resulting model is shown in Figure 2 and a detailed description of the model is provided in the results section 5.
- **Step 2**: Identification of processes involved for acquiring, maintaining, and retiring each asset identified in step 1. This was achieved by applying the asset-processes archetype. Figure 3 depicts the result of applying the asset-processes archetype for the asset student, the leftmost node of Figure 2. Similar results were produced by applying the asset-processes archetypes to the remaining assets i.e. lecturers, programs and instructional materials, facility, IT infrastructure (i.e. e-learning platforms), and external universities.
- **Step 3**: Similarly, for each acquire, maintain and retire process identified in step 2, we identified the assets needed for its execution. Again this was achieved by applying the process-assets archetype. Figure 4 is the results produced by applying the process-assets archetype to the marketing process, the leftmost node of Figure 3. Similar models were produced by applying the process-assets archetype to the remaining processes i.e. recruit and select students, manage students enrolment, student counselling, and manage student graduation.

Steps 2 and Step 3 is repeated for each asset and processes respectively, until when no more processes and assets could be identified.

4.3 Phase 3. Evaluation of the Fractal Process-Asset (FPA) Model

In this phase, the produced educational process architecture was used to evaluate FPA model to determine its suitability. It is during this phase that the second goal of the project was achieved. The evaluation of the model was done through presentation and interview with seven business domain experts.

After completing the design of the process architecture, it was presented to the *business domain experts* (both teachers and administrative staff) individually. The presentation consisted of showing the process architecture. This was done first by showing how to start identifying the assets needed by the main process, in this case, teaching and learning, and then going through the steps of identifying acquire, maintain and retire processes for each asset.

On completion of the presentation to individual business domain expert, the model was validated through a semi-structured interview [9]. An open-ended questionnaire guided the interviews. The results of validation are overviewed in Section 6.

5 The Educational Process Architecture: Results and Explanation

The results of the project include the educational process architecture that shows all major business processes and their interconnection, and detailed description of each process performed at the department. Due to space limitation we present and discuss only a part of the educational process architecture.

5.1 Applying Process-Assets Archetype for the Main Process

The results of applying the process-assets archetype (step 1 of section 4.2) for the main process *teaching and learning* is depicted in Figure 2. From the study, it was learnt that to effectively run the teaching and learning, DSV requires several assets to be available. One of the primary assets that must be available for the main process to run is a student. The *student* is the benefiting stakeholder of the service being offered by the department. The main process is run by *lecturers,* a workforce asset that delivers knowledge to the student. The lecturers require *programs and instructional materials*, an asset that includes the descriptions of programs and courses, and instructional materials for each specific course.

Teaching and learning activities requires infrastructure to make the process of learning effective. This infrastructure includes the teaching *facility* provided by the department, e.g., classrooms and offices and the equipment required for teaching and learning. Another type of the infrastructure utilized by the department to support teaching and learning is *IT Infrastructures*. The IT infrastructure asset refers to the hardware, software applications, network resources and services needed to support the teaching and learning process. The software applications include e-learning platforms.

As part of its organizational infrastructure, the department collaborates with other SU departments as well as *external universities* as partners towards achieving its goals.

In the next section we extend the results presented in Figure **2**, by applying the asset-processes archetype to one of the assets, more exactly to the *student* asset.

5.2 Applying the Asset-Processes Archetype to the Student Asset

Applying the asset-processes archetype to the leftmost node of Figure **2** we get its instantiation as depicted in Figure **3**. From the study, we have identified several business processes for acquiring, maintaining and retiring students.

Acquire Processes: Processes utilized by the department to acquire students include, *marketing, recruit and select students, manage student's enrolment*. The *marketing* process aims at attracting more students. It includes activities such as advertising education programs in the SU Catalogue and on the web, and various marketing seminars. *Recruit and select students* aims at getting students for a specific academic year for various programs offered by the department. It includes activities related to announcing, receiving applications, evaluating student applications, selecting students and sending offers to selected students. The *students' enrolment* process is related to administration of the accepted offers - once a student accepts the department's offer, the department has to manage their enrolment.

Maintain Processes: The processes for maintaining students include the *student counselling*, which is designed to help students to develop their self-knowledge and awareness of options for selecting academic programs or courses. Another process in this group is *manage student complaints and appeals*.

Retire Processes: *Manage student graduation* is the retire process which include activities related to application/petition for graduation, degree audit and course waivers and substitutions. It also includes activities related to notifying students of their graduation statuses. Prepare graduation roster and certificates.

The material discussed in this section is part of the results produced by executing step 2 from section 4.2. Similar models were produced by applying the asset-processes archetypes to the remaining assets i.e. lecturers, programs and instructional materials, facility, IT infrastructure (i.e. e-learning platforms), and external universities.

In the next section we extend the results presented in Figure **3**, by applying process-assets archetype to one of the processes, namely, *marketing*.

5.3 Applying the Process-Assets Archetype to the Marketing Process

Applying process-assets archetype to the leftmost node of Figure **3**, we get its instantiation as depicted in Figure **4**. From the study, we have identified that the marketing process at the department makes use of marketing materials, which include *academic programs offered* as a *value proposition*. The assets *lecturers* and *e-learning platforms* are used as the *reputation* of the department to attract students and stakeholders. To perform the marketing activities, the department requires *marketing personnel* and the *marketing process definition*, which describes the procedure of how the marketing process is to be performed. The assets *programs and instructional materials*, *lecturers* and *e-learning platforms* are general assets types because they are the same as for the main process (teaching and learning).

Figure **4** is the results produced by executing step 3 of section 4.2 to the marketing process. Similar process map can be produced by applying the process-assets archetype to the remaining processes i.e. recruit and select students, manage students enrolment, student counseling, and manage student graduation.

6 Analysis of the Results and Lessons Learned - Summary

In this section, we present the analysis of the results and the lessons learned from applying the Fractal Process-Assets (FPA) model. Specifically, the analysis is seeking to establish:

a. Whether the FPA model could be applied to build a process architecture in practice in a resource efficient way
b. Whether the model can reveal or explicate important facts about the business
c. Whether the produced process architecture could be understood and appreciated by domain specialists.

The analysis is based on the following:

- Own reflections of the *authors* over their experience from the project. This is used to answer the first question (a).
- Own reflection of the authors, which was formulated as questions to business domain experts and then confirmed by the latter during the interviews after presentation of the process architecture to them. This answers the second question (b).
- Interview with business domain experts. This is used to answer the *third question* (c).

Question (a). We arrived to the positive answer when considering the following *self-reflections*.

Designing the educational process architecture consisted of two main phases: investigation of business processes and the actual modeling of the process architecture.

- *Investigation of business processes was the difficult part*. The investigation was done through interviews and document analysis. We interviewed nine business domain experts and the interview took approximately one hour for each participant on different working days. Whereas the analysis of the interview took approximately twenty four hours. The analysis of documents took approximately twenty hours. The investigation and modeling phases were iterative. During the modeling we also performed some follow-up interviews for further clarification, which took at least one hour. Therefore, interviews, analysis of the interviews and document analysis took approximately eighty hours in a span of two months.
- *Actual modeling of the educational process architecture*. Applying the FPA model to design educational process architecture was relatively easy. This took approximately forty hours. The most important thing to do was to be able to identify the assets by applying the process-assets archetypes and apply the asset-processes archetype to identify processes for acquiring, maintaining and retiring the respective assets. Applying the modeling tool (Insightmaker [10]) was also

easy. In Insightmaker, a ghost concept is used to avoid crisscrossing of links and flows. Ghosting allows making a reference to a primitive in the model, which are shown with a partially transparent graphical style. The ghost primitive was quite convenient for identifying assets used in several different processes.

Most of the major business processes related to teaching and learning process were well captured in the model to the degree sufficient to provide a holistic view on business processes in the department. However, it was difficult to represent some of the processes and assets that could be of importance. One of the processes that were difficult to model is the process for acquiring "alumni society". Despite the fact that the department makes use of the alumni society as an asset for marketing, it was not clear how, who and when the asset is created and maintained at the department. This requires further investigations with business domain experts. Another process that was not captured is the process for acquiring financing. Despite the fact that the department makes use of the financial asset to acquire and maintain other assets such lecturers, facility, and IT infrastructure, it was difficult to represent it in the process architecture. Finance as an asset is not directly needed to run the teaching and learning process, however, it is produced since the department gets finances from the government by providing teaching and learning. Therefore, one possible way is to represent the *finance* as an asset produced by the teaching and learning process. With this, we propose to use an arrow pointing from the teaching and learning process to the finance as an asset.

From these discussions, it is evident that applying the FPA model to design the process architecture of an enterprise does not require extensive resources. However an initial effort is required to understand how a particular enterprise operates. Once one understands well the operation of an enterprise in question, the actual application of the FPA model to create the process architecture is relatively fast and easy.

Question (b). We arrived to the positive answer based on own reflections and validation results by business domain experts through the interviews. The produced educational process architecture revealed important and useful business facts. The following are some of the important facts revealed from the process architecture shown in **Figure 6** (a part of the complete educational process architecture):

- While the primary purpose of the *lecturer recruitment* process is to provide the department with qualified teachers, the process architecture reveals that the same process is intended to provide the marketing with competitive advantages of having highly qualified staff, e.g. full professors, Nobel laureates, etc.
- Similarly, while the *academic program development* is primarily aimed at providing the department with quality academic programs and instruction materials for teaching and learning, the process architecture reveals that it also provides marketing with competitive advantages of having attractive programs and high quality teaching materials.

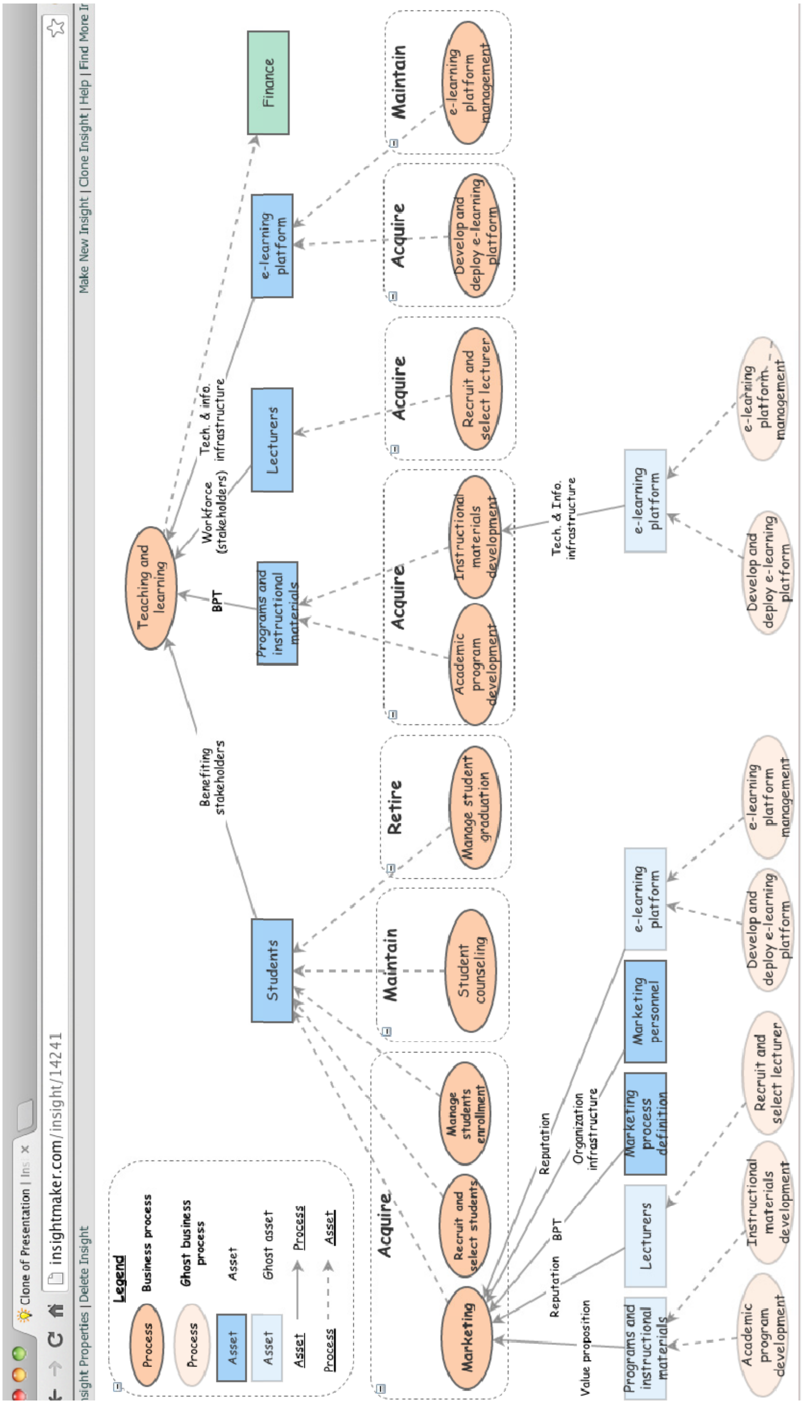


Fig. 6. An instantiation of the process-assets and asset-processes archetype

- This was also the case with the *develop and deploy e-learning platforms* process. While the process aimed at providing teachers and students with a platform for educational delivery and management, the process architecture revealed that it also (1) provides teachers with tools for instructional materials development (2) provides marketing with competitive advantages of having a high quality educational delivery platform.

The above facts were validated by *business domain experts* through semi-structured interviews that followed the presentation of the process architecture. In addition, more facts were revealed during the interviews. For example, one of the experts revealed that the lecturer recruitment process is also intended to provide the department with highly qualified researchers for the research process. While the scope of the study only focused to the teaching and learning process, this revelation shows how the FPA model could be useful to provide a holistic view of all processes in an enterprise.

Question (c). We arrived to the positive answer based on the responses from the interviews with business domain experts. The results of the interview, with our *business domain experts*, which directly followed the presentation, show that the educational process architecture is well understood and appreciated by domain specialists. More specifically, the interviews were aimed to determine the following:

- Whether it is important to make explicit all purposes of all processes in the department. The results show that 100% (56% agree and 44% strongly agree) of the business domain experts agree that it is important to make explicit all purposes of all processes in the department.
- Whether the FPA model is useful for explicating business processes and their interconnection. The results show 100% agree that the FPA model as being useful for explicating processes and their associated purposes.
- Whether the visual diagram that shows how all processes in the department are inter-connected is useful for business planning and development. The results show that 89% (56% agree, 33% strongly agree) agree that the visual diagram that shows how all processes in the department and their inter-connection is useful for business planning and development and 11% are not sure. When asked why he was not sure, an expert said he would be sure after applying the diagram to practical business development.
- Whether the presented visual diagram could be useful. The results show that 89% (56% agree, 33% strongly agree) agree that the presented diagram of putting business process inter-connections is useful for business planning and development and 11% are not sure. Similarly, when asked why he was not sure, an expert said he would be sure after applying the diagram in practice.

7 Conclusion

This paper reports on a project of applying the Fractal Process-Assets (FPA) model in the frame of a real organization – the department of Computer and Systems Sciences (DSV), Stockholm University. The first goal of the project was to create a holistic view on the major business processes and their interconnections to be used for business planning and development. To achieve this goal, the educational process architecture has been created and presented to business domain experts for validation as the second goal of the project.

The analysis from Section 6 shows that the FPA model is suitable for creating a holistic view on the major business processes in an enterprise and their interconnections. The educational process architecture produced is understandable and can be used for business planning and development. Though the study has been conducted only in one organization, there is a likelihood that the results achieved are of general nature.

We believe that our experience report could be of interest for a wider audience. The FPA model in enterprise modeling is a relatively new area, and there is not that much experience on its application reported in the literature. Therefore, the example and discussions presented in this paper may be of use for any researcher or practitioner interested in modeling the process architecture of an enterprise.

Acknowledgements. We are grateful to DSV and the colleagues, Stefan Moller, Maria Bergholtz, Gustaf Juell-Skielse, Tuija Lehtonen, Jan Moberg, Erik Perjons, Sofia Mattsson, Carina Bergholm, Sundén Andrea, and Joakim Snygg who participated in the project as *business domain experts*.

References

[1] Wikipedia Contributors, http://en.wikipedia.org/wiki/Business_process (last accessed March 2014)

[2] Flores, M., Cabello, A., Torredemer, L., Agrawal, M., Keast, J., Terzi, S., Sopelana, A.: Do enterprises implement a process architecture towards Lean in product development? A comparative study among large and small firms. In: 17th International Conference of Concurrent Enterprising (ICE), Aachen, Germany, pp. 1–9 (2011)

[3] Flores, M., Flores, K., Cabello, A., Terzi, S., Rossi, M.: Understanding the approaches to create a process architecture for lean thinking. In: 18th International Conference on Engineering, Technology and Innovation, pp. 1–10. IEEE (2012)

[4] Frolov, V., Mengel, D., Bandara, W., Sun, Y., Ma, L.: Building an ontology and process architecture for engineering asset management. In: Kiritsis, D., Emmanouilidis, C., Koronios, A., Mathew, J. (eds.) Engineering Asset Lifecycle Management, pp. 86–97. Springer, London (2010)

[5] Bider, I., Perjons, E., Elias, M.: Untangling the Dynamic Structure of an Enterprise by Applying a Fractal Approach to Business Processes. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) PoEM 2012. LNBIP, vol. 134, pp. 61–75. Springer, Heidelberg (2012)

[6] Bider, I., Bellinger, G., Perjons, E.: Modeling an Agile Enterprise: Reconciling Systems and Process Thinking. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 238–252. Springer, Heidelberg (2011)

[7] Scheer, A.-W.: ARIS: business process modeling. Springer, Alemania (2000)

[8] Hammer, M., Stanton, S.: How process enterprises really work. Harvard Business Review 77, 108–120 (1999)

[9] Denscombe, M.: The Good Research Guide: For Small-Scale Social Research Projects: For small-scale social research projects. McGraw-Hill International (2010)

[10] Give Team, http://insightmaker.com/ (last accessed March 2014)

Support for Domain Constraints in the Validation of Ontologically Well-Founded Conceptual Models

John Guerson, João Paulo A. Almeida, and Giancarlo Guizzardi

Ontology and Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo (UFES), Vitória ES, Brazil
{jguerson,jpalmeida,gguizzardi}@inf.ufes.br

Abstract. In order to increase the accuracy of conceptual models, graphical languages such as UML are often enriched with textual constraint languages such as the Object Constraint Language (OCL). This enables modelers to benefit from the simplicity of diagrammatic languages while retaining the expressiveness required for producing accurate models. In this paper, we discuss how OCL is used to enrich a conceptual model assessment tool based on an ontologically well-founded profile of the Unified Modeling Language (UML) that assumes multiple and dynamic classification (called OntoUML). In the approach, OCL expressions are transformed into Alloy statements enabling model validation and assertion verification with the Alloy Analyzer. The tool we have developed allows modelers with no Alloy expertise to express constraints in OCL enriching OntoUML models.

Keywords: Conceptual Model Validation · Domain Constraints · OCL · Alloy · OntoUML.

1 Introduction

Conceptual modeling is "the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication" [16]. A conceptual model, in this sense, is a means to represent what modelers perceive in some portion of the physical and social world with purpose of supporting the understanding (learning), problems solving and communication, in other words, a means to represent the modeler's conceptualization [11] of a domain of interest.

For a number of years now, there has been a growing interest in the use of Foundational Ontologies (i.e., ontological theories in the philosophical sense) for supporting the activity of Conceptual Modeling giving rise to an area known as Ontology-Driven Conceptual Modeling. In this setting, the OntoUML language has been designed to comply with the ontological distinctions and axiomatic theories put forth by a theoretically well-grounded Foundational Ontology [11]. This language has been successfully employed in a number of industrial projects in several different domains such as Petroleum and Gas, News Information Management, E-Government, Telecom, among others.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 302–316, 2014.

OntoUML was designed to address a number of deficiencies in UML from a conceptual modeling standpoint. OntoUML addresses a number of problems in UML regarding ontological expressivity, i.e., the ability of a language to make explicit ontological distinctions present in a domain [11]. These distinctions are important to ensure that the modeler may express as accurately as possible [10] a domain conceptualization, making conceptual models more useful to support the understanding, agreement and perhaps, the construction of information systems.

Despite the advances in the quality of conceptual modeling languages, assessing whether a conceptual model indeed reflects the modeler's intended conceptualization remains a challenging task. In order to support the validation of conceptual models in OntoUML, *Benevides et al.* [2] and *Braga et al.* [3] defined a translation from OntoUML conceptual models to Alloy [12]. The idea is to use the Alloy Analyzer to automatically generate logically valid instances for the OntoUML model at hand. By confronting the user with a visualization of these possible model instances, we are able to identify a possible gap between the set of possible model instances (implied by the model) and the set of intended model instances (which the modeler intended to capture). In other words, in this approach, one can detect cases of instances which conform to the OntoUML model but which do not reflect the modeler's intended conceptualization (due to under-constraining) as well as cases of intended possible instances which are not shown as valid ones (due to over-constraining).

Up to this point, the validation of OntoUML models in this approach has been limited to the formulae implied from its diagrammatic notation. However, in complex domains, there are typically a number of domain constraints which cannot be directly expressed by the diagrammatic notation of the language, but which are of great importance for capturing as accurately as possible the modeler's intended domain conceptualization. In order to address this issue, in this paper, we propose the use of OCL expressions as a mean to enhance the expressivity of OntoUML conceptual models with respect to the explicit representation of domain constraints.

This paper extends the approaches of *Benevides et al.* [2] and *Braga et al.* [3] by defining a translation from OCL to Alloy in compliance with the existing transformation of OntoUML. The OCL subset considered is then determined by the expressivity and significance to the OntoUML modeling language. One of the key differences of OntoUML (and as a consequence of our approach) is that it has a full support for dynamic and multiple classification. Although dynamic and multiple classification are in principle supported by UML class diagrams, most approaches that establish formal semantics and analysis/simulation for these diagrams do not address these features[1]. This renders these approaches less suitable to enable the expression of important conceptual structures that rely on dynamic classification (e.g., the classification of persons into life phases: child, teenager, adult; the classification of persons into roles in particular contexts) as well as multiple classification (e.g., the classification of persons according to orthogonal classification schemes such as: living-deceased, male-female).

[1] Probably, due to the strict correspondence that is often established (even if implicitly) between conceptual modeling languages and programming languages that lack such features.

We define a fully-automated translation from OntoUML+OCL models and implement and incorporate into an OntoUML modeling environment. The tool we have developed allows modelers with no Alloy expertise to write constraints in OCL enriching OntoUML models. In addition to instantiating model instances for model simulation, the tool supports the formal verification of assertions written in OCL. Our overall objective is to support the assessment of conceptual models, retaining the simplicity of a diagrammatic language while coping with the expressiveness required to produce accurate conceptual models.

This paper is organized as follows. Section 2 presents a running example based on a traffic accident ontology in OntoUML and OCL. Section 3 presents the existent transformation from OntoUML to Alloy. Section 4 and 5 describes and applies our OCL translation to validate the running example. Section 6 discusses related work and Section 7 presents some concluding remarks.

2 A Conceptual Model in OntoUML

This example was inspired by a governmental project that we conducted for a national regulatory agency for land transportation in Brazil. In this domain, travelers are persons taking part of a travel in a vehicle, possibly becoming involved in traffic accidents; traffic accidents involve victims, crashed vehicles and a roadway; and, accidents may involve a number of fatal victims. A particular sort of accident called rear-end collisions is also identified (accidents wherein a vehicle crashes into the vehicle in front of it). **Fig. 1** depicts an OntoUML conceptual model for that domain.

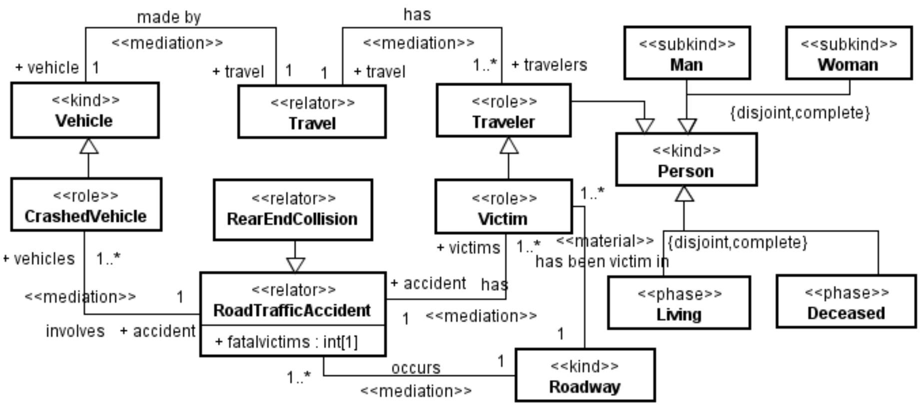


Fig. 1. OntoUML conceptual model of road traffic accidents

OntoUML extends UML by introducing metaclasses that correspond to ontological distinctions put forth by the Unified Foundational Ontology (UFO). For instance, a class stereotyped as a *kind* provides a principle of application and a principle of identity for its instances [11]. It represents a *rigid* concept, i.e., a class that applies necessarily to its instances (e.g., a Person cannot cease to be a Person without ceasing to exist). A kind can be described in a taxonomic structure where its subtypes are also rigid types known as subkinds (e.g., Man and Woman).

A *role,* in turn, is an *anti-rigid* concept, applying contingently to its instances (e.g., a Person can cease to be a Traveller and still exist) and does not provide a principle of identity, instead, it inheres the identity from the unique kind it specializes. A role is also *relational dependent*, i.e., it defines contingent properties exhibited by an instance of a kind in the scope of a relationship (e.g., John plays the role of Victim in an accident contingently and in relation to or, in the context of, that accident).

A *phase* is an anti-rigid concept that it is defined by a partition of a kind and whose contingent instantiation condition is related to intrinsic changes of an instance of that kind (e.g. if Living and Deceased constitutes a Person's phase partition then every Person x is either alive or deceased, but not both. Moreover, a Living Person is a Person who has the intrinsic property of being alive) [11].

A *relator* (e.g. entities with the power of connecting other entities) is a rigid concept and *existentially depends* on the instances it connects through *mediation* relations (e.g., an Accident only exists if Crashed Vehicles, Victims and a Roadway also exist). From an ontological point of view, relators are the truthmakers of the so-called *material* relations. For instance, it is the existence of a particular RoadTrafficAccident connecting Victim X, Crashed Vehicle Y and Roadway Z that makes true the relation has-been-victim-in-roadway(X,Z). In OntoUML, material relations such as this one are considered to be logico-linguistic construction reducible to an analysis of relators, relata and their tying mediation relations. Some UML features such as Bags only occur in an OntoUML model at this linguistic level, i.e., in the context of derived material relations. Other purely linguistic features of UML are dispensed altogether in OntoUML. These include the notion of interface but also the ordered UML meta-attribute (and, consequently, the Ordered Sets or Sequences collections types).

Fig. 2 presents a possible instantiation (simulation) of our model of traffic accidents. It shows a state (current world) where a person (Object1) is classified as Woman and Living and the other person (Object4) as Man and Living, characterizing an example of multiple classification in OntoUML. Both persons, the man and the woman, play the role of travellers in a travel made by the vehicle Object5, crashed in an

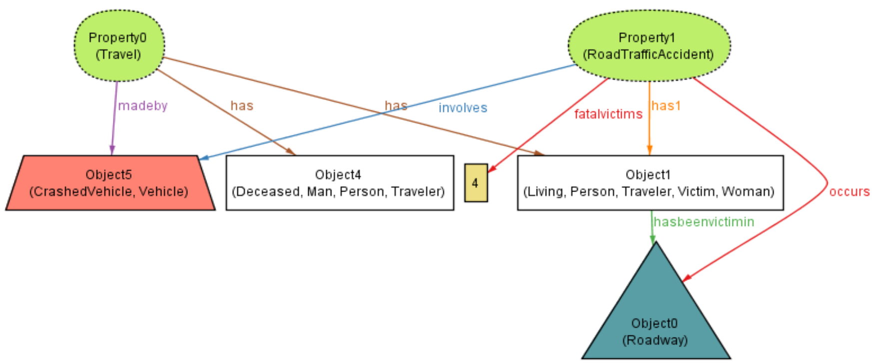


Fig. 2. Automatically generated instantiation without constraints

accident. The simulation shows some situations that may contradict the intended conceptualization for this domain, e.g., that the accident has 4 fatal victims although only one non-fatal victim is identified (Object1) and that the man is a traveller of the vehicle involved in the accident (Object5) but is not a victim of that accident.
Note that this simulation result was generated automatically by the Alloy Analyzer (using the existing approach derived from the work of *Benevides et al.* [2] and *Braga et al.* [3]), and thus, using that approach, we are unable to prevent the inadmissible situations from occurring. That will be addressed in our approach by adding domain constraints to the OntoUML model, which will be expressed in OCL and transformed into the Alloy language.

OCL [18] is a (*semi*) formal [5, p.60] language adopted as a standard by the OMG to represent constraints on MOF-based models. It is declarative, textual and based on first-order logic. OCL is used for a variety of purposes such as to express class invariants, attributes and association-end point derivations, query operations definitions and pre- and post-conditions over operations. The subset of OCL considered here includes both invariants and derivations. An invariant is a condition applied to a class in the model. The condition must be true for every class' instance at any point in time, in OntoUML terms, at every world instance, whether past, current, counterfactual or future world [2]. A derivation, in turn, expresses how an association end-point or an attribute can be inferred from other elements of the model. As a structural conceptual modeling language, OntoUML does not target the representation of operations, thus, as a consequence, we do not support the definition of operations and pre- and post-conditions. It is important to emphasize that no change is required in OCL so that it can be used with OntoUML; a subset of OCL can be meaningfully employed to a lightweight extension of UML.

Fig. 3 shows three OCL constraints for the traffic accident domain. The first invariant states that every rear-end collision must involve exactly two crashed vehicles; the second constraint (a derivation) specifies that the attribute *fatalvictims* is derived from the number of deceased victims in an accident; and the third and last constraint (an invariant) states that (i) every traveler of a vehicle participating in an accident is a victim in that accident, and that (ii) every vehicle, in which there is a victim of an accident, is involved in that accident.

```
context RearEndCollision inv: self.vehicles->size() = 2

context RoadTrafficAccident::fatalvictims: int
derive: self.victims->select(p | p.oclIsKindOf(Deceased))->size()

context RoadTrafficAccident inv: self.vehicles->forAll(v |
self.victims.oclAsType(Traveler)->includesAll(v.travel.travelers))
and self.victims->forAll(p |
self.vehicles->includes(p.travel.vehicle.oclAsType(CrashedVehicle)))
```

Fig. 3. OCL domain constrains for the road traffic accident conceptual model

3 From OntoUML Models to Alloy Specifications

The approach proposed by *Benevides et al.* [2] and *Braga et al.* [3] to support the validation of OntoUML conceptual models uses a lightweight formal method of validation by defining a (semantic preserving) transformation from OntoUML models to Alloy. The resulting Alloy specification is fed into the Alloy Analyzer tool to generate and visually confront the modeler with possible instances of the model. The set of atoms displayed represent instances of the classes of the OntoUML model and the set of relations between those atoms represent instances of the OntoUML relationships.

Alloy [12] is a declarative and first order logic-based language to describe structures accompanied by a tool to explore and display them graphically. An Alloy specification defines the possible structures of atoms and relations. It is comprised mainly of: signatures with fields and constraints (facts, assertions and predicates). An Alloy signature introduces a set of atoms with relations between them declared as fields of signatures [12, p.35]. An Alloy *fact* is a constraint that must always be respected by the structure of atoms and relations. An Alloy *assertion* is a target for verification, i.e., a boolean expression that the Alloy Analyzer will try to invalidate by examining structures allowed by the specification [12, p.93, 119]. The Alloy Analyzer will either conclude that the assertion is invalid, showing a counterexample for it (a structure that invalidates it), or conclude that it holds for structures up to a certain size (the *scope* of verification). An Alloy *predicate* is a boolean expression that can be used in different contexts, e.g., within facts or within commands for verification and simulation.

Fig. 4 shows part of the Alloy code generated by the translation of the OntoUML model of **Fig. 1**. In line 5, the Alloy signature *Object* represents existentially *independent* entities (e.g. instances of kinds, roles, phases, subkinds). In line 6, existentially *dependent* entities (objectified properties, e.g., relators) are represented by the signature *Property*. In line 7, the abstract signature World represents the states of objects and reified properties. This is required to support the notion of modality that underlies OntoUML and thereby model the dynamics of creation, classification, association and destruction of instances. In each World, Objects and Properties may exist, which is specified using the *exists* field (line 8). Worlds are classified into four sub-signatures: *CurrentWorld*, *PastWorld*, *FutureWorld* and *CounterfactualWorld*. These sub-signatures are specified in a separated module imported as an Alloy library to the specification [2] (line 2). In line 9, the *kind* Person is transformed into a binary relation between the World and the object (instance of person) that exists in that World. The rigidity property of persons is represented by a predicate declaration within a fact statement, as showed in line 15. The rigidity predicate is part of a separated module (imported as a library in line 3), which is committed to specify several ontological properties of OntoUML. Similarly, in line 10, the *role* Victim is transformed into a binary relation between the World and the object existing in that World. In line 11, the *relator* RoadTrafficAccident is transformed into a binary relation between the World and the corresponding objectified property that exists in that World. All the classes in OntoUML follow this transformation to Alloy, i.e., they are Alloy binary relations from worlds to extensions, which allows us to capture dynamic classification. Furthermore, in line 12, the attribute *fatalvictims* is represented as a ternary relation (a triple) between the World, the owner of the attribute and its type (in this case Int). In line 13, the mediation *has* is represented as a ternary relation between the World, the Accident, and the Victim. In addition, the existential dependency between the Accident and its Victims is represented by another imported ontological property (a predicate) enforcing the immutability of victims in that accident (line 16). Finally, in line 18 and 19, the association end-point *victim* is represented as an Alloy function which receives as parameters the traffic accident, from which the association end-point is reached, and the world instance in which it exists, returning the set of victims related to that accident.

```
1   …
2   open world_structure[World]
3   open ontological_properties[World]
4   …
5   sig Object {}
6   sig Property {}
7   abstract sig World {
8       exists: set Object + Property,
9       Person: set exists:>Object,
10      Victim: set exists:>Object,
11      RoadTrafficAccident: set exists:>Property,
12      fatalvictims: set RoadTrafficAccident set -> one Int,
13      has1: set RoadTrafficAccident one -> some Victim,
14   … }{ … }
15   fact { rigidity[Person,Object,exists] }
16   fact { immutable_target[RoadTrafficAccident,has1] }
17   fun victims[x: World.RoadTrafficAccident, w: World] : set World.Victim
18   { x.(w.has) }
19   …
```

Fig. 4. Resulting Alloy specification from OntoUML

4 From OCL constraints to Alloy constraints

In this section, we define the translation of OCL constraints in Alloy. We assume the transformation of OntoUML to Alloy discussed in the previous section. We use the symbol [[ ]] to denote a function that receives OCL concrete syntax and returns Alloy textual code.

4.1 Invariants and derivations

OCL invariants and derivations are represented in Alloy as facts with formulae which hold in every possible World for all instances of the Context class, as shown in **Table 1**. The body of an invariant is directly transformed into the body of the corresponding Alloy fact. Derivations in turn force the values of attributes and association ends to match the derivation expression. Note that the Alloy counterpart of OntoUML classes, attributes and associations are World fields, referred in the mappings by the expression *w.[[Class]]*, *w.[[attribute]]* and *w.[[association]]*, respectively. The association end-points are represented as functions which receive as parameters the source object from which the association end-point is reached and the world instance in which it exists. They are referred in the mappings using the Alloy function syntax: *self.[[assocEnd]][w]*, where *assocEnd* represents a function name corresponding to an association end and *self* and *w* are function parameters.

Table 1. Translation of OCL invariants and derivations

OCL constraint	**Alloy statement**
context *Class* inv: *OclExpression*	fact invariant1 { all w: World | all self: w.[[*Class*]] | [[*OclExpression*]] }
context Class::*attribute*:*Type* derive: *OclExpression*	fact derive1 { all w: World | all self: w.[[*Class*]] | self.(w.[[*attribute*]]) = [[*OclExpression*]] }
context Class::*assocEnd*:Set(*Type*) derive: *OclExpression*	fact derive2 { all w: World | all self: w.[[*Class*]] | self.[[*assocEnd*]][w] = [[*OclExpression*]] }

4.2 Expressions

OCL expressions are divided into: if-then-else expressions, let-in expressions, navigational expressions (using the “dot notation”) and operation call expressions. The former two can be directly represented in Alloy by equivalent expressions whilst the last one is not considered here since operations are not meaningful in OntoUML. Navigational expressions deserve special treatment as there are different mappings for attribute access and association end navigation. In **Table 2**, we define the mappings for OCL expressions. We use *be* to represent a boolean expression, *expr* to represent an OCL expression, *battr* to represent a boolean attribute, *var* to represent a variable. The dot notation is equivalent in both OCL and Alloy, thus the only difference in the attribute mappings stem from the fact that an OntoUML boolean attribute is represented as an Alloy subset, therefore, the OCL dot operation in this case is mapped to the Alloy operator *in* (the same mapping choice as taken in [6]).

Table 2. Translation of OCL expressions

OCL expression	**Alloy expression**
if *be* then *be1* else *be2* endif	[[*be*]] implies [[*be1*]] else [[*be2*]]
let *var*: Type = *expr* in *be*	let *var* = [[*expr*]] | [[*be*]]
expr.attribute	[[*expr*]].(w.[[*attribute*]])
expr.assocEnd	[[*expr*]].[[*assocEnd*]][w]
expr.battr	[[*expr*]] in (w.[[*battr*]])

4.3 Iterators

Table 3 shows the mappings from OCL iterators into Alloy. The word *col* represents OCL expressions that result in collections and the letter *v* represents variables.

Table 3. Translation of OCL iterators

OCL iterator	**Alloy expression**
col->forAll(v1,..,vn | *be*)	all v1,..,vn: [[*col*]] | [[*be*]]
col->exists(v1,..,vn | *be*)	some v1,..,vn: [[*col*]] | [[*be*]]
col->select(v | *be*)	{ v: [[*col*]] | [[*be*]] }
col->reject(v | *be*)	{ v: [[*col*]] | not [[*be*]] }
col->one(v | *be*)	#{ v: [[*col*]] | [[*be*]] } = 1
col->collect(v | *expr*)	univ.{ v: [[*col*]], res: [[*expr*]] | no none}
col->isUnique(v | *expr*)	all disj v, v': [[*col*]] | [[*expr*]](v)!=[[*expr*]](v')
col->any(v | *be*)	{ v: [[*expr*]] | [[*be*]] }
col->closure(v| *expr*)	[[*col*]].^{v: univ, res: [[*expr*]] | no none}

OCL iterators are represented in Alloy as quantified formulae and comprehension sets. The *forAll* and *exists* iterators are represented as Alloy formulae quantified universally (keyword *all)* and existentially (keyword *some).* The *select* and *reject* iterators are represented as Alloy comprehension sets (denoted by curly brackets) whilst the *one* iterator is also represented as an comprehension set but using operators such as # (cardinality operator) and = (equality operator) to state that the resulting set must be equal to 1. The *collect* iterator is represented combining comprehension sets, the keyword *univ*, the dot notation and a logical *true* Alloy primitive value (expressed in terms of the keywords *no none).* The *isUnique* iterator is represented as an Alloy formula universally quantified plus the disjointness keyword *disj.* The *any* iterator is represented by an Alloy comprehension set but with a restriction of usage: the modeler must ensure that the boolean expression evaluates to true in exactly one element of the source collection (the same mapping as in [13]). Finally, the *closure* iterator is represented combining comprehension sets, the transitive closure operator (^) and the Alloy true primitive value, similar to the *collect* mapping previously presented.

4.4 Sets

Alloy supports all the OCL set operations since it is a set-based language. Therefore, the OCL set operations represented in Alloy are: *size*, *isEmpty*, *notEmpty*, *includes*, *excludes*, *includesAll*, *excludesAll*, *union*, *intersection*, *including*, *excluding*, *difference*, *symmetricDifference*, *asSet*, *product, sum* and *flatten*. We omit here these mappings since they are rather straight-forward given Alloy's native support for sets.

4.5 Primitive types

Alloy natively supports only the integer and boolean primitive types. They are directly represented in Alloy as well as their operations. However, Alloy does not natively support the OCL *xor* boolean operator and the OCL integer operations *max*, *min* and *abs.* Their mappings to Alloy are shown in **Table 4.** . The supported OCL boolean operations are: *and*, *or*, *implies*, *not* and *xor*; whilst the supported OCL integer operations are the comparison operations (i.e., <, >, <=, >=) as well as some arithmetic ones (i.e., + (sum), - (subtraction), * (multiplication), *div*, *floor*, *round*, *max*, *min* and *abs*). Bit width for integers is by default 7 (and thus range by default from -63 to 64).

Table 4. Translation of OCL primitive types

OCL operation	Alloy expression
e1 xor *e2*	([[*e1*]] || [[*e2*]]) && !([[*e1*]] &&[[*e2*]])
e1.max(*e2*)	int[[[*e1*]]] >= int[[[*e2*]]] => [[*e1*]] else [[*e2*]]
e1.min(*e2*)	int[[[*e1*]]] <= int[[[*e2*]]] => [[*e1*]] else [[*e2*]]
e1.abs()	Int[[[*e1*]]] < 0 => [[*e1*]].negate else [[*e1*]]

4.6 Objects Operations and Meta-Operations

Table 5 depicts the object and meta-operations of OCL translated in Alloy where *T* is a type (i.e., a class) in the model. The *oclIsTypeOf* operation means that the object is of the type T but not a sub-type of T. The *oclIsKindOf* operation in turn checks the same condition but including the subtypes of T. The latter is represented by the Alloy subset operator (*in*) whilst the former is represented by an expression combining the operators *in*, *and*, # (cardinality), & (intersection), + (union) and = (equality), verifying if the object is contained in the set T but not in the union of all subtypes of T (referred by the Alloy expression [[subT1]]+…+ [[subTn]]). The *oclAsType* and *allInstances* operations are directly represented by their source parameter since Alloy is a set-based language.

Table 5. Translation of OCL object operations and meta-operations

OCL operation	Alloy expression
obj.oclIsKindOf(*T*)	[[*obj*]] in w.[[*T*]]
obj.oclIsTypeOf(*T*)	[[*obj*]] in w.[[*T*]] and # w.[[*T*]] & (w.[[*SubT1*]] +…+ w.[[*SubTn*]]) = 0
obj.oclAsType(*T*)	[[*obj*]]
obj.oclIsUndefined()	# [[*obj*]] = 0
Class.allInstances()	w.[[*Class*]]

5 Revisiting the Running Example

In this section, we revisit the running example, now enriched with domain constraints. We use the OCL transformation discussed in the previous sections and generate valid instances of the traffic accident conceptual model. We further exemplify the use of OCL invariants as assertions subject to verification.

5.1 Simulation

Fig. 5 depicts the code generated by applying our OCL transformation. The generated code is added into the specification resulting from the transformation of the OntoUML model that which was partially presented in **Fig. 4**. All elaborated OCL domain constraints are transformed into Alloy facts and thus all instantiations of the OntoUML model will conform to these constraints. The final specification resulting from both OntoUML and OCL mappings is fed into the Alloy Analyzer to generate/check sample structures of the OntoUML+OCL model.

```
context RearEndCollision inv: self.vehicles->size() = 2

fact invariant1 { all w: World | all self: w.RearEndCollision |
# self.vehicles[w] = 2 }
```

```
context RoadTrafficAccident::fatalvictims: int
derive: self.victims->select(p | p.oclIsKindOf(Deceased))->size()

fact derive1 { all w: World | all self: w.RoadTrafficAccident |
self.(w.fatalvictims) = # { p: self.victims[w] | p in w.Deceased } }
```

```
context RoadTrafficAccident inv: self.vehicles->forAll(v |
self.victims.oclAsType(Traveler)->includesAll(v.travel.travelers))
and self.victims->forAll(p |
self.vehicles->includes(p.travel.vehicle.oclAsType(CrashedVehicle)))

fact invariant2 { all w: World | all self: w.RoadTrafficAccident |
(all v: self.vehicles[w] | v.travel[w].travelers[w] in univ.{temp1:
self.victims[w], res: temp1 | no none }) && (all p: self.victims[w] |
p.travel1[w].vehicle[w] in self.vehicles[w]) }
```

Fig. 5. Alloy code resulting from our OCL translation

Fig. 6 depicts a possible instantiation of the traffic accident model enrich with its domain constraints. The figure depicts a current world (a point in time) where a road traffic accident (a rear end collision), between two crashed vehicles resulted in the death of both travelers of the vehicles, and where the two fatal victims were both male persons. All specified OCL constraints are respected (in every point in time, whether past, current or future world). Differently from the unconstrained model that was shown in **Fig. 2**, the derived number of fatal victims is correct, the traveler of the crashed vehicle is indeed a victim of that accident and the *rear end collision* involves two crashed vehicles as required in the definition of this type of accident.

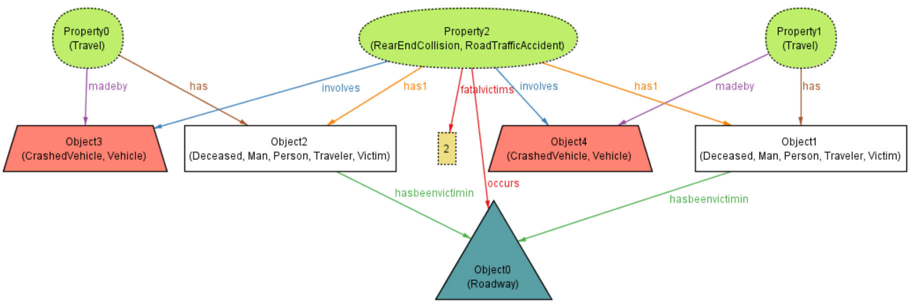


Fig. 6. Automatically generated instantiation with constraints

5.2 Assertion Checking

The same transformation used to generate facts corresponding to OCL invariants can be used to generate assertions which are subject to verification by the Alloy Analyzer. The analyzer will either conclude that the assertion is invalid, showing a counterexample for it (an instance of the model that invalidates the assertion), or conclude that it holds for structures up to a certain size (the *scope* of verification). The approach to verification in Alloy is based on the "small scope hypothesis" which states that, if an assertion is invalid, then it probably has a small counterexample [12, p. 143]. This ensures tractability of assertion verification.

Fig. 7 depicts an assertion written in OCL and its mapping to Alloy. The OCL assertion states that, in a travel, not all travelers are deceased. This is transformed into an assertion plus a check command defining the default scope and the default Alloy bitwidth. Furthermore, in the check command, we also define the number of atoms of the signature World (particularly 1 to ensure a single World atom in the checking).

```
context Travel inv: not self.travelers->forAll(t| t.oclIsKindOf(Deceased))
assert invariant3 {all w: World | all self: w.Travel |
! (all t: self.travelers[w] | t in w.Deceased) }
check invariant3 for 10 but 1 World, 7 Int
```

Fig. 7. OCL assertion and the mapping to Alloy

Fig. 8 shows the counterexample found by executing the check command with the Analyzer, showing thus that the enriched OntoUML model does not guarantee the satisfaction of the assertion. The figure depicts a current world where all travelers of a travel made by a vehicle are actually deceased. The label *$self* means that this particular atom is descendant from the variable *self* in the OCL assertion, where *self* is an instance of Travel. If the modeler intended this assertion to hold (i.e., if he/she believe that the situation is inadmissible in the domain), the OCL expression in the assertion can be considered a fact (an invariant enriching the model), thus preventing this situation from occurring.

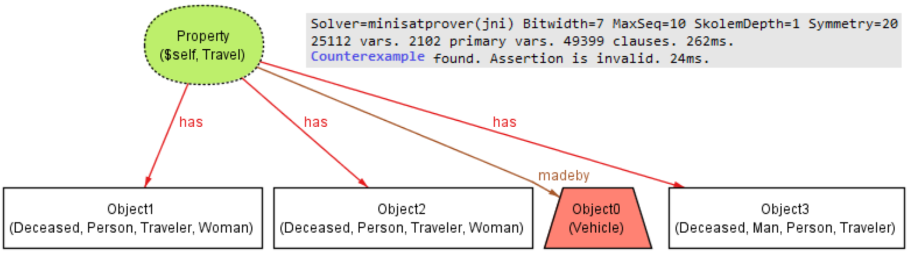


Fig. 8. Counterexample found. Assertion does not hold

6 Related Work

There have been several approaches in the literature to the analysis and validation of UML models and OCL constraints e.g. HOL-OCL [4], USE [9], CD2Alloy [14], UML2Alloy [1]. In particular, a number of these approaches [1], [6], [13], [14], [15] have used Alloy as a lightweight formal method for validating models in UML/OCL. In [1], *Anastasakis et al.* present one of the first extensive approaches for automatic translation of UML+OCL models into Alloy for purposes of model verification and validation. Their tool is called UML2Alloy and although it considers both UML and OCL, it does not support several OCL operators and just a subset of UML is considered. *Cunha et al.* [6] extended the mappings of *Anastasakis et al.* to support, among others, UML qualified associations and dynamics of properties such as the UML *read-only* feature (mutability of properties). They defined a state local signature called Time in the Alloy resulting specification to correctly handle dynamics of properties and pre- and post- conditions. *Kuhlmann et al.* [13] defined a translation from UML and OCL to relational logic and a backwards translation from relational instances to UML model instances (relational logic is the source for the Kodkod SAT-based model instance finder used by Alloy). *Massoni et al.* [15] proposed a transformation of a small subset of UML (class diagrams with classes, attributes and association) annotated with OCL invariants to Alloy. However, they specify the translation only in a systematic and manual way; they do not implement it. Finally, *Maoz et al.* [14] translated UML, particularly class diagrams, to Alloy and then from Alloy's instances back to object diagrams, considering both multiple inheritance and interface implementation. They use a deeper embedding strategy as not all UML concepts are directly translated to a semantically equivalent Alloy construct (for instance, the multiple inheritance feature is transformed to a combination of facts, predicates and functions in Alloy). In addition, they are able to support the analysis of class diagrams, for example, checking if one class diagram is a refinement of some other class diagram [14, p.2]. The translation is fully implemented in a prototype plugin in Eclipse called CD2Alloy, which can (optionally) hide the Alloy resulting specification from the modeler. This translation however does not consider OCL. Besides, the Alloy resulting specification is more difficult to read, less understandable and computationally more complex than other approaches. None of these approaches completely support dynamic and multiple classification, which is essential for ontology-driven conceptual modeling. In fact, besides dynamic and multiple classification, the meta-properties that characterize many of the ontological categories and relations in an ontologically well-founded language are modal in nature. As discussed in [11], the modal distinctions among object types and part-whole relations are paramount from an ontological perspective and play a fundamental role in ontology engineering and semantic interoperability efforts. These modal features (and all language constructs affected by them) require a special treatment in the mapping to Alloy [2] [3]. Our translation of OCL is in pace with all these features.

7 Concluding Remarks

In this paper we have presented an approach to validate OCL-enhanced OntoUML models using a lightweight formal method that uses Alloy for model visual simulation and model checking. We have extended the previous work of *Benevides et al.* [2] and *Braga et al.* [3] by defining a translation from OCL constraints into Alloy statements in accordance with the existent transformation of OntoUML. This allows modelers with no Alloy expertise to write constraints in OCL enriching OntoUML models. This work contributes to facilitating the definition of high-quality conceptual models that, albeit grounded on sound ontological distinctions, lacked several domain constraints and did not cover precisely [10] the modeler's intended conceptualization. The approach supports visual simulation of model instances that conform to the enriched OntoUML model as well as supports checking of assertions through model checking.

The translation to Alloy discussed here is fully implemented and incorporated into the OntoUML Lightweight Editor (OLED[2]) developed in our research group. The Alloy code fragments presented in **Fig. 4** and **Fig. 5** are indeed part of the specification as generated by OLED. OLED is an experimental tool for the OntoUML conceptual modeling language that provides instance simulation (via Alloy and its Analyzer) along with other features such as syntax verification, model editing, model verbalization and model transformations (e.g., to languages such as OWL). OLED manipulates OntoUML models using an OntoUML Eclipse metamodel [7][17]. We have employed the Eclipse MDT OCL [8] plugin for OCL syntax verification, auto-complete, parsing and to implement the OCL mappings to Alloy using the visitor pattern. The infrastructure for OCL manipulation and binding to OntoUML is currently being used in order to implement a transformation of OCL to SWRL, building up on an OntoUML to OWL transformation, and enabling the use of OCL constraints for (runtime) inference.

Acknowledgments. This research is funded by the Brazilian Research Agencies CNPq (grants number 310634/2011-3, 311578/2011-0, and 485368/2013-7) and FAPES (grant number 59971509/12).

References

1. Anastasakis, K., Bordbar, B., Georg, G., Ray, I.: On challenges of model transformation from UML to Alloy. Softw. Syst. Model 9(1), 69–86 (2010)
2. Benevides, A.B., Guizzardi, G., Braga, B.F.B., Almeida, J.P.A.: Validating modal aspects of OntoUML conceptual models using automatically generated visual world structures. J. Univers. Comput. Sci. 16, 2904–2933 (2011)
3. Braga, B.F.B., Almeida, J.P.A., Guizzardi, G., Benevides, A.B.: Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal method. Innov. Syst. Softw. Eng. 6(1-2), 55–63 (2010)

[2] https://code.google.com/p/ontouml-lightweight-editor/

4. Brucker, A.D., Wolff, B.: OHOLCL: A formal proof environment for UML/OCL. In: Fiadeiro, J.L., Inverardi, P. (eds.) FASE 2008. LNCS, vol. 4961, pp. 97–100. Springer, Heidelberg (2008)
5. Cabot, J., Gogolla, M.: Object Constraint Language (OCL): A Definitive Guide. In: Bernardo, M., Cortellessa, V., Pierantonio, A. (eds.) SFM 2012. LNCS, vol. 7320, pp. 58–90. Springer, Heidelberg (2012)
6. Cunha, A., Garis, A., Riesco, D.: Translating between Alloy specifications and UML class diagrams annotated with OCL. Softw. Syst. Model (2013)
7. Eclipse EMF, `http://www.eclipse.org/modeling/emf/`
8. Eclipse MDT OCL, `http://www.eclipse.org/modeling/mdt/`
9. Gogolla, M., Bohling, J., Richters, M.: Validating UML and OCL models in USE by automatic snapshot generation. Softw. Syst. Model. 4(4), 386–398 (2005)
10. Guarino, N.: Toward Formal Evaluation of Ontology Quality. IEEE Intell. Syst. 19(4), 78–79 (2004)
11. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Telematica Instituut, The Netherlands (2005)
12. Jackson, D.: Software Abstractions-Logic, Language, and Analysis, Revised Edition. The MIT Press (2012)
13. Kuhlmann, M., Gogolla, M.: From UML and OCL to Relational Logic and Back. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 415–431. Springer, Heidelberg (2012)
14. Maoz, S., Ringert, J.O., Rumpe, B.: CD2Alloy: Class Diagrams Analysis Using Alloy Revisited. In: Whittle, J., Clark, T., Kühne, T. (eds.) MODELS 2011. LNCS, vol. 6981, pp. 592–607. Springer, Heidelberg (2011)
15. Massoni, T., Gheyi, R., Borba, P.: Formal Refactoring for UML Class Diagrams. In: 19th Brazilian Symposium on Software Engineering (SBES), pp. 152–167 (2005)
16. Mylopoulos, J.: Conceptual Modeling and Telos. In: Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development. Wiley, Chichester (1992)
17. NEMO OntoUML Infrastructure, `http://nemo.inf.ufes.br/en/ontoumlsupport`
18. OMG: Object Constraint Language, version 2.3.1 (2012)

Category Structure of Language Types Common to Conceptual Modeling Languages

Dirk van der Linden[1,2,3] and Hend*erik* A. Proper[1,2,3]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
{dirk.vanderlinden,erik.proper}@tudor.lu
[2] Radboud University Nijmegen, Nijmegen, The Netherlands
[3] EE-Team, Luxembourg, Luxembourg*

Abstract. We investigate the category structure of categories common to conceptual modeling languages (i.e., the types used by languages such as actor, process, goal, or restriction) to study whether they more closely approximate a discrete or graded category. We do this for three distinct groups: students, beginning modelers and experienced modelers. We find that overall most categories exhibit more of a graded structure, with experienced modelers displaying this even more strongly than the other groups. We discuss the consequences of these results for (conceptual) modeling in general, and in particular argue that when a model contains graded categories, it should follow that the (conceptual) validity of instantiations of it should be judged in a graded fashion as well.

Keywords: categorization, conceptual modeling, model semantics.

1 Introduction

We categorize the world around us in different ways depending on the subject matter. Some things we categorize more discretely, like natural things (e.g., fruits and plants), some things we categorize in a more graded way, such as artifactual things (e.g., tools, vehicles). These different categorization tendencies have been shown many times in research, starting around the time of Rosch et al. [22,23]. Also, they have been investigated by many others explicitly elaborating on the category structure for a number of natural and artifactual categories (cf. [8,4,9,10]). On the other hand, some work investigating this has had difficulties in finding significant differences in categorization tendencies between artifactual and natural categories (cf. [17]). There are also arguments that the natural/artifactual distinction is not granular enough, requiring us to also distinguish emotion categories [3]. Regardless of the debate whether particular kinds of categories are usually categorized in a particular way, it is clear that *we do not categorize everything in the same way.*

* The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University and HAN University of Applied Sciences (www.ee-team.eu).

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 317–331, 2014.

The categorization we speak of here deals with membership judgments. That is, whether a certain thing is judged to be a member of a given category. For example, most people would have no problem saying that an apple is a member of the category FRUIT[1], and they will likely reject the notion of a newspaper being so. However, when borderline cases are introduced interesting effects occur [13,14]. Given, for instance, cases that do not have clear or crisp boundaries, like tomatoes or rhubarb, people have more difficulty deciding with certainty whether they are FRUITS or not. In such cases people often tend to give *graded judgments* – things being members of a category to a certain degree. This prevalence of (strongly) graded membership judgments is then often correlated with the structure of the category being graded. Given that many of our modeling efforts (be they the creation of domain models, ontologies to formalize knowledge or support reasoning with, databases to implement schemata, etc.) require us to be as exact as possible about what we aim to model, it is clear that being aware of such differences in membership judgments is an important aspect of properly representing a given domain and the things in it.

The importance of being aware of these different judgments starts during the modeling phase, particularly in settings where there is collaborative modeling and integration efforts (e.g., enterprise modeling). The uncertainty of membership judgments (i.e., what is a valid instantiation for this type, is this instantiation as valid as others) creeps into models, and is often lost, unless explicitly elicited and written down. The effect this has on the validity of a model can occur on two levels, the level of the categories from the domain (i.e., the concepts from the universe of discourse) and the level of the categories from the language (i.e., the types used by a modeling language). Domain categories – the concepts from the universe of discourse – often receive great attention in discussions between modelers and stakeholders as well as in discussion between modelers themselves. This ensures (to some degree) that modelers know what things the stakeholders want to see in a model, and that they understand those things in the same way [16]. However, categories from the language receive such detailed attention far less often, e.g., by asking "*What exactly is this type 'actor' from the language we are using? Does it allow us to model the acting elements from the universe of discourse we know about?*". Instead, we often end up using the semantics of our own natural language [25] – together with all the category structures and nuances that come with it. Because of this, the language that ends up actually being used often differs from the (formal definition of the) modeling language that is used on paper [15]. For example, a modeling language might formally define an actor as a rather specific thing (e.g., requiring it to be a singular abstract entity, and whatever other features might apply), which makes it fairly easy to determine whether something is a valid instantiation of that type – a human being here definitely not being one. On the other hand, one of the modelers (or any reader of the model) might not use (or indeed, be aware of) those semantics, and instead see the type as having a different range of conceptually valid instantiations. This is problematic because it means that important semantics of the

[1] To distinguish categories from words we print them in SMALL CAPS.

model might be lost when it is interpreted by other people not involved in the original modeling process (e.g., during model integration), or stakeholders who were not aware of some of the not explicated particularities. This is exacerbated by the fact that we do not have an insight into the structure of these categories *as used* by people, because not only do we not know what is considered valid, we do not know whether some things are considered more valid than others.

Thus, in this paper we aim to clarify whether the categories common to many modeling languages and methods (i.e., those types used by a language to instantiate domain concepts by) are categorized in a discrete or graded fashion. The implications of this for model creation and usage (particularly for models used to capture and document a certain domain) are important to be aware of. If a category from a language is typically judged in a discrete fashion, the semantics of models are likely easier to communicate, formalize, and keep coherent. However, if such a category is typically judged in a graded fashion, communicating it to others becomes more involved, requiring more explicit discussion, and the formalizations and tools we use need to explicitly support this structure (e.g., by using ontologies with support for features as typicality and centrality).

To the best of our knowledge there has been little empirical research on category structure in the domain of conceptual modeling. In general the field of conceptual modeling lacks empirical research that tests (cf. [7,20,19]), while in this particular case work on formalizations and tools to support graded structures has already been done (e.g, [27,6]). The focus of this work is thus to present an exploratory *empirical* investigation into the structure of categories from modeling languages to determine whether the potential issues we described realistically come into play (i.e., *there are categories from modeling languages that are of a graded nature*). Based on our findings we will discuss how an understanding of these categories can be used to guide the process of model creation and use, for instance by helping modelers and stakeholders in capturing as much useful information about the allowed range of instantiation for a model, enabling others to read and use the model as it was intended by the creators.

The primary findings that we will show in this paper are that most of the categories from modeling languages tend to exhibit a graded structure, that many of the terms used for them are considered partial members, while a surprising amount of terms are also considered clear non-members. The possible complications that might arise because of these and other findings, and what kinds of models they affect are discussed in more detail in the rest of the paper.

The remainder of this paper is structured as follows. We detail our experimental setup in section 2, present the results in section 3, and discuss the consequences they may have for modeling and modeling languages in section 4. Finally, in section 5 we conclude and propose directions for future research.

2 Experimental Setup

What we wish to achieve is examine whether a number of categories more closely resemble graded or discrete categories. We can do this by performing a category membership experiment for the target categories and a number of benchmark categories of which we know whether they are typically judged in a discrete or graded fashion, and to what extent their members are judged so.

2.1 Considerations

There are a number of considerations to take into account with this investigation. First is the issue of the potential participants and their (natural) language. Most importantly, when we ask whether a certain thing is a member of a category or not, we would optimally do that in the participant's native language. However, as the terms used by most modeling languages and methods (i.e., the terms we will use in our experiment) are in English, we need to either use them as-is, or translate them. Given that most modelers use the terms as given by languages (i.e., in English), albeit often appending their own semantics, we will perform the experiment with the terms without localizing them.

For the benchmark we will use datasets from previous research. However, an issue with the existing and still often used datasets is that they can be outdated (e.g., the commonly used Barr & Caplan dataset was published in 1987), and they can be sensitive to cultural differences. Category judgments can shift as certain objects fall out of common use and are replaced by entirely different things, as well as certain objects can be seen differently in different cultures. For example, while in Barr & Caplan's dataset bicycles are found to not be strong members of the category VEHICLE, repeating the experiment with Dutch, Danish or German participants (who are far more likely to use a bicycle as a mode of transport [21]) will likely lead to significantly different results. As such, care will have to be taken when interpreting the results from the benchmark categories to place them into the correct frame of time and culture. While there are other datasets available that were gathered from non-English native speakers (e.g., Ruts et al. [24] who performed an exemplar generation study amongst Belgian students) that might be used to create a more even dataset, they often only include full members and lack the necessary borderline and non-members.

Finally, there is the question of the granularity of the categories from the modeling languages that we will investigate. On the highest level there is the distinction between entities and relationships (and sometimes values), which are the main categories used by certain non-domain-specific languages (e.g., ER, ORM). However, it would be more interesting to look into the more specific categories (e.g., PROCESS, RESOURCE, ACTOR) used by domain-specific languages (e.g., BPMN, e3Value, ArchiMate) as they are more likely to yield discriminating results. This will also make it possible to eventually distinguish between groups with different focuses (e.g., the BPM community, the ArchiMate community) and find out if there are significant differences between them in terms of categorization. Thus, for this investigation we will focus on categories found in domain-specific languages.

2.2 Method

Participants: Fifty-six participants participated in the present study. Twenty-one of them were advanced (3rd or 4th year) students at an undergraduate university of applied science with a focus on computing science and modeling, thirty-five were professional modelers employed at a research institute with a focus on IT and used modeling languages and tools to varying degrees. All participated voluntarily and received no compensation for their participation.

Materials: The materials used for the benchmark in the experiment were based on the list of exemplars reported on by Barr & Caplan [4]. We used 5 full, 5 partial and 5 non-members terms for both of the benchmarks. They were translated and presented in Dutch for the twenty students, but presented in English for the participants at the public research center, given that this was the only shared language between all participants and all participants were sufficiently fluent. In this text we consistently refer to them in English. For this benchmark we included the categories FRUIT and VEHICLES (see Table 4 in the appendix). For the modeling part of the experiment we investigated the categories ACTOR, EVENT, GOAL, PROCESS, RESOURCE, RESTRICTION and RESULT. These categories and related terms result from an earlier performed analysis on modeling languages and methods commonly used in enterprise modeling, which was reported on in [18]. The terms used for the members of these categories are the terms as used by the modeling languages and methods, based on the official (or most-used) specification (see Table 2 in Ref. [18] for the entire list, not replicated here due to space considerations).

Procedure: The procedure was based on Estes' [9] setup. Participants were divided into three groups (students, beginning modelers and expert modelers) and completed the task through on online survey. In this survey, participants were instructed to judge whether a list of given terms were either full, partial or non-members for the current category. Participants were informed beforehand that partial member scores meant that the exemplar belonged to the category, but to a less degree than others. This was first done for the two benchmark categories, and followed in the same way for each of the investigated categories from the modeling languages. The orders of the terms in each category were randomized for each participant. Care was taken to validate that participants filled out the survey seriously by comparing results and checking for long strings of repeating answers that the randomization should have prevented from occurring.

3 Results

The proportion of graded membership judgments for the terms used in the benchmark which are partial members are shown in detail in Table 1. The terms listed here are solely the partial members (as determined by the original datasets). What was to be expected is that the typically discrete category (FRUIT) would show lower proportions of graded judgments compared to the typically graded category (VEHICLES). The given scores indicate the proportion of partial member judgments (e.g., 19% of students, 13% of beginning modelers, and 30% of expert modelers considered an avocado as a partial member of the FRUIT

Table 1. Partial member proportions for the partial member terms of the benchmark

Category	Term	Student	Beginner	Expert	Ref. [4]	Ref. [9]
FRUIT	avocado	0.19	0.13	0.30	0.37	0.16
	coconut	0.24	–	0.05	0.38	0.37
	tomato	0.33	0.27	0.25	0.34	0.05
	cucumber	0.19	–	0.25	0.23	0.21
	rhubarb	0.14	0.20	0.15	0.45	0.26
VEHICLES	gondola	0.24	0.20	0.20	0.50	0.21
	tricycle	0.14	0.13	0.10	0.64	0.58
	wheelchair	0.29	0.27	0.50	0.70	0.63
	horse	0.48	0.27	0.55	0.54	0.50
	husky	0.38	0.27	0.55	0.27	0.21

category). Shown are respectively the scores for students, beginning modelers, expert modelers, and the scores as reported by Barr & Caplan [4], and Estes [9].

A more detailed overview of the average amount of full, partial and non-member judgments for each investigated category is given in Table 2. The results are given for each investigated group (students, beginning modelers and expert modelers), and indicate the proportion of membership judgments. For example, students considered 47% of the presented terms for the ACTOR category to be full members, 18% to be partial members and 35% to be non-members. The primary points of interest here are the higher scoring partial and non-member results, as they indicate words actually used by modeling languages that are either only considered to be partially reflective of their category (e.g., a 'market segment' would be only considered somewhat an ACTOR), or are considered not to be exemplars of that category (e.g., a 'requirement unit' would not be considered an ACTOR).

Table 2. Average amount of membership scores (full, partial and non-members) for each group of investigated categories

	student ($n = 20$)			beginner ($n = 15$)			expert ($n = 21$)		
Category	full	partial	non	full	partial	non	full	partial	non
ACTOR	0.47	0.18	0.35	0.30	0.14	0.55	0.41	0.25	0.35
EVENT	0.46	0.14	0.41	0.39	0.16	0.45	0.29	0.19	0.51
GOAL	0.65	0.11	0.23	0.60	0.16	0.24	0.56	0.20	0.24
PROCESS	0.66	0.14	0.20	0.62	0.22	0.16	0.41	0.32	0.28
RESOURCE	0.59	0.19	0.22	0.62	0.19	0.20	0.54	0.22	0.24
RESTRICTION	0.50	0.21	0.29	0.55	0.18	0.27	0.39	0.24	0.37
RESULT	0.73	0.16	0.11	0.86	0.07	0.08	0.76	0.16	0.09
FRUIT	0.44	0.10	0.45	0.47	0.05	0.42	0.49	0.09	0.41
VEHICLE	0.48	0.14	0.37	0.49	0.13	0.37	0.51	0.20	0.29

Table 3 gives a detailed overview of specific modeling language terms considered partial members by at least ≥ 30% of one of the investigated groups. A clear difference can be seen between the groups for most categories, with the expert modelers displaying on average a much higher amount of graded judgments than the students or beginning modelers. On average students considered 15% of the investigated terms to be partial members, while beginning modelers did so for 32% and expert modelers considered 83% to be partial members.

Table 3. Terms considered partial members by ≥ 30% of at least one group. The terms listed here are *only* those considered partial members, thus not including the terms considered full or non-members. The amount of terms listed here is respectively 43%, 32%, 26%, 48%, 50%, and 25% of the total amount of terms investigated for each respective category.

Category	Term	Student	Beginner	Expert
ACTOR	unit			✓
	requirement unit			✓
	infrastructural component	✓		✓
	organizational component			✓
	device			✓
	application software			✓
	organizational unit			✓
	hardware			✓
	software	✓		✓
EVENT	behavior			✓
	function			✓
	interaction			✓
	activity		✓	
	task		✓	✓
	service task			✓
	value activity	✓		✓
	contribution			✓
	operation			✓
GOAL	expectation	✓	✓	✓
	requirement			✓
	consumer needs			✓
	target			✓
PROCESS	organizational service			✓
	infrastructure service			✓
	information service			✓
	other service		✓	✓
	IT service		✓	✓
	service			✓
	sub flow		✓	✓
	process flow			✓
	dependency path		✓	

Table 3. (*Continued*)

	game		✓	✓
	task			✓
RESOURCE	artifact		✓	✓
	hd			✓
	location		✓	
	data object		✓	✓
	business object			✓
	object	✓	✓	
	data input			✓
	input			✓
	value object	✓		✓
	network device		✓	
	representation		✓	
	value port		✓	
	device	✓		
RESTRICTION	belief		✓	
	priority		✓	
	value		✓	
	interface	✓		
	catching			✓
	throwing	✓		✓
	license			✓
	trust			✓
	interrupting			✓
	non-interrupting			✓
	strategy			✓
	strategic objective	✓	✓	✓
RESULT	end event		✓	✓
	payoff			✓

4 Discussion

We will first discuss the results in general, showing how they support the assumption that there are categories in modeling languages that are of a graded nature. We will then discuss in more detail to what kind of models and modeling languages our results are most applicable and consequences our findings entail for them. Finally, we also discuss a number of limitations of our current study that should be kept in mind when interpreting the results.

4.1 General Discussion

It was expected that the partial member judgments for the natural and artifactual benchmark categories would show a difference, with the artifactual category displaying a higher proportion of graded judgments. Although compared to the results from Barr & Caplan [4] and Estes [9] the overall amount of graded judgments seems to be lower, the relative distribution still seems intact. This is the case for both the beginning and expert modelers (the proportion of some graded judgments for VEHICLES being at least twice as large compared to the ones for FRUITS). This is not the case for the student group, as the difference between the benchmark categories there was found to be much smaller. This could be explained by the lower amount of experience with (and exposure to) modeling (and modeling languages) students have. This is further reflected in Table 3 where there are far less words considered partial members by students than by the more experienced modelers.

On average the proportion of partial member judgments is 0.16 for students, 0.16 for beginning modelers, and 0.23 for expert modelers. When we compare these scores to the average proportion of partial member judgments for the discrete and graded benchmark categories in Table 2 (respectively 0.10 and 0.14 for the students, 0.05 and 0.13 for the beginning modelers and 0.09 and 0.20 for the expert modelers), we can see that for the two groups of modelers most scores shown for the categories from modeling languages more clearly reflect the graded benchmark category than the discrete one. Thus, as a careful first investigation we seem to have found support that most categories from modeling languages are of a graded nature. Given that the distribution of terms for these categories was not the same as the benchmark categories (i.e., the benchmark categories were made up of equal amounts of full, partial and non-members, while for the categories from the modeling languages we were unaware of this distribution, with them likely containing proportionally more full members) this makes it all the more acceptable to support the idea described in the introduction that *these categories can be seen as exhibiting a graded structure.*

Another interesting finding is the high amount of non-member judgments found in many of the categories. It is striking that the terms we have used in modeling languages and methods are sometimes considered absolute non-members of their related category. In particular, it can be seen that EVENTS are the largest category for non-members across all groups (respectively 0.41, 0.45, and 0.51), while ACTORS and RESTRICTIONS also have a high amount of non-members in some groups. A possible explanation for this is that people are quicker to judge about things they are specialized in, for example a process modeler having more snap judgments about concepts to do with processes, and thus also being more willing to rule out terms. In practice this means that the terminology we use originating from some languages might not reflect our innate category judgments at all, raising the question whether this is a bad thing (e.g., because the terminology is far away from our naive understanding and semantics) or perhaps not that much of a problem (e.g., because the mismatch between a term and our understanding of it in a given context makes it easier to 'redefine' and use it in that context).

As already hinted at and most clearly visible in Table 3, there is a striking difference between the groups we investigated when it comes to the proportion of partial member judgments. The expert modelers have a far higher amount of partial member judgments compared to the students, and in a lesser degree to the beginning modelers. An exception to this are PROCESSES and RESOURCES, which are judged more comparably between beginning and advanced modelers. This might be explained by the fact that the department of the research institute which the majority of the participants were working in has a strong focus on service science and is thus focused on many efforts involving processes (e.g., process modeling). An explanation for the difference between these groups might be that students simply have had less exposure to modeling terminology and are thus more likely to give absolute judgments. On the other hand, there is also the possibility that the (expert) modelers are, through training and experience, cognitively better equipped to deal with situations with abstract and vague concepts (cf. [26]), which could manifest in a higher amount of graded judgments.

4.2 Applicability of Our Findings

Before we move on to discuss the consequences of these findings for model creation and use, we need to specify more clearly to what kinds of models and languages they are applicable. Models created with more general modeling languages like UML, ER, and ORM are less affected by the existence of graded categories, as the main types (i.e., entities and relationships) they use are already so abstract that one would not so much expect *subtle* misunderstandings that stay unnoticed to arise in the same way as they would in domain specific languages. Furthermore, when languages like these need to be made more specific, they can do so by, e.g., explicitly capturing the necessary facts in ORM, or using UML stereotyping to create the needed new semantics. The semantics given by the modelers can then become an explicit part of the language.

However, when it comes to domain-specific languages our findings become much more relevant. This is because the semantics of the types used by (and often pre-defined in) these languages are less abstract than the ones mentioned above, and the risk of subtle misunderstandings that are not immediately noticed is higher. With the plethora of domain-specific languages (e.g., ArchiMate, BPMN, e3Value, i*, ITML, ADeL) in active use today all with their own focus (e.g., enterprise architecture, processes, value exchanges, goals, IS implementations, IS deployments) our findings could have consequences for many modeling efforts. The consequences we discuss should thus be taken to be most relevant for domain-specific modeling languages like these and any artifacts based on the models created with them.

4.3 Consequences for Modeling

When it comes to the modeling languages and models that are affected by our findings, we see a number of different kinds of models:

1. *models used to communicate* between, and with different modelers and stakeholders (e.g., conceptual models)

2. *models used to formalize* information from a given domain, for whatever purpose (e.g., ontologies, models as documentation)
3. *models used to execute* by non-human systems (e.g., compiled source code)

This list is not intended as a taxonomy of models, nor as an exhaustive list of the different kinds of models that are affected by graded category structures. It is merely a starting point to reason about the different consequences we see our work having for different kinds of models. We furthermore do not mean to imply that kinds of models are mutually exclusive (e.g., that models used to communicate are never used to formalize or transformed into executable models).

Models used to communicate involve conceptual models of many possible purposes (e.g., capturing a domain, models used to guide decision making). As we have shown that the categories used by modeling languages are likely of a graded nature, the models created by them necessarily also contain categories of a graded nature. The most important consequence here is that an instantiation of a model is not just simply valid or invalid, but will display degrees of validity as well. If the category goal is seen as a graded structure, with some things being better goals than others, it is thus possible to instantiate a model that contains some goal type with two different cases that are both valid, but not equally so. As the formal semantics of most modeling languages do not explicitly support such degrees of validity, it is important that we are clear about the limits of conceptual validity of our models. In other words, to ensure people read and use models in a similar way, we need to ensure that we provide clear examples of possible valid instantiations, and perhaps more importantly, clear examples of that which we consider invalid as well.

For example, while 'hardware' and 'software' are both considered partial members (by at least the experienced modelers in our study) of the category ACTOR, the exact degree to which they are both considered so is something that is likely different for different (groups of) people. If we are creating a model used for the implementation of an information system, which would likely incorporate such terms for the things that act to support and execute business activities, we need to be clear to what degree they can both be seen as ACTORS. For instance, the modelers or stakeholders might envision the hardware as the actual acting part, with the software providing the instructions for doing it so, and thus find a model where 'hardware' is said to act out a business function more valid than where 'software' does so. However, others might disagree and see 'software' as the actual thing that acts. As these interpretations can be different from group to group, it is thus important to involve explicit discussions about the degrees of validity for different things we use in our models during model creation.

Models used to formalize are for instance models that capture knowledge about a certain domain and attempt to formalize it in order to reduce the amount of ambiguity. A formalization involving graded categories needs to ensure that membership requirements are not discrete, and more important, take into account the relevant properties of a graded category (e.g., centrality and typicality of members). There is work in the field of ontology engineering that strives towards explicitly supporting these structures, e.g., [2] and explicit modification of ontology formalizations to incorporate the noted features [27,28], and critiques and extensions of proposed work, e.g., [6]. If such formalizations are not used, and instead a classical approach based on discrete judgments is used, much semantic information about the domain and the judgments from the original modelers is lost. This can lead to misinterpretations by other readers and users of the model if there is no communication between them and the original modelers anymore. For instance, someone might consider a horse as a VEHICLE (albeit an atypical one) and thus consider it to be somewhat of a valid vehicle in their created ontology. However, when this is formalized discretely, any other member of the VEHICLE category (e.g., a car) would be considered on equal footing with the horse, while this has no grounding in the real world whatsoever. As such, the formalization can no longer be considered a correct representation of the real world and loses a lot of its value.

Models used to execute are for instance source code which is run by an interpreter, or compiled and then executed. Other options are models interpreted by model provers, expert systems, or ontologies used for automated reasoning and so on. For example, a model used by an expert system to check for a number of possible cases (e.g., a medical advice system) might need graded structures and judgments in order to correctly reason with the real-world information. A number of formalizations for e.g., descriptive logics have been proposed to incorporate graded features like typicality and centrality [5,12,11]. These models are affected in a similar way to the ones used to formalize, meaning that their formalizations need to support any graded structures found in them. This is all the more important to ensure here, as executable models are often no longer read and interpreted by people, and thus any errors or oversights in them are less likely to be corrected.

4.4 Limitations

While it is good to find that our results hint towards the modeling categories having a graded nature, care must be taken not to immediately extrapolate this finding and use it to judge the structure of the investigated modeling categories in general. For one, this has been only one study, with two of our groups of participants being people with *professional* experience in conceptual modeling. For these reasons repeating the study presented here with additional groups of (experienced) people to validate whether they share the same graded structure would be a prudent thing to do.

Furthermore, as categorization judgments are something inherent to people, it would also be useful to perform this study on specific subgroups of modelers (e.g., process modelers, enterprise architects, goal modelers) to analyze whether the proportion of graded responses is different for specific categories or not (i.e., test whether categories that modelers are focused on receive less partial member judgments). One could for instance hypothesize that people who are specialized in a topic have less semantic flexibility in regards to the categories of that topic.

Related to the terms we used, it might also be interesting to see whether the introduction of model context (i.e., presenting the terms while being used in a model) instead of the isolated terms themselves would yield different results. Nonetheless, the results from our study investigating the terms in isolation also provides useful insight into the amount of terms that would typically not be considered as good representatives of their functional category. Furthermore, this might provide an additional source of complexity and confusion for participants, as with the amount of terms we used in the study, a large amount of different modeling languages would be used, some of which participants are likely not familiar with.

It should also be noted that the study presented here talks about the structure of the category in terms of it being graded or discrete, but does not aim to give a representation of the *internal* structure. Further studies involving explicitly eliciting typicality and centrality of the terms investigated here could be done in an attempt to discover such structures. It is very likely that the internal structure of the categories (which is regardless of the graded or discrete question) is specific to different groups of people, as it can be readily expected that process modelers will have a different central core for a number of categories than, for example, goal modelers. Thus, such studies should also be performed with a number of different groups of modelers.

Finally, as referred to earlier, the distribution of the terms for the modeling categories was not optimal (i.e., not evenly divided between full, partial and non-member), which makes it more difficult to infer detailed general statements about the structure. Such work on the detailed structure of these categories like described above can be undertaken in further research, where the individual category members are rated on typicality and centrality in order to attempt to build an actual representation of a shared category structure. Such findings could then be used to create a more evenly distributed set of modeling terms for further membership judgment experiments.

5 Conclusion

We have presented a study into the category structure of types used by most modeling languages. This study showed that many of these modeling categories are likely of a graded nature (that is, some things are considered to be better members than others), which can have an effect on the semantics of models and their derivatives. We have discussed the implications for validity of models and proposed that more study into the understandings specific groups have of such categories would be a worthwhile avenue of research. The main contribution of this work has been empirically showing that *the categories we use to model are likely of a graded nature*, which before was only assumed (or worse, ignored). More specifically, we have shown that the modeling terminology from actual modeling languages and methods are affected by this graded nature as well. In future work we hope to extend this research to different groups with a strong focus on a specific domain to investigate potential categorization differences between different people operating in different domains.

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

References

1. Adelson, B.: Comparing natural and abstract categories: A case study from computer science. Cognitive Science 9(4), 417–430 (1985)
2. Furst, F., Kuntz, P., Trichet, F.: Conceptual and Lexical Prototypicality Gradients Dedicated to Ontology Personalisation. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1423–1439. Springer, Heidelberg (2008)
3. Altarriba, J., Bauer, L.M.: The distinctiveness of emotion concepts: A comparison between emotion, abstract, and concrete words. The American Journal of Psychology, 389–410 (2004)
4. Barr, R., Caplan, L.: Category representations and their implications for category structure. Memory & Cognition 15(5), 397–418 (1987)
5. Britz, K., Heidema, J., Meyer, T.: Modelling object typicality in description logics. In: Nicholson, A., Li, X. (eds.) AI 2009. LNCS, vol. 5866, pp. 506–516. Springer, Heidelberg (2009)
6. Cai, Y., Leung, H.-F.: A formal model of fuzzy ontology with property hierarchy and object membership. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 69–82. Springer, Heidelberg (2008)
7. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58(3), 358–380 (2006)
8. Diesendruck, G., Gelman, S.: Domain differences in absolute judgments of category membership: Evidence for an essentialist account of categorization. Psychonomic Bulletin & Review 6(2), 338–346 (1999)
9. Estes, Z.: Domain differences in the structure of artifactual and natural categories. Memory & Cognition 31(2), 199–214 (2003)
10. Estes, Z.: Confidence and gradedness in semantic categorization: Definitely somewhat artifactual, maybe absolutely natural. Psychonomic Bulletin & Review 11(6), 1041–1047 (2004)
11. Freund, M., Descles, J.P., Pascu, A., Cardot, J.: Typicality, contextual inferences and object determination logic. In: FLAIRS, vol. 4, pp. 491–495 (2004)
12. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Reasoning about typicality in preferential description logics. In: Hölldobler, S., Lutz, C., Wansing, H. (eds.) JELIA 2008. LNCS (LNAI), vol. 5293, pp. 192–205. Springer, Heidelberg (2008)
13. Hampton, J.A.: Similarity-based categorization and fuzziness of natural categories. Cognition 65(2-3), 137–165 (1998)
14. Hampton, J.A., Dubois, D., Yeh, W.: Effects of classification context on categorization in natural categories. Memory & Cognition 34(7), 1431–1443 (2006)
15. Henderson-Sellers, B.: UML - the Good, the Bad or the Ugly? Perspectives from a panel of experts. Software and System Modeling 4(1), 4–13 (2005)
16. Hoppenbrouwers, S.J.B.A.: Freezing language: conceptualisation processes across ict-supported organisations. Ph.D. thesis, Radboud University Nijmegen (2003)
17. Kalish, C.W.: Essentialism and graded membership in animal and artifact categories. Memory & Cognition 23(3), 335–353 (1995)

18. van der Linden, D.J.T., Hoppenbrouwers, S.J.B.A., Lartseva, A., Proper, H.A.(E.): Towards an investigation of the conceptual landscape of enterprise architecture. 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. 526–535. Springer, Heidelberg (2011)
19. Moody, D.: Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. Data & Knowledge Engineering 55(3), 243–276 (2005)
20. Persson, A., Stirna, J.: Why enterprise modelling? an explorative study into current practice. In: Dittrich, K.R., Geppert, A., Norrie, M. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 465–468. Springer, Heidelberg (2001)
21. Pucher, J., Buehler, R.: Making cycling irresistible: Lessons from the Netherlands, Denmark and Germany. Transport Reviews 28(4), 495–528 (2008)
22. Rosch, E.: Natural categories. Cognitive Psychology 4(3), 328–350 (1973)
23. Rosch, E., Mervis, C.B.: Family resemblances: Studies in the internal structure of categories. Cognitive Psychology 7(4), 573–605 (1975)
24. Ruts, W., De Deyne, S., Ameel, E., Vanpaemel, W., Verbeemen, T., Storms, G.: Dutch norm data for 13 semantic categories and 338 exemplars. Behavior Research Methods 36, 506–515 (2004), doi:10.3758/BF03195597
25. Sowa, J.: The Role of Logic and Ontology in Language and Reasoning. In: Theory and Applications of Ontology: Philosophical Perspectives, pp. 231–263. Springer (2010)
26. Wilmont, I., Barendsen, E., Hoppenbrouwers, S., Hengeveld, S.: Abstract reasoning in collaborative modeling. In: HICSS, pp. 170–179 (2012)
27. Yeung, C.-M.A., Leung, H.-F.: Ontology with likeliness and typicality of objects in concepts. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 98–111. Springer, Heidelberg (2006)
28. Yeung, C.A., Leung, H.F.: A formal model of ontology for handling fuzzy membership and typicality of instances. Comput. J. 53(3), 316–341 (2010)

Appendix

Table 4. The categories and terms for the benchmark as adapted from [4] and [9], followed by the used Dutch translations for the student group

Category	Term
FRUIT (discrete)	apple, pear, plum, banana, pineapple, avocado, coconut, tomato, cucumber, rhubarb, carrot, onion, potato, rose, spinach
VEHICLES (graded)	bus, car, truck, van, taxi, gondola, tricycle, wheelchair, horse, roller skates, husky (dog), lawnmower, bus driver, carton, newspaper
FRUIT (discrete)	appel, peer, pruim, banaan, ananas, avocado, kokosnoot, tomaat, komkommer, rabarber, wortel, ui, aardappel, roos, spinazie
VEHICLES (graded)	bus, auto, vrachtwagen, busje, taxi, gondel, driewieler, rolstoel, paard, rolschaatsen, husky (hond), grasmaaier, buschauffeur, doos, krant

Model Comprehension and Stakeholder Appropriateness of Security Risk-Oriented Modelling Languages

Raimundas Matulevičius

Institute of Computer Science, University of Tartu, Estonia
rma@ut.ee

Abstract. Modelling and management of the security risks from the early stages of information systems development could help to envision early security threats, their consequences and potential countermeasures. However, the security modelling languages could bring benefit only if they are correctly applied and the stakeholders comprehend models and agree about their meaning. In this paper we analyse how humans comprehend the security risk-oriented/aware modelling (SRM) languages and models. Specifically, by applying the semiotic quality framework, we investigate (*i*) concepts of the security risk management, and (*ii*) participant and modeller appropriateness regarding the SRM languages. Our results indicate the best and worst perceived SRM constructs and highlight few challenges to improve the SRM languages.

Keywords: Security risk management, security modelling languages, security requirements, and information systems.

1 Introduction

Security engineering is "concerned with lowering the risk of intentional unauthorized harm to valuable assets to level that is acceptable to the system's stakeholders by preventing and reacting to malicious harm, misuse, threats, and security risks" [4]. It plays a vital role in the modern system development. However, the literature reports [5] [16] that security concerns often arise only during the implementation or maintenance of the actual system. This is a serious limitation to the secure system development, since the early security consideration (e.g., at the business process and/or requirements) could help developers to elicit security threats, their consequences and design countermeasures and, then, the design alternatives, that do not offer a sufficient security level, could be discarded without suffering high cost.

Modelling languages (e.g., Secure *i** [6], Secure Tropos [12], misuse cases [16], mal-activities [15], and UMLsec [5]) provide powerful means to understand the security concerns during the early system development stages. However, on one hand, security analysts are not aware of the benefits of these languages. They are not taught of the language syntax and semantics used to describe and analyse security-related problems. On the other hand, the modelling languages themselves should be engineered in a way, which would stimulate and support the security modelling. In other words, modelling languages need to be of the *right* quality to express security concerns from the early stages of the system development.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 332–347, 2014.

In this paper we analyse the quality of the security risk-oriented/aware modelling (SRM) languages. Our scope includes security risk-oriented BPMN [1], security risk-aware Secure Tropos [9] and security risk-oriented misuse cases [17]. All these languages have been extended to express the concepts of the domain model for the information security risk management (ISSRM) [3] [10]. More specifically, we investigate how the *human stakeholders* understand the models created using the SRM languages, and whether such models carry the security related information, which is correctly understood by the model readers. According to [7], language perception concerns language user's ability to understand the concepts of the modelling language. This challenged us to consider how humans understand the SRM constructs when reading and creating the model for the security risk management. Our research questions are:

RQ.1: How do the human participants *comprehend* the models for the security risk management?
RQ.2: How do the *participants perceive* (the *constructs* of) the SRM languages?
RQ.3: How do the *modellers* use (the *constructs* of) the SRM languages?

To answer these research questions, we have performed an empirical study at the University of Tartu (Estonia). We have collected data from the (potential) security analysts who reviewed the SRM models. The third question was answered by analysing the SRM models created by the respondents.

The paper is structured as follows. In Section 2 we overview the semiotic quality framework and the ISSRM domain model, both used to evaluate the SRM languages. Section 3 introduces the SRM languages and illustrates their models. Section 4 presents the details of the empirical study and gives the major results. Finally, in Section 5 we discuss the findings and conclude the study.

2 Theory

2.1 Domain Model for IS Security Risk Management

Definition of the SRM languages is performed following the domain model [3] [10] for information systems security risk management. This domain model, displayed in Fig. 1 contains the process guidelines that help identify the vulnerable assets, determine their security objectives, assess the risks, and elicit security requirements to mitigate these risks.

Asset-related concepts (i.e., *business and IS assets, IS assets, and security criterion*) explain the organisation's values that need to be protected. The needed protection level is defined as the security needs, typically in terms of confidentiality, availability and integrity. *Risk-related concepts* (i.e., *risk*, *impact*, *event*, *vulnerability*, *threat*, *attack method*, and *threat agent*) define the risk itself and its components. Risk is a combination of threat with one or more vulnerabilities, which leads to a negative impact, harming some assets. An impact shows the negative consequence of a risk on an asset if the threat is accomplished. A vulnerability is a weakness or flaw of one or more IS assets. An attack method is a standard means by which a threat agent executes

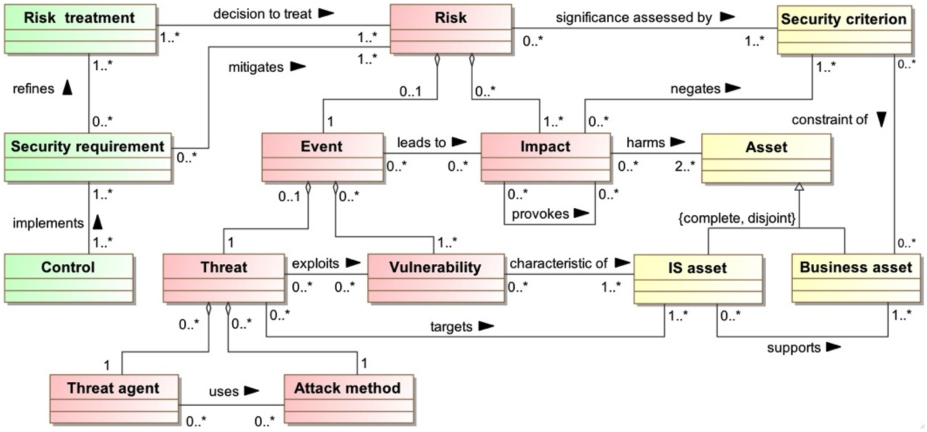


Fig. 1. The ISSRM Domain Model (adapted from [3] [10])

a threat. *Risk treatment-related concepts* (i.e., *risk treatment* decision, *security requirement* and *control*) describe how to treat the identified risks. A risk treatment leads to security requirements mitigating the risk, implemented as security controls.

In this study the ISSRM domain model is applied for three purposes: (*i*) for presenting the major concepts and constructs of the SRM languages both at the *semantic* and *syntactic* levels; (*ii*) for understand the *semiotic clarity* of the SRM languages; and (*iii*) for analyses of how humans SRM languages and their models.

2.2 Semiotic Quality Framework

In this study we apply the semiotic quality framework (SEQUAL) [7]. The SEQUAL framework (*i*) distinguishes between quality characteristics (goals) and means potentially to achieve these goals; (*ii*) is based on the constructivistic that recognises model creation as part of a dialog between participants whose knowledge changes as the process takes place; and (*iii*) is closely linked to linguistic and semiotic concepts, in particular to the link between syntax, semantics and pragmatics. The framework could be applied to assess the quality of the conceptual models (e.g., in terms of different *quality types*, like physical, semantic, syntactic, pragmatic, and others) and the quality of the modelling languages (e.g., in terms of *appropriateness* to domain, comprehensibility, participant, modeller, and others).

In this paper we consider how humans comprehend the models created using the SRM languages. Our second goal is to understand what the participant appropriateness (i.e., the link between the participant knowledge and the SRM language) is. Finally, we analyse the correspondence between the SRM language and the knowledge of the ones who perform the modelling, i.e., the modeller appropriateness.

3 Security Risk-Oriented/Aware Modelling Languages

In this study we have analysed three SRM languages. These are extensions of BPMN [14], Secure Tropos [12], and misuse cases [16] towards security risk management.

3.1 Security Risk-oriented BPMN

The primary purpose of Business Process Model and Notation (BPMN, version 2.0) is the business process modelling [14]. In Fig. 2 we present the security risk management model expressed in BPMN following the extensions introduced in [1] (see Table 1). The *business assets* are expressed using the BPMN *tasks* and *data objects* (e.g., Account data). Here the *confidentiality* of the Account data is considered, thus the *data object* is equipped with the *lock* icon. These business assets are supported by the *IS assets* expressed using *Pool* (e.g., Software System) and *tasks* (e.g., Accept registration, and Insert data to database).

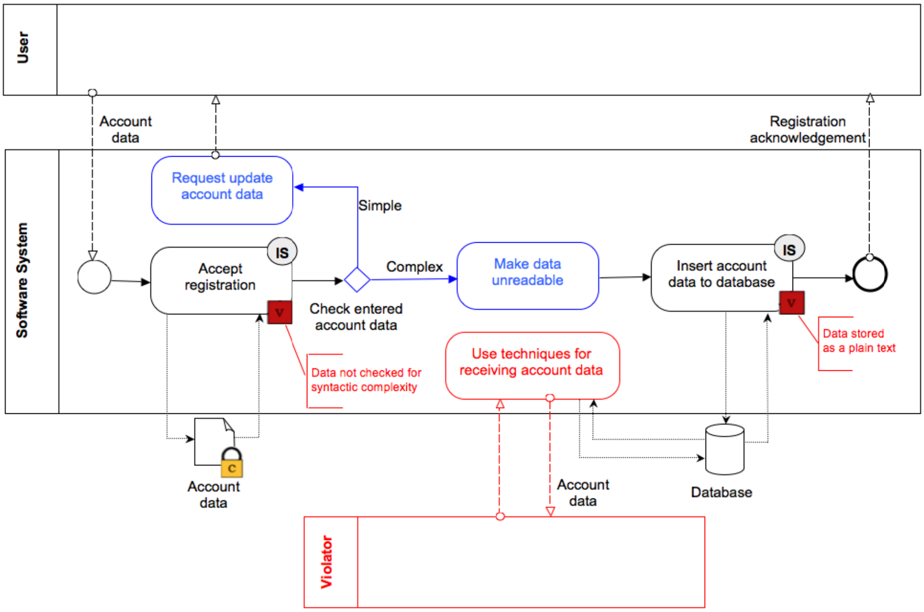


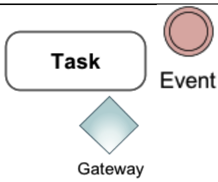
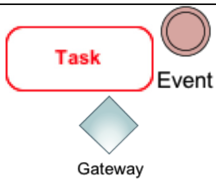
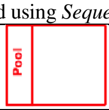
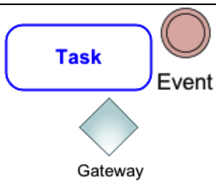
Fig. 2. Security risk management using BPMN

The security *threat* is defined using BPMN constructs for the *threat agent* (i.e., *pool* Violator) and *attack method* (i.e., *task* Use techniques for receiving account data). The risk *event* becomes possible because the there exist few *vulnerabilities*, such that the data are not checked for syntactic completeness and data are stored as a plain text. If risk happens, the Account data would lose their confidentiality (see *unlock* icon on Account data). Mitigation of the identified risk in introduced using *tasks* (e.g., Make data unreadable and Request update account data) and *gateways* (e.g., Check entered account data).

3.2 Security Risk-aware Secure Tropos

Secure Tropos [12] is an extension of the Tropos methodology [2], which supports development of IS through four stages – early and late requirements engineering, architectural and detailed design. Secure Tropos suggests an incremental way to

Table 1. BPMN construct alignment and extension to the ISSRM concepts

ISSRM domain model		**BPMN constructs**	**Examples and notes**
0		**1**	**2**
Asset-related concepts	Assets	 combined using *Sequence flows*	<u>Note</u>: constructs used to address both *business* and *IS* assets. However, in the current example only IS assets are expressed as a process consisting of Accept registration and Insert account data to database tasks.
	Business assets		Data object Account data
	IS assets		Pool Software System; Data store Database
	Security criterion	 added to *Task* or *Data object*	<u>Note</u>: Security objective is expressed using the *Lock* visual construct. Security criterion is defined on the construct on which the lock is added. <u>Example</u>: Confidentiality of account data
Risk-related concepts	Impact		Confidentiality of account data is negated; Data is harmed
	Vulnera-bility	 added to the *IS asset* constructs, such as *Task* or *Data store*	<u>Note</u>: actual vulnerability is added as the annotation, linked to the vulnerability point. <u>Example</u>: Data not checked for syntactic complexity, and Data stored as a plain text.
	Attack method	 combined using *Sequence flows*	Task Use techniques for receiving account data
	Threat agent		Pool Violator
Risk treatment-related concepts	Risk treatment	—	Risk reduction
	Security requiremen t	 combined using *Sequence flows*	Tasks Make data unreadable and Request update account data
	Control	—	—

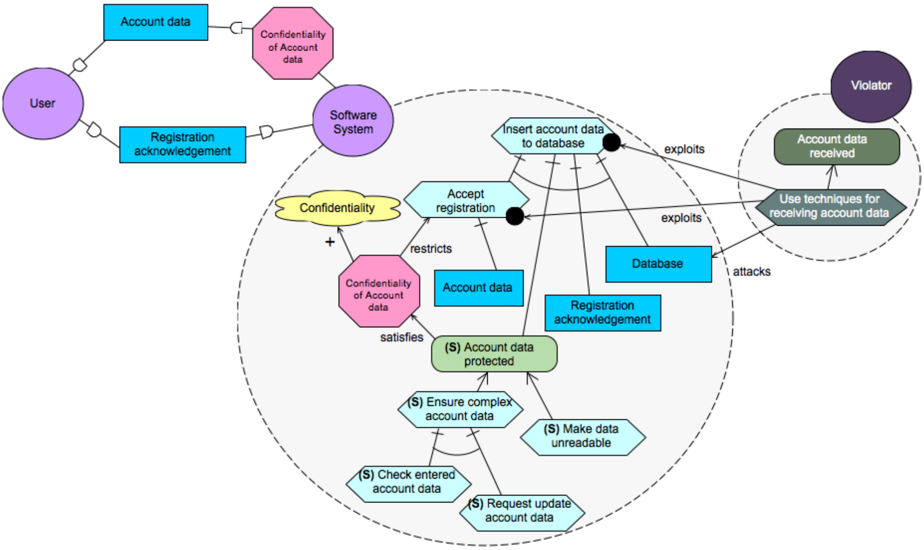


Fig. 3. Security risk management using Secure Tropos

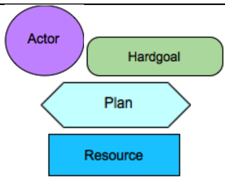
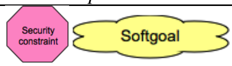
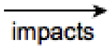
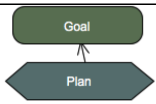
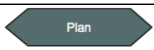
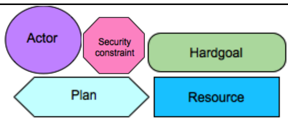
analyse security by introducing a construct, i.e., *security constraint*, to represent security restrictions that the system must have and actors must respect. The Secure Tropos extensions regarding security risk management are presented in [9] (Table 2).

Typically development of the Secure Tropos model starts with the social actor dependency analysis. In Fig. 3 two dependencies are defined between actors User and the Software System. A *security criterion* (i.e., Confidentiality of Account data) restricts registration acceptance (i.e., *task* Accept registration) and contributes to the general *Confidentiality* property of the systems. The *business assets* (i.e., Account data) are supported by the *IS assets* defined using the *actor* (i.e., Software System), *plan* (i.e., Accept registration and Insert account data to database), and *resource* (i.e., Database) constructs. To reach his *goal* Account data received, a *threat agent* (i.e., Violator) performs the *attack method* consisting of one *plans* – Use techniques for receiving account data. In this way the attacker *exploits* the *vulnerability* point (a characteristic of plans Accept registration and Insert account data to database) and *targets* (i.e., *attacks*) the Database. To mitigate this security *event*, a security goal Account data protected is introduced. This goal could be achieved in two alternative ways: either by ensuring complex account data or by making data unreadable.

3.3 Security Risk-Oriented Misuse Cases

Misuse cases diagrams [16] are an extension of the UML use cases towards security engineering. Sindre and Opdahl define a *misuse case* as a list or sequence of steps, if performed by a *misuser* (i.e., and actor who is using the system with unfavourable intents) successfully, cause harm to the stakeholders or the system itself. In [17] the misuse case diagrams are analysed regarding security risk management (Table 3). In Fig. 4 the *actor* User communicates to the Software System to Submit account data. Hence the *security criterion* Confidentiality of account data is introduced.

Table 2. Secure Tropos construct alignment and extension to the ISSRM concepts

ISSRM domain model		**Secure Tropos**	**Examples and notes**
0		**1**	**2**
Asset-related concepts	Business assets	 combined using *dependency*, *contribution*, *means-ends*, and *decomposition* links	Resource Account data
	IS assets		Actor Software system, tasks Insert data to database and Accept registration, resource Database
	Security criterion	 combined using *contribution* and *security constraint decomposition* links	Confidentiality Confidentiality of account data
Risk-related concepts	Impact		Impacts confidentiality; Note: this relationship is not illustrated.
	Event	 *or* a combination of constructs to express *Vulnerability*, and *Threat*	Stealing account data (this construct is not illustrated); Violator who has the goal Account data received, performs plan Use techniques for receiving account data because of the IS asset vulnerabilities.
	Vulnerability	 added to the *IS asset* construct such as *Goal*, *Task*, or *Resource*	Precise vulnerability is not defined. Note: System vulnerabilities are observed when accepting registration and inserting data to the database.
	Threat		Goal Account data received; Plan Use techniques for receiving account data.
	Attack method	 potentially combined with other *Tasks* using *decomposition* links	Plan Use techniques for receiving account data
	Threat agent		Actor Violator
Risk treatment-related concepts	Risk treatment	—	Risk reduction
	Security requirement	 combined using *dependency*, *contribution*, *means-ends*, and *decomposition* links	Goal Account data protected; Plans Make data unreadable, Ensure complex account data, Check entered account data, and Request update account data.
	Control	—	—

The *use cases* Accept registration and Insert account data to database are considered as the *IS assets* since they both support the process of the account data submission.

The *misuser* (i.e., Violator) uses a *threat method* (e.g., misuse case Use techniques for receiving account data). The *IS assets* Accept registration and Insert account data to database are threatened because they have *vulnerabilities* Data not checked for syntactic complexity and Data stored as a plain text (vulnerability construct is adapted from [13]). The security *event* leads to the *impact*, which is defined as Personal information is stolen. This impact harms the *business asset* (i.e., Insert account data to database) and negates the *security criterion* (i.e., Confidentiality of account data). To mitigate the risk, *security use cases* – Ensure complex account data and Make data unreadable – are introduced.

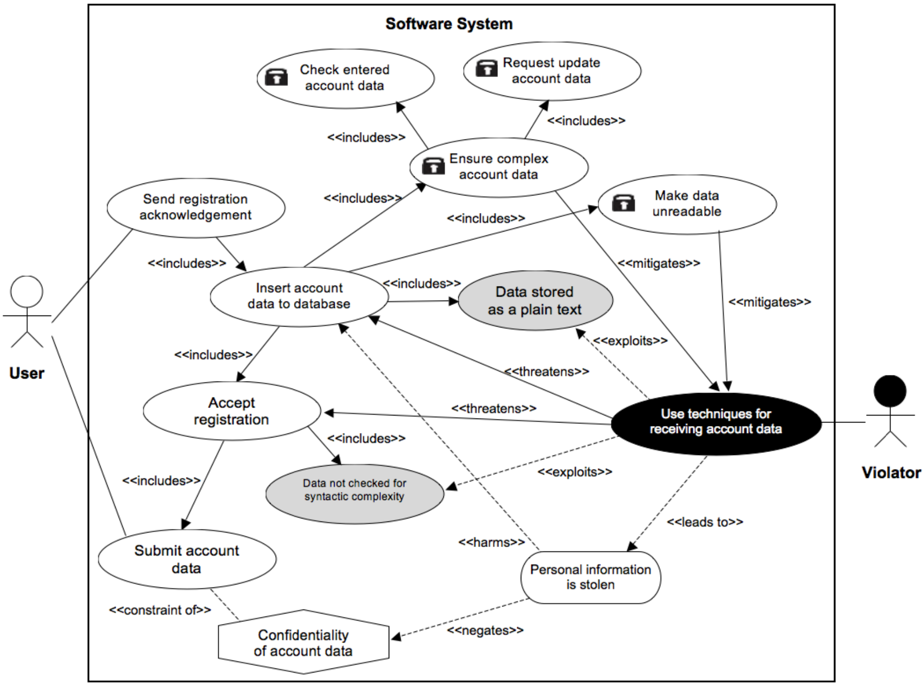


Fig. 4. Security risk management using misuse case diagram

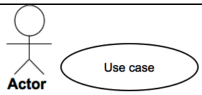
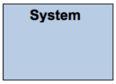
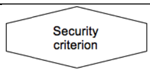
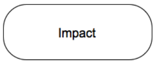
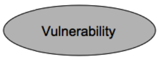
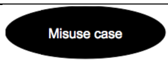
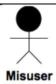
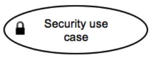
4 Evaluation of SRM Languages and Models

4.1 Understanding Semiotic Clarity: Pre-study

Semantic alignment of the language constructs to the concepts of the ISSRM domain model is shown in Tables 1, 2, and 3 (see columns 0 an 1). According to the principle of the *semiotic clarity* [7] [11], the resource should be a one-to-one correspondence between a visual language construct and its referent concept in the semantic domain. In [8] the semiotic clarity of the SRM languages is considered with respect to the

ISSRM domain model. For instance (see Table 4), we have found that in all languages there is one-to-one correspondence regarding the *Threat agent* concept. However, regarding other ISSRM concepts, the languages contain redundancy (i.e., two language constructs have the same or overlapping semantics), overload (i.e., the same language construct has several meanings), incompleteness (i.e., language does not convey information on a certain phenomenon) and under-definition/excess (i.e., a language construct has no semantics) limitations. It is, however, not a surprise

Table 3. Misuse case diagram construct alignment and extension to the ISSRM concepts

ISSRM domain model		Misuse case diagrams	Examples and notes
0		1	2
Asset-related concepts	Assets		
	Business assets	 combined using *communication*, *extends*, *includes* links	Use case Submit account data; Note: Use cases are also used to express IS assets.
	IS assets		System boundary Software system; Use cases Accept registration, Insert account data to database, and Send registration acknowledgement.
	Security criterion		Security criterion Confidentiality of account data.
Risk-related concepts	Impact		Impact Personal information is stolen. Note: It harms Insert account data to database; It negates Confidentiality of account data.
	Vulnera-bility		Vulnerabilities Data stored as a plain text and Data not checked for syntactic complexity.
	Attack method	 potentially combined with other *misuse cases* using *includes* and *extends* links	Misuse case Use techniques for receiving account data
	Threat agent		Misuse Violator
Risk treatment-related concepts	Risk treatment	—	Risk reduction
	Security requiremen t	 potentially combined using *extends*, *includes* links	Security use cases Make data unreadable, Ensure complex account data, Check entered account data, and Request update account data.
	Control	—	—

because these languages were not designed to deal with security risk management at the first place. While applying them for their primary purpose (i.e., BPMN for business process description; Secure Tropos – secure goal definition; and misuse cases – relation of functional and security requirements), the semiotic clarity analysis could potentially contribute with the completely different results.

4.2 Study Design

In order to answer the research questions defined in Section 1, we have investigated how *human stakeholders* understand the SRM languages. We have conducted an empirical study at the University of Tartu during the course on "Principles of Secure Software Design". The course was taken by the 51 graduate student in their 1st year of study (the 4th year of the whole university curriculum): 5 students were following the

Table 4. Summary of the semiotic clarity results

Semiotic clarity	BPMN	Secure Tropos	Misuse cases
One-to-one correspondence	*Threat agent*	*Threat agent*	*Security criterion, Impact, Vulnerability, Threat agent*
Limitations of the semiotic clarity			
Redundancy	*Assets*	*Event*	*Assets*
Overload	*Assets*	*Assets*	*Assets*
Incompleteness	*Security criterion, Risk, Impact, Event, Vulnerability, Threat, Risk treatment* and *Control*	*Risk, Impact Vulnerability, Threat, Risk treatment,* and *Control*	*Risk, Event, Threat, Risk treatment,* and *Control*
Under-definition (excess)	*Assets, Attack method,* and *Security requirements*	*Assets, Security criterion, Attack method,* and *Security requirements*	*Assets, Attack method,* and *Security requirements*

Computer Science study program, 15 – Software Engineering, and 31 – Cyber-security study program. Treatment of the participants included lectures and workshops on security risk management, security modelling, security requirements, model driven security, and development processes of the secure software. In addition to lectures and workshops, the participants were supported with the related literature (e.g., [1], [3], [9], [10], [17]).

The study consists of few stages. In the first stage, participants were requested to analyse one security risk model (created using either BPMN, Secure Tropos, or misuse cases) consisting of diagrams for *asset*, *risk*, *risk treatment*, and diagram, which combined all three concerns together (see Fig. 2, 3, and 4). Then participants were asked to fill the open-ended questionnaire.

The questionnaire was requesting to identify the actual ISSRM concept expressed in the model (e.g., what is the *business asset*?, what is the *security criterion*?). This analysis contributed to the answer of the first research question (RQ.1). The examples of the correct answers are illustrated in Tables 1, 2, and 3 (columns 2). For instance, the ISSRM *business asset* is or is related to an Account data (i.e., account data per se in the BPMN and Secure Tropos models, and use case Submit account data in misuse case diagram). None of the models represents the ISSRM *control* concept (see Tables 1, 2, and 3), so the expected respondent's answer was "*not represented*". There is also the cases where one ISSRM concept (e.g., *IS asset*) is expressed using the combination of several terms (e.g., Software System, Database, Accept registration, and Insert account data to database, see the BPMN model and Table 1). In this situation respondents had to identify all the statements regarding the concepts in order to show that they comprehend the model.

To assess readers' perceiveness of the SRM languages (i.e., RQ.2), we asked our respondents to write down *language constructs* that are used to model the ISSRM concepts. Respondents had to identify the constructs (or their combinations) as illustrated in Tables 1, 2, and 3. For example, in Secure Tropos, the ISSRM *assets* are expressed using a combination of constructs (i.e., *actor, hardgoal, plan and resource*). We expected that respondents would identify *resource* as used to model *business assets* (because in the given Secure Tropos model, see Fig. 3, only *resource* Account data is used); a combination of *actor* (i.e., Software system), *plan* (i.e., Insert data to database and Accept registration), and *resource* (i.e., Database) constructs is used to express *IS assets*.

In the second stage of the study, participants were requested to use one SRM language (either BPMN, Secure Tropos or Misuse cases) and to create a simple model to manage security risks. The complexity of the requested models corresponds to the ones shown in Figures 2, 3, and 4. In order to answer the third research question (RQ.3) we analysed, which language constructs are used to express different ISSRM concepts. We considered that the language constructs are used correctly if they (or their combination) corresponded to the language construct alignment to the ISSRM concepts (as illustrated in Tables 1, 2, and 3).

4.3 Results

Model Comprehension. We have received 29 responses (10 on the BPMN model, 9 – on Secure Tropos and on 9 misuse case diagrams) and 28 of them were considered as valid responses. We have excluded one response regarding the misuse case diagrams due to the low value of its overall score (6.3%). Obviously, this respondent had the problem in understanding either the given task itself (most probable) or the model; whatever reason was, we considered it as the outlier from our analysis.

In Table 5 we present the results of the *comprehension* of the models created using the SRM languages as the percentage of the concepts correctly identified by the respondents. For instance 100% of the BPMN model regarding the *business asset* means that every respondent correctly identified the *business asset* in the BMPN model.

The findings (see Table 5) indicate that five ISSRM concepts (namely, *business assets*, *IS assets*, *attack method*, *security requirement*, and *control*) are understood better from the security risk-oriented BPMN than other two models. Two ISSRM concepts (i.e., *security criterion* and *vulnerability*) are understood better from the security risk-oriented misuse cases; and one (i.e., *risk treatment*) – security risk-aware Secure Tropos. On the model level, the best-comprehended model is created using BPMN (58%); however the difference with the Secure Tropos and misuse case models is relatively small.

Table 5. Comprehension of the models created using SRM languages

ISSRM concepts		**Security risk-aware/oriented**			**Overall**
		BPMN model	**Secure Tropos model**	**Misuse cases diagrams**	
	Number of valid responses	10	9	9	28
Asset concepts	Business asset	100%	89%	89%	**93 %**
	IS asset	50%	47%	22%	**42 %**
	Security criterion	70%	89%	100%	**86 %**
Risk concepts	Impact	55%	33%	56%	**43 %**
	Vulnerability	40%	22%	56%	**40 %**
	Attack method	80%	44%	33%	**54 %**
	Threat agent	100%	100%	89%	**96 %**
Risk treatment concepts	Risk treatment	40%	44%	33%	**39 %**
	Security requirement	69%	53%	56%	**64 %**
	Control	20%	0%	0%	**7 %**
	Overall	**58 %**	**50 %**	**49 %**	**52 %**

Participant Appropriateness. We have received 39 answers: (*i*) all ten responses (100%) were considered valid regarding the security risk-oriented BPMN; (*ii*) only four out of nine (44%) answers were found valid regarding the security risk-aware Secure Tropos; and (*iii*) six out of ten (60%) responses were found valid regarding the security risk-oriented misuse cases. The major reason for the outliers was the lack of language knowledge (although all participants were following the Master programs, their studies were different regarding the disciplines, and previously they also had finished different Bachelor study programs before), thus, leading to the misinterpretation of the given assignment.

The summarised results of the valid responses are provided in Table 6. Hence, the table presents the percentage of the correctly identified language constructs for the corresponding ISSRM concepts. Five ISSRM concepts (namely, *impact*, *vulnerability*, *attack method*, *risk treatment*, and *security requirement*) are better perceived for the security risk-aware Secure Tropos than other two languages. In addition, three concepts (i.e., *business asset*, *IS asset* and *threat agent*) are equally perceived in all three modelling languages. Only one ISSRM concept is better recognised in BPMN (i.e., *control*) and one in misuse cases (i.e., *security criterion*).

The best perceived SRM language by the participants is Secure Tropos (77%). However one should note that, firstly, only four valid responses are taken into account in this analysis. Secondly, the difference of the overall score among the modelling languages is not large to claim that one SRM language is perceived better than another.

Modeller Appropriateness. This study stage resulted in 10 models created using BPMN, 4 models in Secure Tropos, and 10 misuse case diagrams. In Table 7 we summarise the analysis results. Four ISSRM concepts (i.e., *business assets*, *IS assets*, *impact*, and *security requirements*) were the most correctly expressed in security risk-oriented misuse cases, and two ISSRM concepts (i.e., *vulnerability* and *attack method*) were the most-correctly modelled in Secure Tropos. *Security criterion* and *threat agent*

Table 6. Participant appropriateness of the modelling language

ISSRM concepts		Security risk-aware/oriented			Overall
		BPMN	Secure Tropos	Misuse cases	
	Number of valid responses	10	4	6	20
Asset concepts	Business asset	90%	100%	100%	**95%**
	IS asset	47%	50%	50%	**48%**
	Security criterion	45%	75%	83%	**59%**
Risk concepts	Impact	50%	75%	33%	**33%**
	Vulnerability	60%	100%	67%	**70%**
	Attack method	60%	100%	83%	**75%**
	Threat agent	90%	100%	100%	**95%**
Risk treatment concepts	Risk treatment	50%	75%	17%	**45%**
	Security requirement	90%	100%	67%	**88%**
	Control	40%	25%	33%	**35%**
	Overall	**58%**	**77%**	**58%**	**62%**

Table 7. Modeller appropriateness of the modelling language

ISSRM concepts		Security risk-aware/oriented			Overall
		BPMN	Secure Tropos	Misuse cases	
	Number of created models	10	4	10	24
Asset concepts	Business asset	23%	58%	100%	**42%**
	IS asset	54%	75%	95%	**67%**
	Security criterion	60%	100%	95%	**87%**
Risk concepts	Impact	60%	50%	77%	**70%**
	Vulnerability	50%	100%	80%	**65%**
	Attack method	33%	100%	90%	**68%**
	Threat agent	90%	100%	100%	**96%**
Risk treatment concepts	Risk treatment	-	-	-	**-**
	Security requirement	50%	50%	95%	**64%**
	Control	-	-	-	**-**
	Overall	**46%**	**72%**	**90%**	**67%**

are rather equally expressed both in secure Tropos models and misuse case diagrams. Regarding the separate languages, the best expression of the ISSRM concepts is found by using the security risk-oriented misuse case diagrams (90%) and the worst – using the BPMN (46%).

4.4 Threats to Validity

Conclusion validity deals with the experiment's treatment. The participants were given treatment related to the principles of the secure software design, but not to the SRM languages in particular. Validity also depends on the questionnaire. To mitigate this latter threat one researcher not involved in the experiment has reviewed the questionnaire.

One possible threat is the misinterpretation of the ISSRM concepts. To mitigate, we have provided self-study material about the application of the ISSRM domain model and the use of the SRM languages. In addition we identified data outliers. Having the same person in charge of both the treatment and the experiment design has the advantage that the terminology is more consistent between the given treatment and the one used in the SRM evaluation questionnaire.

A threat to external validity is that, being students, the participants had little real ambition to assess the SRM languages and their models. Hence the motivation for the participating might have been smaller than in the real case. To mitigate this threat, the students were rewarded with the subject points. The case was academic and of a small size; thus the finding might be different in the practical settings. Being students, the respondents had some basic knowledge, but on the other hand quite a lot of the participants had the real industrial experience, as well. Thus, we consider that the participants' expertise is quite close to the level of junior practitioners.

5 Discussion and Conclusion

In this paper we have analysed three SRM languages – extensions of BPMN, Secure Tropos and misuse case diagrams. We have investigated the model comprehension, participant appropriateness and modeller appropriateness of the SRM languages. It should be noted that our purpose is not to criticize any of these languages, but to understand how to manage security risks using them. Below we provide and discuss answers to the research questions.

RQ.1: *How do the human participants comprehend the models for the security risk management*? The best-comprehended ISSRM concepts are *business asset* (93%), *security criterion* (86%), and *threat agent* (96%). The worst comprehended concept is *control* (only 7% of the correct – i.e., *not represented* – answers). "Not even the most brilliant model would be of any use if nobody was able to understand it" [7, pp. 231]. We have observed that comprehension of the analysed models did not depend on the SRM language per se. It is rather driven by the *security risk management* process. For instance, respondents were able to recognise the key ISSRM concepts from the construct labels and connections. However, this shows limitations of the SRM languages. According to [7], model comprehension could be improved through *language perception*. This suggests that the language users (both participants and modellers) have to learn and to understand (*i*) the *constructs* of the SRM languages, and (*ii*) the *construct alignment* to the ISSRM domain model.

RQ.2: *How do the participants perceive (the constructs of) the SRM languages*? We have found that the best perceived constructs are *business asset* (95%), *threat agent* (95%), and *security requirements* (88%). The worst perceived constructs are *impact* (33%) and *control* (35%). As discussed in [7], "the knowledge of the participants is not static, i.e., it is possible to educate persons" to use the SRM languages.

RQ.3: *How do the modellers use (the constructs of) the SRM languages*? Modeller appropriateness is a mean to achieve the semantic quality in the sense that modellers are able to express all relevant knowledge [7]. The *threat agent* (96%) and *security criterion* (87%) are two ISSRM concepts that were the most correctly expressed using the SRM languages. The worst expressed concept was *business asset* (42%). Modeller appropriateness characterises the active use of the SRM languages. Potential improvements towards the modeller appropriateness could be definition of better construct discriminability [11]. For instance, the better-discriminating constructs should be defined for *business assets*, *security requirements*, *impact* and *vulnerability*. Other principles for language notation design, as suggested in [11], could potentially also be applied in this context, too.

As the future work, we plan to perform similar analysis in order to validate our observations. The systematic research is also needed to understand the link between different quality types of the SRM languages.

References

1. Altuhhova, O., Matulevičius, R., Ahmed, N.: An Extension of Business Process Model and Notation for Security Risk Management. Accepted at the International Journal of Information System Modelling and Design (2013)
2. Bresciani, B., Perini, A., Giorgini, P., Fausto, G., Mylopoulos, J.: TROPOS: An Agent oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems 25, 203–236 (2004)
3. Dubois, E., Heymans, P., Mayer, N., Matulevičius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management. In: Nurcan, S., Salinesi, C., Souveyet, C. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 289–306. Springer, Heidelberg (2010)
4. Firesmith, D.G.: Engineering Safety and Security Related Requirements for Software Intensive Systems. In: ICSE 2007 Tutorial (2007)
5. Jurjens, J.: Secure Systems Development with UML. Springer, Heidelberg (2005)
6. Elahi, G., Yu, E.: A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 375–390. Springer, Heidelberg (2007)
7. Krogstie, J.: Model-Based Development and Evolution of Information Systems - A Quality Approach. Springer (2012)
8. Matulevičius, R.: Comparing Modelling Languages for Information Systems Security Risk Management. In: Seyff, N., Koziolek, A. (eds.) Modelling and Quality in Requirements Engineering: Essays Dedicated to Martin Glinz on the Occasion of His 60th Birthday, pp. 207–220. Monsenstein and Vannerdat, Münster (2012)
9. Matulevičius, R., Mouratidis, H., Mayer, N., Dubois, E., Heymans, P.: Syntactic and Semantic Extensions to Secure Tropos to Support Security Risk Management. Journal of Universal Computer Science 18(6), 816–844 (2012)
10. Mayer, N.: Model-based Management of Information System Security Risk. Ph.D. thesis, University of Namur (2009)
11. Moody, D.L.: The "Physics" of Notations: Towards a scientific basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Eng. 35, 756–779 (2009)

12. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), 285–309 (2007)
13. Røstad, L.: An Extended Misuse Case Notation: Including Vulnerabilities and The Insider Threat. In: Proceedings of the REFSQ (2006)
14. Silver B.: BPMN Method and Style: A Levels-based Methodology for BPMN Process Modeling and Improvement using BPMN 2.0. Cody-Cassidy Press (2009)
15. 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)
16. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering Journal 10(1), 34–44 (2005)
17. Soomro, I., Ahmed, N.: Towards Security Risk-orineted Misuse Cases. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 689–700. Springer, Heidelberg (2013)

Classification and Qualitative Analysis of Non-Functional Requirements Approaches

M. Mahmudul Hasan, Pericles Loucopoulos, and Mara Nikolaidou

Harokopio University of Athens, Athens, Greece
{m.hasan,p.loucopoulos,mara}@ hua.gr

Abstract. A considerable number of methods and tools have been proposed for the treatment of non-functional requirements (NFRs). There is ample evidence that NFRs play a significant role in the Information Systems Engineering process. However, there is surprisingly an absence of an agreed position regarding the definition of NFRs, their classification and presentation. This paper reports on a systematic literature review of the documented NFR approaches, classifies these approaches according to different criteria and provides a qualitative analysis of their scopes and characteristics. The results of this analysis can serve system developers as the means of deriving appropriate methods and tools of NFRs engineering process in the system development.

Keywords: Requirement Engineering; Non-functional requirements; NFRs; NFRs approach; Systematic Literature Review.

1 Introduction

Requirements Engineering (RE) is arguably one of the most challenging area in system development with many challenges still remaining [1]. Within RE, the treatment of Non-Functional Requirements (NFRs) has received much but fragmented attention. The primary agenda of NFRs research is to deal with quality aspects of the target system, be it a business process or a software system. NFRs are often ignored and inadequately specified and rarely treated as first-class elements as Functional Requirements (FRs) [2]. It is not too farfetched to state that the reasons that can help us understand why these approaches are not explicitly dealt with in the system development projects are the high abstraction level and lack of understanding of their positions, operational scopes and characteristics [3].

There is certainly a paucity of studies that identifies, classifies and analyzes the documented NFRs methods and techniques and discusses their positions, scopes and characteristics. The work presented in this paper is partly based on the study by Loucopoulos et al. [2] that classified NFRs approaches into discovery, specification, negotiation and validation & verification phases of NFRs engineering process. However, the classification has its own credits and shortcomings. This paper augments this work by presenting a systematic literature review (SLR) of documented NFRs approaches, classifying these approaches and providing a qualitatively discussion of

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 348–362, 2014.

their scopes, characteristics and merits. The findings of this paper can serve system developers as the means of deriving appropriate methods and tools of NFRs engineering process of a system development based on the scopes and characteristics of the solution design.

This paper is organized as follows. First, the research methodology is defined and described the strategic operations and introduced the classification scheme. Second, the selected papers are categorized and discussed according their relevance to the categories of classification scheme. Third, an analysis of the results is provided to answer the research questions. Finally, the paper concludes with reflections and suggestions for future research.

2 Methodology

This study is based on a systematic literature review described in [4] to investigate and classify primary studies in the area of NFRs. The SLR process can be one of the two types [5], the review aggregates results related to a specific research question named conventional SLR and the review finds and classifies the primary studies in a specific research topic named mapping SLR. This study falls into the mapping category of SLR and follows the systematic steps (Fig.1) suggested by Kitchenham [4] and Petersen [6]. Mapping SLR is the best choice of research methodology because this study undertakes qualitative analysis of NFRs approaches by illustrating them in a tabular form of specific categories and discussing their relative characteristics other than quantitative-based statistical form of analysis led by conventional SLR [5].

Fig. 1. Systematic Literature Review [4-6]

2.1 Research Questions (RQs)

The focus of the RQs is to identify and classify documented methods and techniques in NFRs engineering process phases and discuss their operational scopes and characteristics. This study addresses the following particular research questions:

RQ1. What are the documented approaches in NFRs engineering process?
RQ2. What are the scopes and characteristics of these NFRs approaches?

To address RQ1, this paper identifies and classifies the documented approaches in different NFRs engineering process phases suggested by [7]. In order to derive appropriate methods and tools for the systematic treatment of quality requirements in NFRs engineering process of a system development, it is essential to identify the scope of the approaches, i.e. how and in what circumstances the methods and tools operate in different NFRs engineering process phases. Therefore, RQ2 addresses the scopes and characteristics of the NFRs approaches. The researchers argue for three different scopes of NFRs solution design. First, integrate goals in the solution design suitable for the large and complex system development environment to show the interrelations of domain elements and alternative design decisions [8]. Second, aspect-oriented solution design promotes automated tools and methods to separate concerns and concentrate one concern at a time to reduce system development complexities [9]. And third, pattern-based solution designs are suitable in the process of using previous practice assuming similar kind of system development environment to optimize available experience in shortening required time and effort of dealing NFRs [10]. These three scopes are selected in the classification process to discuss the common characteristics of solution designs in each scope.

2.2 Search Strategy

Four databases, IEEE Xplore, Springer Link, Science Direct and ACM digital library selecting all journals and conference proceedings were explored in the literature search process. DBLP citation index and Google Scholar search engine were also explored in order to find relevant literature. A comprehensive set of keywords was generated based on the documented definitions and terminology of NFRs [11, 12]. The keywords were used in the search process applying its synonyms as well as combination and altering the word's order to identify the utmost number of relevant primary studies. For example:

- ("Non-functional") AND ("Requirements" OR "Characteristics" OR "Attributes" OR "Properties" OR "Restrictions" OR "Constraints").
- ("Quality") AND ("Requirements" OR "Characteristics" OR "Attributes" OR "Properties" OR "Restrictions" OR "Constraints")

The literature search was conducted in December 2013 and there was no time frame limitation of the research publication, i.e. publications from any year were considered. Backward and forward search procedures were also used in the literature search in order to obtain relevant citation of the articles found by the literature search. Backward search was performed by reviewing relevant citation in the reference list of identified literature. Forward search was performed by reviewing the literature that cited the identified literature. Furthermore, author citation index of the selected literature was also reviewed to find the relevant literature on NFRs.

2.3 Inclusion and Exclusion Criteria

The inclusion criteria of the literature selection were solely correlated to the relevance of NFRs to answer the research questions. The literatures were included based on:

- The publications written only in English language were considered.
- The abstracts explicitly in the notion of NFRs as a primary focus were considered.

The exclusion criteria were based on the deceive concept of NFRs to answer the research questions. The literatures were excluded based on:

- If there were more publications of the same Research Group on the same approach development then the most complete publication of the approach were considered.
- The literatures those were not considered NFRs as the primary contribution.

2.4 Quality Assessment

In order to ensure the validity of search strategy, literature selection and data extraction, multiple discussions were carried out among the authors to decide the search strategy and inclusion/exclusion criteria of primary studies. Furthermore, Kitchenham [4] suggests for an investigation of documented SLR studies in identical research fields to validate the search strategy. A recent study [2] employed SLR in the field of NFRs was reviewed and observed the reference list to assess the volume of potential relevant primary studies and validate the adopted search strategy of this paper. Once the relevant primary studies on NFRs have been obtained through the adopted search strategy, the primary studies need to be assessed for their actual relevance to provide evidence of answering the research questions [13]. Since all the information from selected primary studies is not obviously relevant to the research questions hence the relevant information needs to be extracted avoiding the likelihood biasness. Therefore, data extraction criteria were discussed among the authors to ensure the reliability of data extraction of answering the research questions.

2.5 Literature Selection and Data Extraction

A total of 372 papers were found from the results of different searches and initial screening based on title, abstract and keywords. From these 120 papers address NFRs as primary contribution were selected based on inclusion and exclusion criteria and 92 papers of NFRs approach development were finally selected for data extraction. The data were extracted reading abstract, introduction and conclusion of the literatures to identify the contributions. Irrelevant literatures were excluded attaching a short description of its rationale. The extracted data from each article includes:

1. Literature reference;
2. Name of the approach;
3. Process of the approach;
4. Scope of the approach;
5. Other characteristics of the approach.

The approaches are categorized in five NFRs engineering process phases [7]: elicitation of the requirements in system development, specify their necessary details in informal documentation language, prioritization among alternative requirements, modelling the requirements in formal languages, and finally validation and verification of the quality of specified requirements.

2.6 Data Analysis

This paper presents the documented NFRs approaches according to their respective phases of NFRs engineering process and scope of the solution design. And, qualitatively analyze and discuss the results and answer the research questions accordingly. The field ‘name of the approach’ and ‘process of the approach’ answer RQ1 discussing NFRs approaches in different process phases. The field ‘scope of the approach’ and ‘other characteristics of the approach’ address RQ2 discussing the scopes and characteristics of the documented NFRs approaches.

3 Results

The results are presented in two dimensions. First, illustrate the results in a tabular format (table1 to table5) based on the NFRs engineering process and scope of each approach. Second, discuss their classifications and characteristics in details. Tables are generated into three sections to illustrate the approach’s scope in each NFRs engineering process phase. Each entry of the approach is given a name tagged with its reference. Some of the approaches are explicitly named by the designer of the solutions and others are given a name within the notion of solution design to provide an easy way of distinguishing them beyond their actual reference.

Table 1. Elicitation of NFRs

Goal-oriented	Aspect-oriented	Pattern-based
• Use-case Questioner[14] • MOQARE[15] • NFRs elicitation model[16] • Actor–based model[17] • Usability catalogue[18] • NFRs Layered Framework[19] • Goal-based requirement extraction[20]	• Usability Elicitation Framework[21] • NFR Classifier[22] • Semi supervised text Analysis[23] • QA-Miner[24] • NFR Incorporation Framework[25] • Speech recognition[26]	• Experience-based Method[27] • NFRs elicitation Framework [28] • ElicitO[29] • Efficiency use-cases[30] • NFR recomm-endation[31]

In goal-oriented elicitation approach, goal-based questionnaire is proposed to extract NFRs by questioning stakeholders [14, 15], business process model [16], use-case of domain model [17], and taxonomy [18] in the system development. Goal decomposing [19] and goal analysis [20] methods are proposed to identify stakeholder, generate their expected goals based on developer’s knowledge and experience, then decompose the goals into sub-goals and identify NFRs for each sub-goals. In aspect-oriented approach, automated [21, 22] and semi supervised [23-25] text analysis, speech detection [26] tool-based elicitation techniques are proposed to identify NFRs from available textual requirements documents and in the form of oral documents (e.g. meeting minutes, interview notes, and memos). In pattern-based approach, experience-based elicitation [27], domain ontology [28-30] are proposed to assist requirements analysts in NFRs elicitation process. Recommendation system [31] is also proposed to understand expected realistic NFRs in the system development.

Table 2. Specification of NFRs

Goal-oriented	Aspect-oriented	Pattern-based
• Usability Elicitation Framework[21] • Performance evolution model[32]	• NFR classifier[33] • NFR Locator[34]	• NFR catalogue[35] • NFR taxonomy[36] • Usability catalogue[18] • NFR classification[37]

In goal-oriented approach, requirements are illustrated in hand-drawn user interface looks and specify necessary details of the requirements [21, 32]. In aspect-oriented approach, requirements specification techniques are proposed for automated requirements specification and categorization from a wide variety of requirements document [33, 34]. In pattern-based approach, NFRs documentation based on requirements analyst's prior experience on a particular situation of NFRs handling [35], NFR taxonomies [18, 36, 37] are proposed to guide requirements analysts in elicitation process.

Table 3. Prioritization of NFRs

Goal-oriented	Aspect-oriented	Pattern -based
• sureCM Framework[38] • sureCM Framework for security-usability conflicts resolution[39] • Analytical Hierarchy Process [40] • Matrix map conflicts[41] • Quality Attribute Risk and Conflict Consultant [42] • Quantifying NFRs[43] • Business rules [44] • Prioritized system QAs [45] • Pareto Algorithm[46] • Context-aware recommend [47] • FQQSIG model [48]	• NFR trade-off profiling[49]. • Personal Construct Theory[50] • Architecture-driven requirements prioritization[51] • NFR prioritization algorithm[52]	• NFRs conflicts catalogue[53] • NFR dependency classification[54] • NFR conflicts analysis[55] • Constraint hierarchy trade-off[56]

In goal-oriented prioritization approach, conflict analysis [38-43], business rules [44], user satisfaction priority lists [45, 46] based techniques are proposed to deal with NFRs interdependencies towards trade-off and prioritization. Techniques are also proposed to make automatic trade-off decision [47, 48] among NFRs alternatives based on their relative interdependencies. In aspect-oriented solution design, several approaches are proposed to provide required information about NFRs and its correlation [49-51], user satisfaction [52] to the process of trade-off analysis and requirements prioritization. In pattern-based approach, catalogue of potential NFRs conflicts [53], classification of NFRs dependencies [54], conflicts identification from prior experience [55], quality constraint hierarchy [56] are proposed to aid conflict analysis in the selection among alternatives.

Table 4. Modelling of NFRs

Goal-oriented	Aspect-oriented	Pattern-based
• NFR Representation multi-model [57] • Ontology based Quality modelling [58, 59] • NFR use-case Model [60, 61] • NFR integration Framework[62] • NFR use cases and scenarios models[8] • NFR Framework[63-69]	• Activity-based quality model [70] • Efficiency use-cases[30] • Ontology-based NFR conceptualization[71] • NFR Traceability model[72] • QRA Framework[73, 74] • NFR integration Framework[75, 76] • ProcessNFL language[77] • UML Profile[78]	• Quality requirements BP framework[79-84] • NoFun language [85]

Several goal-oriented modelling approaches are proposed to represent the interrelations of different system development viewpoints [57], domain ontology and NFRs [58, 59], functional and non-functional requirements [60-62], soft-goal interdependencies of NFRs [8, 63-69]. Aspect-oriented activity-based modelling approaches are proposed to illustrate the elicitation process [30, 70], communication process of intra and interrelations among NFRs dependencies and functional requirements [71-76] for better traceability. Some approaches are proposed to develop new representation language [77] and extend an existed language [78] to describe NFRs properties in the system development process. In pattern-based modelling, approaches are proposed to illustrate the visibility of quality requirements in the operation process of business model [79-84] to aid requirements elicitation and evaluation process. A language is proposed to define ISO/IEC quality characteristics in different system development contexts for better requirements understanding [85].

Goal-oriented approaches are proposed to measure the adequacy and quality of NFRs in requirements specification using domain knowledge [86, 87], abstract interpretation [88], interrelations of NFRs [89], quantitative size and effort estimation [90, 91], goal-centric traceability links between NFRs [92], and reasoning on NFRs in different contexts [93]. Also, some approaches are proposed to evaluate and validate system behaviour conflicts [94], changing requirements during system development [95] and relative priority of NFRs in trade-off analysis [96]. In aspect-oriented approach, a text mining tool is proposed to identify possible defects for the measurement of NFRs quality in the specification document [97]. And, an evaluation approach is proposed to evaluate NFRs specification by the clarity of its description [98]. Pattern-based approaches are proposed for the assessment of NFRs specification based on experience accumulated from similar project development [99-102], knowledge of the requirements characteristics and catalogue in a particular domain [103, 104].

Table 5. Validation and Verification (V&V) of NFRs

Goal-oriented	Aspect-oriented	Pattern-based
• Spectrum analysis [86] • TCM Framework [87] • Abstract Interpretation-based verification[88] • EMIMCE model[89] • COSMIC-FFP method[90] • Quantitative measure[91] • Goal-centric traceability[92] • Automate verification[93] • Execution-based Model Checking[94] • Goal decomposition[95] • Quantitative priority assessment framework[96]	• QR mining framework[97] • NFR Evaluation Model[98]	• Bayesian Belief Network [99] • Bayesian Reliability Prediction[100] • Model-based approach[101] • NFR pattern approach[102] • Performance Requirements Framework[103] • Scenario-based assessment[104]

4 Analysis

In this section the available NFRs approaches are discussed in the scope of its common characteristics in each NFRs engineering process phase. The elicitation approaches promote three ways of NFRs extraction from different sources depending on the suitable context of system development. Goal-oriented approaches are appropriate in the process of asking goal based questionnaire to the NFRs sources, e.g. stakeholder, domain model, NFRs taxonomies, etc. Pattern-based approaches are suitable in the process of using expertise knowledge and skills of the system developers from their experience in similar kind of system development environment to extract NFRs. And, aspect-oriented approaches are the automated and semi supervised tools suitable for identifying NFRs from textual or oral documents.

The scope of NFRs specification process supports either NFRs elicitation process or requirements prioritization process. Pattern-based approaches aid NFRs elicitation process providing domain specific and generic NFRs taxonomies generated from previous experience to guide elicitation process. The elicited requirements, i.e. outcomes of the elicitation process are classified and documented according to their relevance from a wide variety of requirements by automated aspect-oriented methods and tools. Goal-oriented approaches specify necessary details of the documented requirements according to the context of system development. Therefore, both the categorized NFRs document outcomes of the aspect-oriented approaches and their specified details outcomes of the goal-oriented approaches help system developers understand requirements characteristics in the process of requirements prioritization.

The scope of NFRs prioritization process is divided into two prioritization activities. Aspect-oriented and pattern-based approaches provide reasons of NFRs conflicts describe their relative concerns and generate their compositions without being engaged in the core process of conflicts analysis and NFRs prioritization. The information of requirements conflicts are generated from two sources. Aspect-oriented approaches are suitable in the process to provide dynamic context aware information of NFRs interdependencies and user satisfaction priorities of the system development. Pattern-based approaches are suitable to provide stationary information from the catalogue and prior recorded organizational experience. And, goal-oriented approaches are designed to perform the actual trade-off analysis and prioritize conflicting NFRs.

The scope of goal-based modelling illustrates viewpoints and use cases to represent various relationships among NFRs and activities to deal with NFRs with a specific purpose in mind. Goal-oriented modelling approaches are suitable for conceptual modelling to represent NFRs concepts in an organized manner. Aspect-oriented modelling approaches are suitable for mainly two modelling activities. First, visualize the intra relationships and dependencies among NFRs and interrelationships to functional requirements (FRs) in the process of integration and tracing NFRs into FRs of a system development. Second, develop and extend NFRs representation languages to describe NFRs and design decisions for maintaining traceability among them. And, pattern-based NFRs modelling approaches are suitable to describe NFRs and its visibility into another model, for example illustrate NFRs in business process model to aid software developer in capturing NFRs.

The scope of validation and verification process is mainly addressed into two types of activities in methods and techniques development. First, ensure the quality of NFRs specification is of high quality, i.e. identify missing and unnecessary requirements in the specification document. Goal-oriented approaches measure the adequacy and necessity of NFRs in requirements specification based on the interrelations of NFRs, the relative advantages and disadvantages of NFRs, etc. Aspect-oriented approaches evaluate the quality of requirements specification based on possible defects and clarity of requirement's descriptions in the specification document. Pattern-based approaches assess the quality of requirements specification based on the experience accumulated from similar project development knowledge of the requirements characteristics and catalogue in a particular domain. Second, some goal-oriented approaches are also appropriate to verify NFRs conflicts and priorities.

5 Conclusion

This paper systematically reviews the documented approaches dealing with NFRs in system development. The main contribution of this paper is to classify these approaches into its respective positions in NFRs engineering process and discuss their scopes and characteristics to guide system developers deriving appropriate methods and tools for the treatment of NFRs in system development. The review shows that methods and techniques are available in all NFRs engineering process phases and the approaches are developed within various scopes and characteristics. Elicitation approaches are designed to elicit NFRs from goal-oriented dynamic and pattern-based static sources of requirements with aspect-oriented methods and tools. NFRs taxonomies are developed in pattern-based specification approaches, aspect-oriented specification approaches list and classify elicited NFRs where goal-oriented approaches specify necessary details of NFRs in a system development. Necessary information for the goal-oriented prioritization approaches are provided by aspect-oriented and pattern-based approaches. Goal-oriented modelling approach illustrates various NFRs concepts of system development, aspect-oriented modelling visualizes the dependencies among NFRs, and pattern-based modelling illustrates NFRs visibility into another model for ease the NFRs elicitation process. The adequacy and necessity of NFRs in requirements specification are assessed by the information of goal-oriented approaches and available experience of pattern-based approaches. Aspect-oriented approaches evaluate the quality of specification document. Some goal-oriented approaches also verify the requirement's conflicts and priorities. In overall, the analysis of the positions, scopes and characteristics of documented NFRs approaches would be useful for system developers to find the appropriate methods and techniques in handling NFRs engineering process of a system development. However, there is much work to do in the systematic process of NFRs engineering since all activities are isolated and disorderly sequenced of various methods and tools. Our future work puts forward the design of a comprehensive NFRs meta-modelling architecture of sequentially ordered activities with suitable methods and techniques of each process in NFRs engineering phase.

Reference

1. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The brave new world of design requirements. Information Systems 36(7), 992–1008 (2011)
2. Loucopoulos, P., Sun, J., Zhao, L., Heidari, F.: A Systematic Classification and Analysis of NFRs. In: 19th Americas Conf. on Information Systems, Chicago, Illinois (2013)
3. Paech, B., Kerkow, D.: Non-functional requirements engineering - quality is essential. In: 10th Int. Workshop on RE Foundation for Software Quality (2004)
4. Kitchenham, B.: Procedures for performing systematic reviews. Keele University (2004)
5. Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., Linkman, S.: Systematic literature reviews in software engineering – A tertiary study. Information and Software Technology 52(8), 792–805 (2010)
6. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th Int. Conference on Evaluation and Assessment in Software Engineering (2008)
7. Valaski, J., Reinehr, S., Malucelli, A.: The Role of Ontologies in Software Requirements Engineering: A Systematic Review
8. Cysneiros, L.M., Leite, J.: Driving Non-Functional Requirements to Use Cases and Scenarios. In: 15th Brazilian Symposium on Software Engineering (2001)
9. Umar, M., Khan, N.A.: Analyzing Non-Functional Requirements (NFRs) for software development. In: 2nd IEEE Int. Conf. on Software Engineering and Service Science 2011 (2011)
10. Kaiya, H., Sato, T., Osada, A., Kitazawa, N., Kaijiri, K.: Toward quality requirements analysis based on domain specific quality spectrum. In: ACM Symposium on Applied Computing, pp. 596–601. ACM, Fortaleza (2008)
11. Glinz, M.: On non-functional requirements. In: 15th IEEE Int. Requirement Engineering Conference (2007)

12. Mairiza, D., Zowghi, D., Nurmuliani, N.: An Investigation into the Notion of Non-Functional Requirements. In: ACM Symposium on Applied Computing, NY, USA (2010)
13. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering, EBSE Technical Report EBSE-2007-01 (2007)
14. Rahman, M.M., Ripon, S.: Elicitation and Modeling Non-Functional Requirements–A POS Case Study. International Journal of Future Computer and Communication 2(5), 485–489 (2013)
15. Herrmann, A., Peach, B.: MOQARE: Misuse-oriented quality requirements engineering. Requirements Engineering 13(1), 73–86 (2008)
16. Jaramillo, A.F.: Non-functional requirements elicitation from business process models. In: 5th Int. Conf. on Research Challenges in Information Science, pp. 1–7 (2011)
17. Mala, G.S.A., Uma, G.V.: Elicitation of Non-functional Requirement Preference for Actors of Usecase from Domain Model. In: Hoffmann, A., Kang, B.-H., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 238–243. Springer, Heidelberg (2006)
18. Cysneiros, L.M., Werneck, V.M., Kushniruk, A.: Reusable Knowledge for Satisficing Usability Requirements. In: IEEE Int. Conf. on RE, pp. 463–464 (2005)
19. Rao, A.A., Gopichand, M.: Four Layered Approach to Non-Functional Requirements Analysis. International Journal of Computer Science Issues 8(6.2), 371–379 (2011)
20. Antón, A.I., Bolchini, D., He, Q.: The Use of Goals to Extract Privacy and Security Requirements from Policy Statements. In: 26th Int. Conf. on Software Engineering (2003)
21. Rivero, L., Marczak, S., Conte, T.: An Approach for the Elicitation of Usability Requirements in the Development of Web Applications. In: CEUR Workshop Proceeding of Requirements Engineering. Rio de Janeiro, Brazil (2013)
22. Cleland-Huang, J., Settimi, R., Zou, X., Solc, P.: Automated classification of non-functional requirements. Requirements Engineering 12(2), 103–120 (2007)
23. Casamayor, A., Godoy, D., Campo, M.: Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. Information and Software Technology 52(4), 436–445 (2010)
24. Rago, A., Marcos, C., Diaz-Pace, J.A.: Uncovering quality-attribute concerns in use case specifications via early aspect mining. Journal of RE 18(1), 67–84 (2013)
25. Song, X., Duan, Z., Tian, C.: Non-Functional Requirements Elicitation and Incorporation into Class Diagrams. In: Shi, Z., Vadera, S., Aamodt, A., Leake, D. (eds.) IIP 2010. IFIP AICT, vol. 340, pp. 72–81. Springer, Heidelberg (2010)
26. Steele, A., Arnold, J., Cleland-Huang, J.: Speech Detection of Stakeholders' Non-Functional Requirements. In: 1st Int. Workshop on Multimedia RE (2006)
27. Doerr, J., Kerkow, D., Koenig, T., Olsson, T., Suzuki, T.: Non-functional requirements in industry - Three case studies adopting an experience-based NFR method. In: 13th IEEE Int. Conf. on Requirements Engineering (2005)
28. Wang, T., Si, Y., Xuan, X., Wang, X., Yang, X., Li, S., Kavs, A.J.: A QoS ontology cooperated with feature models for non-functional requirements elicitation. In: 2nd Asia-Pacific Symposium on Internetware, pp. 1–4. ACM, Suzhou (2010)
29. Al Balushi, T.H., Sampaio, P.R.F., Dabhi, D., Loucopoulos, P.: ElicitO A Quality Ontology-Guided NFR Elicitation. In: 13th Int. Conf. on Requirements Engineering Foundations for Software, Trndheim, Norway (2007)
30. Dörr, J., Kerkow, D., Knethen, A., Paech, B.: Eliciting Efficiency Requirements with Use Cases. In: REFSQ, pp. 37-46 (2003)
31. Zhang, X.-L., Chi, C.-H., Chen, D., Wong, R.K.: Non-functional Requirement Analysis and Recommendation for Software Services. In: 20th Int. Conf. on Web Services (2013)

32. Ho, C.W., Johnson, M.J., Williams, L., Maximilien, E.M.: On Agile Performance Requirements Specification and Testing. In: Agile Conference (2006)
33. Rashwan, A., Ormandjieva, O., Witte, R.: Ontology-Based Classification of Non-functional Requirements in Software Specifications: A New Corpus and SVM-Based Classifier. In: 37th Annual Computer Software and Applications Conference, pp. 381–386 (2013)
34. Slankas, J., Williams, L.: Automated extraction of non-functional requirements in available documentation. In: 1st Int. Workshop on Natural Language Analysis in Software Engineering (2013)
35. Nalchigar, S., Salay, R., Chechik, M.: Towards a Catalog of Non-Functional Requirements for Model Transformations. University of Toronto (2013)
36. Galster, M., Bucherer, E.: A Taxonomy for Identifying and Specifying Non-Functional Requirements in Service-Oriented Development, pp. 345-352 (2008)
37. Odeh, Y., Odeh, M.: A New Classification of Non-Functional Requirements for Service-Oriented Software Engineering (2011)
38. Mairiza, D., Zowghi, D., Gervasi, V.: Conflict characterization and Analysis of Non Functional Requirements: An experimental approach. In: 12th IEEE Int. Conf. on Intelligent Software Methodologies, Tools and Techniques (2013)
39. Mairiza, D., Zowghi, D.: An Ontological Framework to Manage the Relative Conflicts between Security and Usability Requirements (2010)
40. Kassab, M.: An integrated approach of AHP and NFRs framework. In: 7th IEEE Int. Conf. on Research Challenges in Information Science (2013)
41. Abdul, H., Jamil, A., Imran, U.: Conflicts Identification among Non-functional Requirements using Matrix Maps. In: World Academy of Science Engineering and Technology 44, pp. 1004–1009 (2010)
42. Boehm, B., In, H.: Aids for Identifying Conflicts Among Quality Requirements. IEEE Software 13(2), 25–35 (1996)
43. Hill, R., Wang, J.: Quantifying non-functional requirements: a process oriented approach. In: RE 2004, pp. 352–353 (2004)
44. Liu, C.L.: Ontology-Based Conflict Analysis Method in Non-functional Requirements. In: 9th Int. Conference on Computer and Information Science (2010)
45. Dabbagh, M., Sai Peck, L.: A Consistent Approach for Prioritizing System Quality Attributes. In: 14th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (2013)
46. Aguilar, J.A., Garrigós, I., Mazón, J.-N.: A Goal-Oriented Approach for Optimizing Non-functional Requirements in Web Applications. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 14–23. Springer, Heidelberg (2011)
47. Danylenko, A., Lowe, W.: Context-aware recommender systems for non-functional requirements. In: 3rd Int. Workshop on Recommendation Systems for SE (2012)
48. Zhu, M.X., Luo, X.X., Chen, X.H., Wu, D.D.: A non-functional requirements tradeoff model in Trustworthy Software. Information Sciences 191, 61–75 (2012)
49. Saadatmand, M., Cicchetti, A., Sjodin, M.: Toward Model-Based Trade-off Analysis of Non-functional Requirements. In: 38th EUROMICRO Conf. on Software Engineering and Advanced Applications 2012 (2012)
50. González-Baixauli, B., do Leite, J.C.S.P., Laguna, M.A.: Eliciting Non-Functional Requirements Interactions Using the Personal Construct Theory. In: 14th IEEE Int. Requirements Engineering Conference (2006)

51. Koziolek, A.: Research Preview: Prioritizing Quality Requirements Based on Software Architecture Evaluation Feedback. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 52–58. Springer, Heidelberg (2012)
52. Thakurta, R.: A framework for prioritization of quality requirements for inclusion in a software project. Software Quality Journal 21(4), 573–597 (2013)
53. Mairiza, D., Zowghi, D.: Constructing a Catalogue of Conflicts among Non-functional Requirements (2010)
54. Mehta, R., Ruiz-Lopez, T., Chung, L., Noguera, M.: Selecting among alternatives using dependencies: An NFR approach. In: 28th Annual ACM Symposium on Applied Computing, pp. 1292–1297. ACM, Coimbra (2013)
55. Sadana, V., Liu, X.F.: Analysis of Conflicts among Non-Functional Requirements Using Integrated Analysis of Functional and Non-Functional Requirements. In: 31st Annual Int. Computer Software and Applications Conference (2007)
56. Guan, Y., Ghose, A.K.: Use constraint hierarchy for non-functional requirements analysis. In: Lowe, D., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 104–109. Springer, Heidelberg (2005)
57. Gonzalez-Huerta, J., Insfran, E., Abraho, S., McGregor, J.D.: Non-functional requirements in model-driven software product line engineering. In: 4th Int. Workshop on Non-Functional System Properties in Domain Specific Modeling Languages, Innsbruck, Austria, pp. 1–6 (2012)
58. Jingbai, T., Keqing, H., Chong, W., Wei, L.: A Context Awareness Non-functional Requirements Metamodel Based on Domain Ontology. In: IEEE Int. Workshop on Semantic Computing and Systems (2008)
59. Heidari, F., Loucopoulos, P., Brazier, F.: Ontology for quality specification in requirements engineering. In: 4th Int. Conf. on Models and Ontology-based Design of Protocols, Architectures and Services, MOPAS 2013 (2013)
60. Supakkul, S., Chung, L.: Integrating FRs and NFRs A Use Case and Goal Driven Approach. In: Int. Conf. on Soft. E Research, Management & Applications, pp. 30–37 (2005)
61. Chung, L., Supakkul, S.: Representing NFRs and FRs: A goal-oriented and use case driven approach. In: Dosch, W., Lee, R.Y., Wu, C. (eds.) SERA 2004. LNCS, vol. 3647, pp. 29–41. Springer, Heidelberg (2006)
62. Cysneiros, L., Leite, J., Neto, J.: A framework for integrating non-functional requirements into conceptual models. Requirements Engineering 6(2), 97–115 (2001)
63. Chung, L., Nixon, B.A.: Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach. In: 17th Int. Conf. on Soft. Engineering (1995)
64. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Transactions on Software Engineering 18(6), 483–497 (1992)
65. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering. Int. Series in Software Engineering 5, 476 (2000)
66. Chung, L., Nixon, B.A., Yu, E.: Using non-functional requirements to systematically support change. In: 2nd IEEE Int. Symposium on RE, pp. 132–139 (1995)
67. Kassab, M., Daneva, M., Ormandjieva, O.: Scope management of non-functional requirements. In: 33rd EUROMICRO Conf. on Software Engineering and Advanced Applications 2007, pp. 409–417 (2007)
68. Yrjönen, A., Merilinna, J.: Extending the NFR framework with measurable non-functional requirements. In: 2nd Int. Workshop on Non-functional System Properties in Domain Specific Modeling Languages, Denver, Colorado, USA (2009)

69. Affleck, A., Krishna, A., Achuthan, N.R.: Optimal Selection of Operationalizations for Non-Functional Requirements. In: 9th Asia-Pacific Conf. on Conceptual Modelling, Adelaide, Australia, pp. 69–78 (2013)
70. Wagner, S., Deissenboeck, F., Winter, S.: Managing quality requirements using activity based quality models. In: 6th Int. Workshop on Software Quality, NY, USA (2010)
71. Kassab, M., Ormandjieva, O., Daneva, M.: An Ontology Based Approach to Non-functional Requirements Conceptualization. In: 4th Int. Conf. on Software Engineering Advances (2009)
72. Kassab, M., Ormandjieva, O., Daneva, M.: A Metamodel for Tracing Non-functional Requirements. In: World Congress on Computer Science and Info. Engineering (2009)
73. Sun, J., Zhao, L., Loucopoulos, P., Zhou, B.: QRA: A Quality Requirements Analysis Approach for Service Systems. In: IEEE Int. Conf. on Services Computing, Santa Clara, California, USA, pp. 25–32 (2013)
74. Sun, J., Loucopoulos, P., Zhao, L.: Representing and Elaborating Quality Requirements: The QRA Approach. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 446–453. Springer, Heidelberg (2013)
75. Tonu, S.A., Tahvildari, L.: Towards a framework to incorporate NFRs into UML models. In: IEEE WCRE Workshop on Reverse Engineering to Requirements, pp. 13–18 (2005)
76. Cysneiros, L.M., do Leite, J.C.S.P.: Nonfunctional requirements from elicitation to conceptual models. IEEE Transactions on Software Engineering 30(5), 328–350 (2004)
77. Rosa, N.S., Paulo, P.R.F., Justo, G.R.R.: Process NFL: A language for describing non-functional properties. In: 35th Annual Hawaii Int. Conf. on System Science (2002)
78. Zhu, L., Gorton, I.: UML profiles for design decisions and non-functional requirements. In: 2nd Workshop on Sharing and Reusing architectural Knowledge Architecture, Rationale, and Design Intent. 2007, Washington DC, USA, p. 8 (2007)
79. Pavlovski, C.J., Zou, J.: Non-Functional Requirements in Business Process Modeling. In: 5th Asia-Pacific Conf. on Conceptual Modelling (2008)
80. Kedad, Z., Loucopoulos, P.: Considering quality factors for business processes during requirement engineering. In: 5th Int. Conf. on Research Challenges in Info. Science (2011)
81. Heidari, F., Loucopoulos, P., Kedad, Z.: A Quality-Oriented Business Process Meta-Model. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 85–99. Springer, Heidelberg (2011)
82. Loucopoulos, P., Heidari, F.: Evaluating Quality of Business Processes. Modelling and Quality in Requirements Engineering, Essays Dedicated to Martin Glinz on the Occasion of His 60th Birthday, 61–73 (2012)
83. Heidari, F., Loucopoulos, P.: Quality evaluation framework (QEF): Modeling and evaluating quality of business processes. Int. Journal of Accounting Info. Systems (2013)
84. Heidari, F., Loucopoulos, P., Brazier, F.: Business Process Modelling for Measuring Quality. Int. Journal of Advances in Intelligent Systems 6(3&4), 342–355 (2013)
85. Botella, P., Burgues, X., Franch, X., Huerta, M., Salazar, G.: Modeling non-functional requirements. Jornadas de Ingenieria de Requisitos Aplicada (2001)
86. Kaiya, H., Ohnishi, A.: Quality Requirements Analysis Using Requirements Frames. In: 11th Int. Conf. on Quality Software (2011)
87. Kaiya, H., Tanigawa, M., Suzuki, S., Sato, T., Kaijiri, K.: Spectrum Analysis for Quality Requirements by Using a Term-Characteristics Map. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 546–560. Springer, Heidelberg (2009)
88. Cortesi, A., Logozzo, F.: Abstract interpretation-based verification of non-functional requirements. In: Jacquet, J.-M., Picco, G.P. (eds.) COORDINATION 2005. LNCS, vol. 3454, pp. 49–62. Springer, Heidelberg (2005)

89. Wena, X., Luoa, X., Ouyangb, J.: A Novel Evaluation Model for Non-functional Requirements in Trustworthy Software. Journal of Information & Computational Science 10(11), 3561–3577 (2013)
90. Kassab, M., Ormandjieva, O., Daneva, M., Abran, A.: Non-Functional Requirements Size Measurement Method (NFSM) with COSMIC-FFP. In: Cuadrado-Gallego, J.J., Braungarten, R., Dumke, R.R., Abran, A. (eds.) IWSM-Mensura 2007. LNCS, vol. 4895, pp. 168–182. Springer, Heidelberg (2008)
91. Kassab, M., Daneva, M., Ormandjieva, O.: Early quantitative assessment of non-functional requirements. In: University of Twente Report (2007)
92. Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezhanskaya, E., Christina, S.: Goal-centric traceability for managing non-functional requirements. In: 27th Int. Conf. on Software Engineering, pp. 362–371 (2005)
93. Sun, H., Basu, S., Lutz, R., Honavar, V.: Automata-based verification of non-functional requirements in web service composition, in Dept. of Computer Science Technical Report, Iowa State University (2009)
94. Drusinsky, D., Man-Tak, S.: Validating quality attribute requirements via execution-based model checking. In: 21st IEEE Int. Symposium on Rapid System Prototyping (2010)
95. Daneva, M., Kassab, M., Ponisio, M.L., Wieringa, R., Ormandjieva, O.: Exploiting a Goal-Decomposition Technique to Prioritize Non-functional Requirements (2007)
96. Liu, X.F.: A quantitative approach for assessing the priorities of software quality requirements. Journal of Systems and Software 42(2), 105–113 (1998)
97. Terawaki, Y.: Framework for Quantitatively Evaluating the Quality Requirements of Software System. In: Yamamoto, S. (ed.) HCI 2013, Part I. LNCS, vol. 8016, pp. 383–392. Springer, Heidelberg (2013)
98. Saito, Y., Monden, A., Matsumoto, K.: Evaluation of Non Functional Requirements in a Request for Proposal (RFP). In: 7th Int. Conf. on Software Process and Product Measurement (2012)
99. Zhang, H., Jarzabek, S., Yang, B.: Quality prediction and assessment for product lines. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 681–695. Springer, Heidelberg (2003)
100. Cortellessa, V., Singh, H., Cukic, B.: Early reliability assessment of UML based software models. In: 3rd Int. Workshop on Software and Performance, New York, USA, pp. 302–309 (2002)
101. Ghezzi, C., Sharifloo, A.M.: Model-based verification of quantitative non-functional properties for software product lines. Information and Software Technology 55(3), 508–524 (2013)
102. Supakkul, S., Hill, T., Chung, L., Leite, T.: An NFR pattern approach to dealing with NFRs. In: 18th IEEE Int. Requirements Engineering Conference 2010, pp. 179–188 (2010)
103. Nixon, B.A.: Management of performance requirements for information systems. IEEE Transactions on Software Engineering 26(12), 1122–1146 (2000)
104. Gregoriades, A., Sutcliffe, A.: Scenario-based assessment of nonfunctional requirements. IEEE Transactions on Software Engineering 31(5), 392–409 (2007)

Context-Based Variant Generation of Business Process Models

Ahmed Tealeb, Ahmed Awad, and Galal Galal-Edeen

Information Systems Department, Faculty of Computers and Information,
Cairo University, Egypt
ahtealeb@gmail.com, a.gaafar@fci-cu.edu.eg, galal@acm.org

Abstract. Nowadays, variability management of process models is a major challenge for Process-Aware Information Systems. Process model variants can be attributed to any of the following reasons: new technologies, governmental rules, organizational context or adoption of new standards. Current approaches to manage variants of process models address issues such as reducing the huge effort of modeling from scratch, preventing redundancy, and controlling inconsistency in process models. Although the effort to manage process model variants has been exerted, there are still limitations. Furthermore, existing approaches do not focus on variants that come from organizational or informational perspectives of process models. This paper introduces an approach to generate context-sensitive process model variants that come from adaptations in the organizational perspective. The approach is inspired by real life scenarios and has its conceptualization based on general concepts such as abstraction, and polymorphism.

Keywords: Process variants, Context-based algorithms, Abstraction, Polymorphism, Business Process Management (BPM), Process-Aware Information Systems (PAISs), Organizational perspective.

1 Introduction

The continuous need of organizations to manage their business processes has been the motivation for founding Process-Aware Information Systems (PAISs) applications such as Workflow Management Systems (WfMSs) and Business Process Management Systems (BPMSs). In recent years, the increasing adoption of PAISs has resulted in large process model repositories [1]. One of the ongoing research challenges in the PAISs area is variability management. Nowadays, variability management of process models is a major challenge of PAISs. One of the fundamental challenges of modeling business process is to deal with the multitude of variants that may exist for a particular process [2]. Each process variant constitutes an adjustment of a reference or basic process model to specific requirements. Efficient management for process model variants is a critical issue for organizations with the aim of helping them *reduce* the huge effort of modeling from scratch, prevent redundancy, and tackle inconsistency in process models.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 363–377, 2014.

Despite the effort done in current approaches e.g., in Provop [3], C-EPCs [4], and PPM [5] to manage process model variants, there are still limitations in each such as the difficulty to maintain the process model, and inconsistency between different variants of a process model. Furthermore, current approaches focus on dealing with variants coming from change in control and behavioral perspectives of process models. However, variants originating from organizational and informational perspectives still need to be studied, as we discuss in this paper.

The aim of this paper is to propose a context-based approach for generating variants of process models focusing on the organizational perspective. Organizational perspective is one of the different views integrated in the process model. It identifies the hierarchy of the organization who will execute the business process. We argue that changes in the organization structure will generate different variants of the process model. Our approach tries to generate process models that are close to the execution environment which contributes to the quality of the model and linking it to reality. Our approach is based on general concepts such as abstraction, and polymorphism applied to real life cases.

The rest of this paper is organized as follows: in section 2, we introduce basic concepts related to our approach. In section 3, we discuss two motivating scenarios. In section 4, we present our suggested context-based algorithms for managing variants of process models. In section 5, we discuss related work. Finally in section 6, we conclude with discussion of our approach and outline directions for future research.

2 Organizational Structures

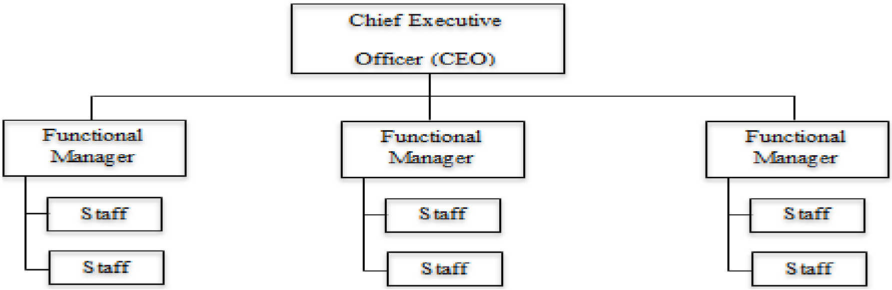
The structure of an organization may influence many processes or activities such as the availability of resources and the chain of approvals based on the hierarchy of roles. Based on the organizational structure, organizations are classified into three types: Functional, Projectized, and Matrix [6].

Functional Organization. In a functional organization, staff is grouped by specialty such as marketing, accounting, and engineering. Moreover, each employee has one clear supervisor, usually called "Functional Manager" as represented in Figure 1(a). Furthermore, each department does its work independently from other departments in a functional organization.

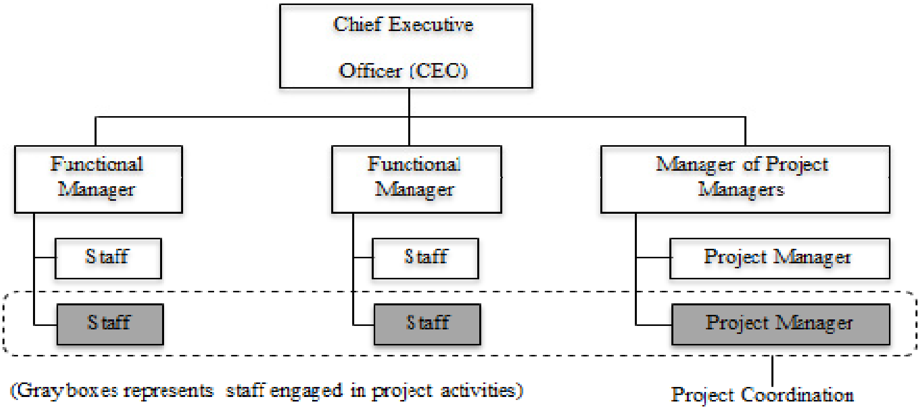
Projectized Organization. In a projectized organization, staffs members are assigned to projects. Moreover, each employee has one clear supervisor, often called "Project Manager" as represented in Figure 1(a), replacing the functional manager by the project manager. Furthermore, staff members from one or more departments can be shared as resources in a project work in a projectized organization.

Matrix Organization. Matrix organization is a mix of both functional and projectized organizations. Matrix organizations are classified as weak, balanced, or strong based on the level of power and influence between functional and project managers.

In weak matrix organizations, a functional manager has the power of authority on resources and a project manager performs as a coordinator with a lower level of authority. A balanced matrix organization does not provide the project manager with the full authority over the project and project funding. In strong matrix organizations, a project manager has the power of authority on resources like a functional manager as represented in Figure 1(b).



(a) Functional Organization



(b) Strong Matrix Organization

Fig. 1. Organization Structures (adapted from [6])

Many organizations are composed of all of the structures shown at various levels. An organization with such structure is called a “Composite Organization”. The variation in organization structure will affect the business process as we discuss in sections 3 and 4.

3 Motivating Scenarios and Requirements

In this section, we discuss two real life scenarios that motivated the development of our approach. We introduce two business processes and their variants as follows:

Vacation Leave Request and Course Registration in the sections 3.1 and 3.2 respectively. We are using a standard Business Process Model and Notation (BPMN) [7] to model the base and variant process models for each business process.

3.1 Vacation Leave Request

Vacation leave request is one of the most frequently executed business processes in each organization. The process of vacation request starts when an employee submits a vacation request. Once the leave request is registered, the supervisor of the employee receives the request; the supervisor either approves or rejects the request. If the request is rejected, the process sends a message with the rejection reasons. If the request is approved, the process sends the request to a final review by the Administrative Department and at the end the system will update the payroll system accordingly.

Figure 2 represents a *base model* for the process of "Vacation Leave Request". In the base model, there is one and only one manager who has the authority to *approve* or to *reject* the leave request in the organization. We referred to abstract role manager by the tagging notation <<Abstract>> before as in Figure 2.

The vacation leave request process may vary from one organization to another such as in Ministry of Interior (MOI) – State of Qatar. The source of variation stems from the *organizational structure* that differs from one organization to another (see section 2). We discuss how to obtain these variants in detail in section 4.1.

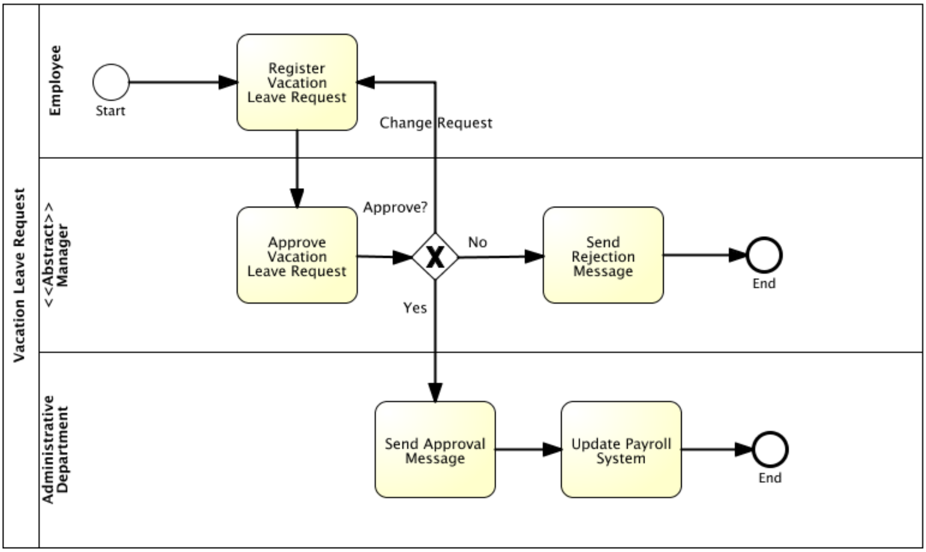


Fig. 2. Vacation Leave Request - Base Process Model

3.2 Course Registration

Course registration is a key business process operated in academic institutions. The process of course registration starts when the department of academic affairs offers the available courses for the semester. Then, students select some of the offered courses to register. The academic affairs officer must approve the student's courses. If the student's courses are approved it will go to the financial officer to calculate tuition fees. A student pays the calculated fees; then finally he/she will be registered for these courses in the selected semester.

Figure 3 represents a *base model* for the process of "Course Registration". In the base model, the "Student Courses" data object is an output from "Select Courses" activity and input for another activity called "Calculate Tuition Fees". Moreover, the activity called "Calculate Tuition Fees" which in this base model can be considered *Abstract* for any student in general either *Graduate* or *Undergraduate* Student(s). We referred to abstract task by the tagging notation <<Abstract>> before as in Figure 3.

The registration process may vary from a student to another such as in Qatar University. The variation stems from the student type that may be *Graduate* or *Undergraduate*. The activity of "*Calculate Tuition Fees*" will be implemented differently based on the *Student Type*. We discuss how to obtain these variants in detail in section 4.2.

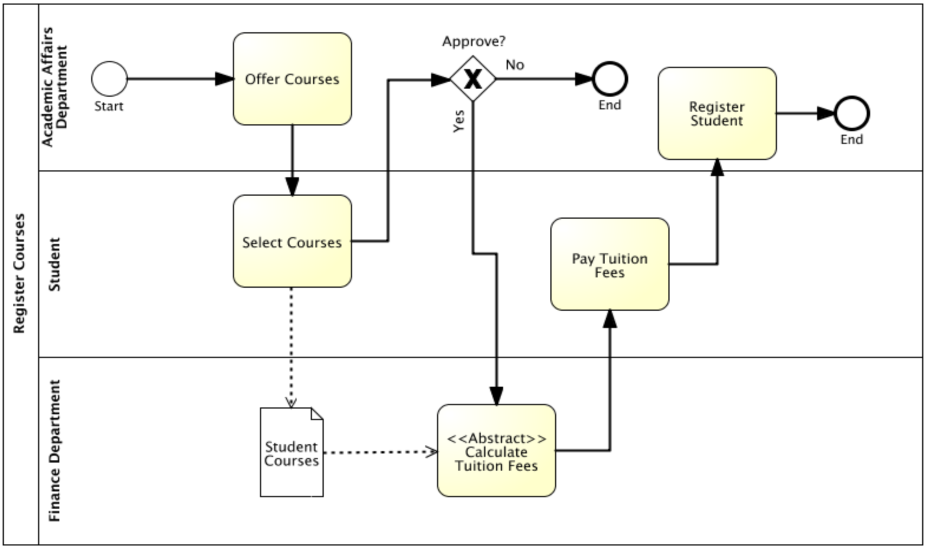


Fig. 3. Register Courses - Base Process Model

In the next section, we introduce two context-based algorithms for variants generation of business process models; "Organizational Structure", and "Polymorphism" algorithms.

4 Context-Based Variants Generation

In this section, we introduce two algorithms that represent a solution to manage variants of process models. The most common factor between these algorithms is its contextual basis on general concepts such as abstraction and polymorphism, and organizational structures mentioned in section 2. We present two context-based algorithms as follows: *Organizational Structure* and *Polymorphism* in the sections 4.1, and 4.2 respectively.

4.1 Organizational Structure-based Variant Generation

The "organizational structure" algorithm manages the variability of process models caused by different organizational charts such as shown for the "Vacation Leave Request" base model in the section 3.1 and its variants generated in this section. The "organizational structure" algorithm consists of the following:

Algorithm Parameters.
Parameter (1): [Base Model] – the base model of a process; e.g., "Vacation Leave Request - Base Process Model" in Figure 2.
Parameter (2): [Role] – the abstract role who implements a part of the base model, we call it a *fragment of the process*; e.g., In Figure 2, "Manager" is the abstract role who is authorized to approve or to reject the vacation leave request.
Parameter (3): [Organizational Structure] – the organizational chart as the driver of variation that will replace parameter (2) *i.e.,* the abstract role.

In the "Vacation Leave Request" in section 3.1, Organizational Structure at Ministry of Interior (MOI) – State of Qatar may be one of the following cases:

a) *Hierarchy or chain of managers list* such as the "Functional or Projectized Organization Process Model Variant" in Figure 4.

Figure 4 represents a variant of the base model of Figure 2 in case of the structure of the organization follows either *Functional Organization or Projectized Organization* (see section 2). The employee in this process model is managed by one manager, who is also managed by another, and so on. The top-down hierarchy structure of the MOI organization in Figure 4 is *Assistant Director -> Head of Section -> Direct In-Charge -> Employee*.

b) *No relationship between the managers list* such as the "Strong Matrix Organization Process Model Variant" in Figure 5.

Figure 5 represents another variant process model in case of the structure of the organization follows *Matrix Organization* (see section 2). The employee in this process model has two managers; Project Manager and Functional Manager. The hierarchy structure in this case follows the structure of Strong Matrix Organization as in Figure 5.

Algorithm Manipulation.

Activities or Tasks.

The *abstract role* executes a part of the process called *Fragment*. Fragments are repeated for each concrete level of the organizational chart: Each manager performs the same fragment as given in the base model; a fragment is represented in Figure 6.

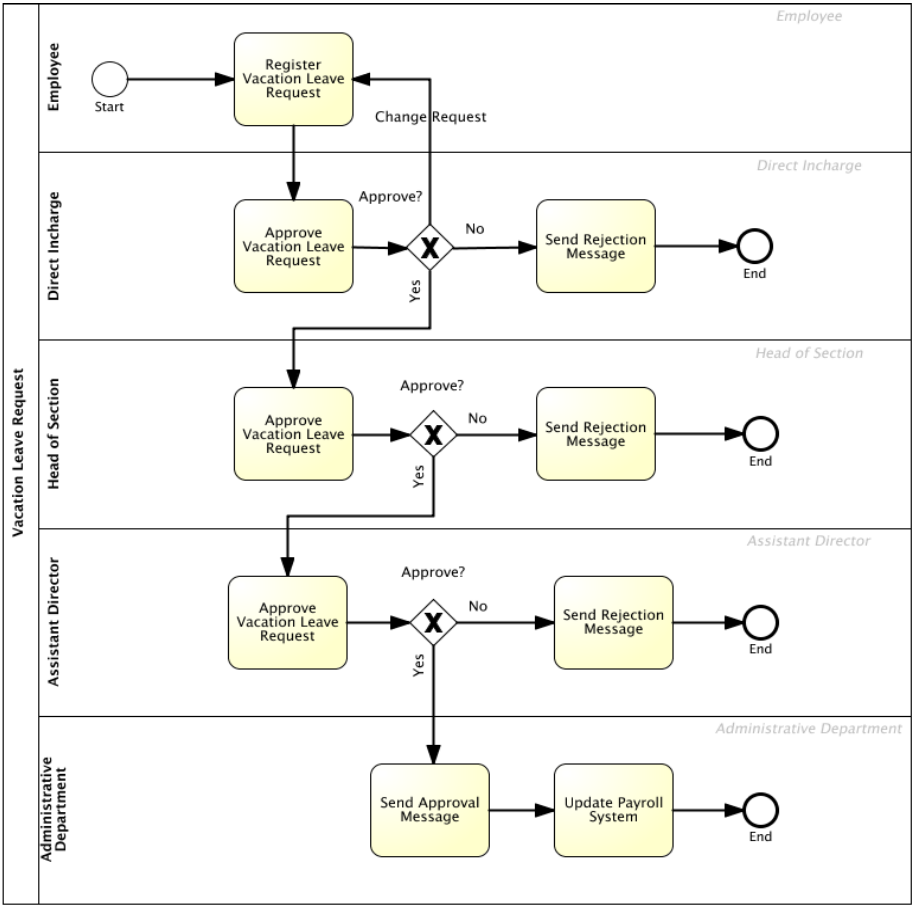


Fig. 4. Vacation Leave Request - Functional or Projectized Org. Process Model Variant

Control Flow Relationships among Fragments.

Case 1: Sequential fragments: Approval by managers is done sequentially as in "Functional Organization Process Model Variant" in Figure 4. The approval on vacation leave request requires a multilevel approval sequentially; the approval will be granted by the direct in-charge then by the head of section then by the assistant director. So, the approval in this case consists of three serialized approvals.

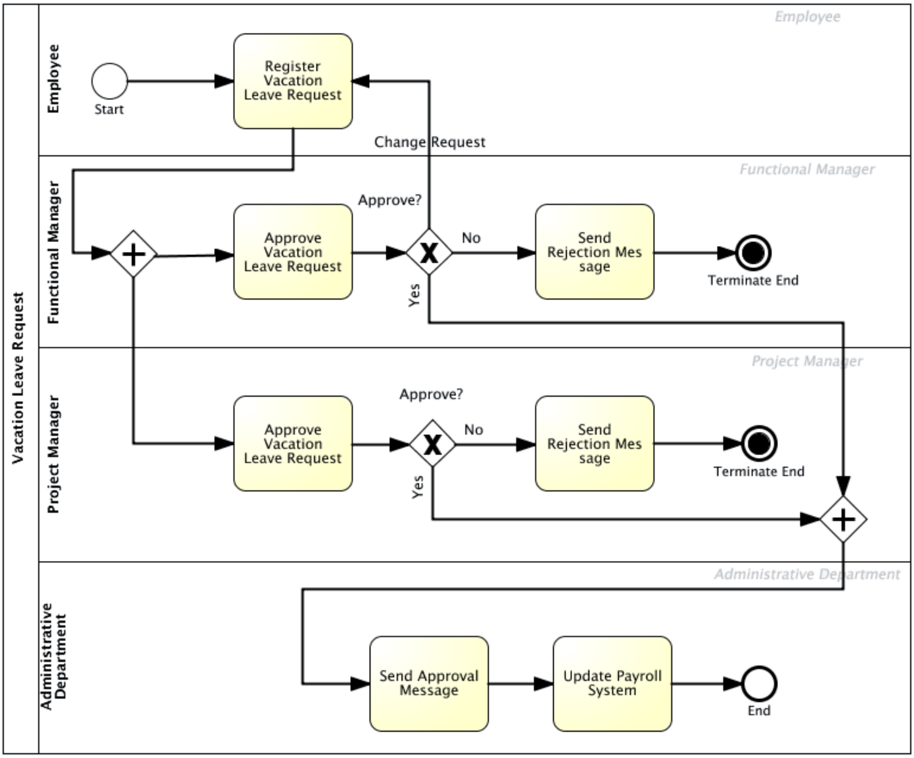


Fig. 5. Vacation Leave Request - Strong Matrix Organization Process Model Variant

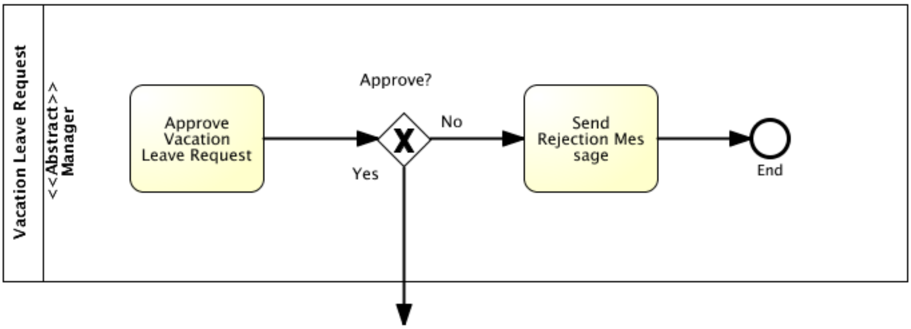


Fig. 6. Assigned Fragment for the entire Organization Chart

Case 2: Parallel fragments: Approval of managers is done in parallel as in "Strong Matrix Organization Process Model Variant" in Figure 5. The vacation request must be approved from both *Project Manager* and *Functional Manager* before sending it to the administrative department. Furthermore, if one or both of the mentioned above managers rejected the vacation request; End Event notation in the base model will be replaced by Terminate End Event notation. Terminate End Event notation will force the termination of the process in case of any manager rejects the vacation request. This replacement solves a deadlock problem which may be caused if the End Event notation is still existing. In other business processes, the semantics of approval may be different from in Figure 5. For example, in the process "Review a Paper", the approval or rejection of paper depends on the number of reviewers who approved it versus the number of reviewers who rejected. However, the semantics of approval is not our interest in this paper.

Algorithm 4.1 Organizational Structure-Driven Process Variant Generator.
Inputs: BM is the base model of a process, AR is the abstract role who implements a fragment of the process, OC is the organizational chart
Outputs: VPM variant process model(s)
Variables:
Fragment is part of the process the abstract role will execute.
CR is the concrete role(s) of the abstract role AR
VPM initially is the base model *without* the abstract role AR.

```
1 Fragment = AR.getFragment ()
2 For each CR in OC (AR)
3    VPM.AddRole (CR)
4    VPM.AssignFragmentToRole (Fragment, CR)
5       If OC.IsChain () then
6          VPM.LinkSequentialFragment ()
7       else
8          VPM.ReplaceEndToTerminate (Fragment)
9          VPM.LinkParallelFragment ()
10       end if
11 end for
```

Algorithm 4.1 is mainly designed according to the organizational structure algorithm discussed in section 4.1. Line 1 gets the fragment, which the abstract role executes in the base model. Line 2 reads the given organizational chart and gets the different concrete roles CR. Line 3 adds a new role in VPM with the name of CR returned from Organizational Chart OC. Line 4 assigns the fragment returned in Line 1 to the CR added in Line 3. Lines 5-10 is a condition related to the type of OC. In case, the organizational chart is a chain of managers, the method "LinkSequentialFragment" will be performed to link the fragments of concrete roles sequentially. Otherwise, the method "ReplaceEndToTerminate" will replace the End Event notation by Terminate Event notation. Then, the method "LinkParallelFragment" will be performed to link the fragments of concrete roles in a parallel way. Line 11 ends the read of the organizational chart after reaching the last CR.

So, we can conclude that the common source for *all variants* introduced for the process of "Vacation Leave Request" before is the *Organizational Chart*. "Organizational Structure" algorithm makes use of "Abstraction" as a general concept to implement the task differently based upon the variation of role. For example, "Approve Vacation Leave Request" task implemented differently based on the variation of "Organizational Chart" as in the process of "Vacation Leave Request" in section 3.1.

4.2 Polymorphism-Based Variant Generation

The "polymorphism" algorithm manages the variability of process models caused by different implementation for a task or an activity such as the "Course Registration" base model in the section 3.2 and its variants are generated in this section. The "polymorphism" algorithm consists of the following:

Algorithm Parameters.
Parameter (1): [Base Model] – the base model of a process; e.g., "Register Courses - Base Process Model" in Figure 3.
Parameter (2): [Abstract Role] – the abstract role who owns or affects the implementation of a set of abstract tasks; e.g., In Figure 3, "Student" as an abstract role who has abstract task "*Calculate Tuition Fees*".
Parameter (3): [Set of Abstract Tasks] – the set abstract tasks in the process will implement different logic based on role's type, such as the abstract method "*Calculate Tuition Fees*" in Figure 3.
Parameter (4): [Data Object] – provides information about how documents, data, and other objects are used and updated. Moreover, data object provides data and the concrete role for parameter (3) *i.e.,* the abstract role. e.g., "Student Courses" data object in Figure 3 provides both courses information and student concrete type "Graduate" or "Undergraduate" as a concrete role for the set of abstract tasks in parameter (3).

In the "Course Registration" in section 3.2, Student at Qatar University may be one the following:

a) *Graduate Student* such as the "Graduate Student Process Model Variant" in Figure 7.

Figure 7 represents a variant process model in case of the type of the student is *Graduate Student*. The abstract activity of "*Calculate Tuition Fees*" in Figure 3 became a concrete activity "*Calculate Tuition Fees for Graduate Student*" that will be implemented in a specific way for *Graduate Student(s).*

b) *Undergraduate Student* such as the "Undergraduate Student Process Model Variant" in Figure 8.

Figure 8 represents another variant process model in case of the type of the student is *Undergraduate Student*. The abstract activity of "*Calculate Tuition Fees*" in Figure 3 became a concrete activity "*Calculate Tuition Fees for Undergraduate Student*" that will be implemented in a specific way for *Undergraduate Student(s).*

Algorithm Manipulation.

Activities or Tasks.
Activity of the *abstract role* implemented differently for each sub role of abstract role: the "Calculate Tuition Fees" task read the student's type from "Student Courses" data object. The activity of "Calculate Tuition Fees" will be implemented differently for each child of "Student" abstract class.

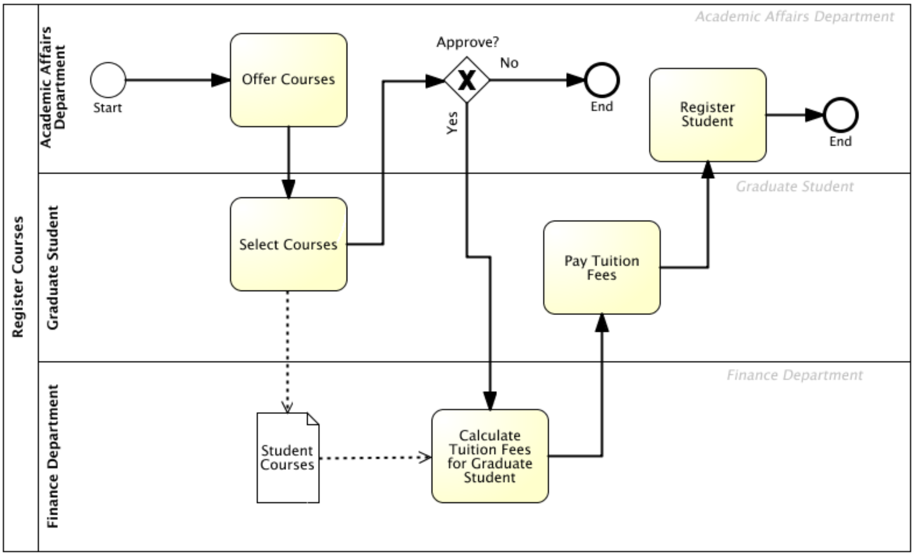


Fig. 7. Register Courses - Graduate Student Process Model Variant

Another manipulation for the case of "Course Registration" is given in section 3.2. We will have only one process model that handles all variants. We assume adding a "*Choice Block*" in Figure 3 after the data object "Student Courses". The choice block will evaluate the role type from the data object "Student Courses". If the role type is "Graduate Student", then the choice block directs to the method "*Calculate Tuition Fees for Graduate Student*". Otherwise, the choice block directs to the method "*Calculate Tuition Fees for Undergraduate Student*". However, this manipulation is not detailed in algorithm 4.2.

Algorithm 4.2 Polymorphism Process Variant Generator.
Inputs: BM is the base model of a process, AR is the abstract role who owns a set of abstract tasks, SAT is the set of abstract tasks in the process will implement different logic based on role's type, DO the data object that provides data and the concrete role for SAT
Outputs: VPM[] variant process model(s)

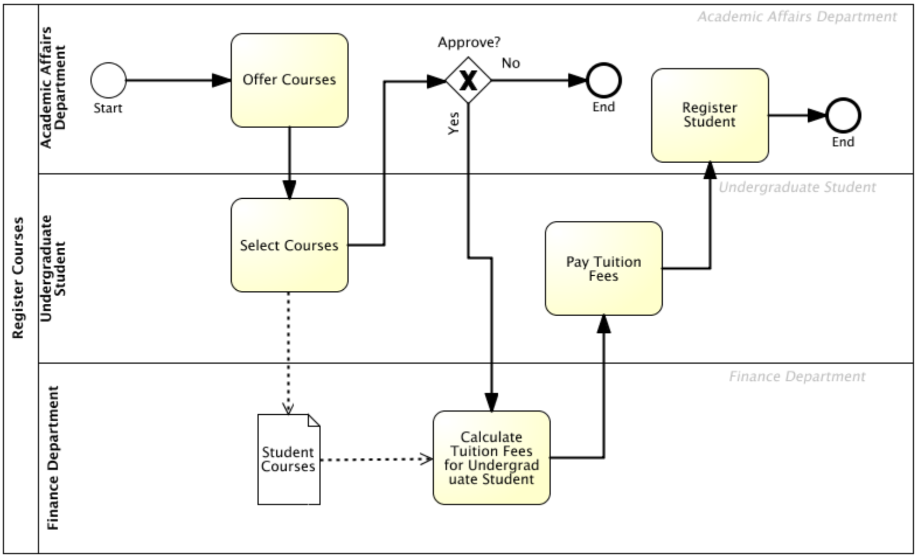


Fig. 8. Register Courses - Undergraduate Process Model Variant

Variables:
CT is concrete task(s) of the concrete role
AT is abstract task of the SAT
VPM initially is the base model

```
1 int i = 0;
2 For each CR in AR.concreteRoles ()
3   VPM[i] = BM
4   VPM[i].replaceRole (AR, CR)
5   For each AT in SAT
6     CT = getConcreteTaskforRole (CR, AT, DO)
7     VPM[i].replaceTask (AT, CT)
8   end for
9     i = i + 1
10 end for
11 return VPM[]
```

Algorithm 4.2 is mainly designed according to polymorphism algorithm discussed in section 4.2. Line 1 defines and initializes a counter for the array of VPM. Line 2 reads the given abstract role AR and gets the different concrete roles CR. Line 3 initialize the VPM to be like the base model BM. Line 4 replaces the abstract role AR by the concrete role CR. Lines 5-8 for each abstract task AT from SAT, algorithm retrieves the concrete tasks CT and replace the AT by CT in the generated VPM. Line 9 increases the counter for VPM. Line 10, reads concrete roles CR of abstract role AR is finished. Line 11, returns the generated VPMs.

So, we can conclude that the common source for *all variants* introduced for the process of "Course Registration" before is the Type of Student. The implementation of the activity called "*Calculate Tuition Fees*" will differ from process model to another based on the *Student's Type* (Graduate or Undergraduate). "Polymorphism" algorithm makes use of the concepts of "Polymorphism" and "Abstraction" to implement the task differently based on the variation of role. For example, "Calculation Tuition Fees" task implemented in different ways based on the variation of "Student's Type" as in the process of "Course Registration" in section 3.2.

5 Related Work

Several approaches have been developed in recent years to manage the different variants of process models, Such as PROcess Variants by OPtions (Provop), Configurable Event-driven Process Chains (C-EPCs), and Partial Process Models (PPM). In this section, we state the pros and cons for each approach.

Provop is an approach for managing a set of related process variants throughout the entire Business Process Life Cycle (BPLC) [8]. In Provop, a specific variant is derived by adjusting the basic process model using a set of well-defined change operations [3]. Change Operations represent the difference between basic model and variant such as INSERT, DELETE, and MOVE process fragments, and MODIFY process elements attributes. Furthermore, Provop supports the context-aware process configuration either statically or dynamically [9]. The methodology of Provop or the Provop lifecycle as illustrated in Figure 4 in [10] consists of three major phases: the modeling phase, the configuration phase and the execution phase. Provop has been extended with a procedure to guarantee the correctness and soundness for a *family of configurable process variants* [11, 12]. An extension has been developed for ARIS Business Architect to cope with variability in process models based on Provop [2]. Provop uses a bottom-up technique from process variants to the basic process model.

The concept of configurable process model has been defined by [4]. It deals with variants of process models by merging them into a single configurable model. According to [13], configurable process models are integrated representations for variants of a process model in a specific domain. A framework to manage the configuration of business process models consists of three parts: *a conceptual foundation for process model configuration, a questionnaire-based approach for validating modeling, and a meta-model for holistic process configuration* [13]. C-EPCs are configurable version of Event-driven Process Chains (EPCs), which provides a means to capture variability in EPC process models. C-EPCs identify a set of variation points which are called *configurable nodes* in the model and constraints, which are called *configuration requirements* to restrict the different combinations of allowed variants in order to be assigned for variants called *alternatives* [13]. C-EPCs uses a top-down technique from holistic or reference process model to process variants.

PPM is a query-based approach that depends on defining process models views to maintain consistency among process variants [14]. These views are defined using a visual query language for business process models called BPMN-Q [15]. Based on BPMN-Q, a framework for querying and reusing business process models has been developed by [16]. The methodology behind PPM is using inheritance mechanisms from software engineering to make best use of the reusability as a concept of Object-Oriented Modeling of object orientation [5]. PPM approach provides support for consistency of process model variants, and allows handling issues for multiple inheritance levels. PPM uses both top-down and bottom-up techniques in handling process variants. Context issues related to variants of business process are not covered in the PPM approach.

Despite the significant effort has gone into the current approaches; Provop [3], C-EPCs [4], and PPM [5] to manage process models variants, there are still a number of limitations in each. For Provop, each variant is defined and maintained through the base model only. Meanwhile, the changes in any process variant may not be consistent with other variants of the same process. For C-EPCs, specifying all variants in a holistic reference model for a particular process is difficult to maintain, since that reference model will be large process model. For PPM, the requirement of context configurations is not supported. Furthermore, current approaches are focused on dealing with variants that originate from change in control and behavioral perspectives of process models. However, organizational and informational perspectives still need to be studied. So, there is a necessity to develop an approach that discovers sources of variation based on process model perspectives *i.e.,* functional, behavioral, organizational, or informational, relationships between the different process model variants. Then, it should respond in an adaptive way to the characteristics of the case in hand.

6 Conclusions and Future Work

This paper introduced a conceptual approach for variability management of the process models. The approach helps practitioners, such as process owners and/or designers in designing and managing variations of their process models depending on the case in hand. The most significant finding behind the approach is the importance of general concepts such as *abstraction, polymorphism* which differs from one case to another based on the situation. Therefore, the characteristics of the case dominate in deciding the most suitable concept. In this paper, we presented two context-based algorithms to derive variants of process models. We applied the approach to real life process models to further illustrate our ideas.

In future work, we seek to find more algorithms and apply the approach for more real world cases in different domains such as IT, Healthcare, Education. Also, some of the generated variants from our approach may suffer from behavioral anomalies such as deadlocks. So, we will consider this for future research. Finally, a proof-of-concept prototype that validates the concept behind approach will be implemented.

References

1. Dijkman, R., La Rosa, M., Reijers, H.A.: Managing Large Collections of Business Process Models - Current Techniques and Challenges. Computers in Industry 63(2), 91–97 (2012)
2. Reichert, M., Rechtenbach, S., Hallerbach, A., Bauer, T.: Extending a Business Process Modeling Tool with Process Configuration Facilities: The Provop Demonstrator. In: Proceedings of Business Process Management Demonstration Track (BPMDemos 2009), Ulm, Germany. CEUR Workshop Proceedings series, p. 489 (2009)

3. Hallerbach, A., Bauer, T., Reichert, M.: Managing Process Variants in the Process Lifecycle. In: 10th Int. Conference on Enterprise Information Systems (ICEIS 2008), ISAS-2, Barcelona, Spain, pp. 154–161 (2008)
4. Rosemann, M., van der Aalst, W.M.P.: A Configurable Reference Modelling Language. J. of Information Systems 32(1), 1–23 (2007)
5. Pascalau, E., Awad, A., Sakr, S., Weske, M.: Partial Process Models to Manage Business Process Variants. Int. J. of Business Process Integration and Management 5(3), 240–256 (2011)
6. Project Management Institute (PMI).: A Guide to the Project Management Body of Knowledge: PMBOK® Guide. 5th edn., Project Management Institute Inc., Pennsylvania (2013)
7. OMG: Business Process Model and Notation 2.0 (BPMN 2.0) Specification, Final Adopted Specification. Technical Report, OMG (2011), http://www.omg.org/spec/BPMN/2.0/PDF
8. Hallerbach, A., Bauer, T., Reichert, M.: Managing Process Variants in the Process Life Cycle. Technical Report TR-CTIT-07-87, University of Twente (2007), http://dbis.eprints.uni-ulm.de/414/1/TB-CTIT-07-87%28Alena%29.pdf
9. Hallerbach, A., Bauer, T., Reichert, M.: Context-based Configuration of Process Variants. In: 3rd Int. Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008), Barcelona, Spain, pp. 31–40 (2008)
10. Hallerbach, A., Bauer, T., Reichert, M.: Capturing Variability in Business Process Models: The Provop Approach. J. of Software Maintenance and Evolution: Research and Practice 22(6-7), 519–546 (2010)
11. Hallerbach, A., Bauer, T., Reichert, M.: Guaranteeing Soundness of Configurable Process Variants in Provop. In: 11th IEEE Conference on Commerce and Enterprise Computing (CEC 2009), Vienna, Austria, pp. 98–105. IEEE Computer Society Press (2009)
12. Hallerbach, A., Bauer, T., Reichert, M.: Correct Configuration of Process Variants in
13. Provop. Technical Report UIB-2009-03, University of Ulm, Faculty of Electrical Engineering and Computer Science (2009), http://dbis.eprints.uni-ulm.de/491/1/TR_HBR_03_09.pdf
14. La Rosa, M.: Managing Variability in Process-Aware Information Systems. PhD Thesis, Queensland University of Technology, Brisbane, Australia (2009), http://eprints.qut.edu.au/20531/1/Marcello_La_Rosa_Thesis.pdf
15. Pascalau, E., Awad, A., Sakr, S., Weske, M.: On Maintaining Consistency of Process Model Variants. In: Muehlen, M.z., Su, J. (eds.) BPM 2010 Workshops. LNBIP, vol. 66, pp. 289–300. Springer, Heidelberg (2011)
16. Awad, A.: BPMN-Q: A Language to Query Business Processes. In: Proceedings of the 2nd Int. Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), St.Goar, Germany. LNI P-119, pp. 115–128 (2007)
17. Sakr, S., Awad, A.: A Framework for Querying Graph-Based Business Process Models. In: Proceedings of the 19th International World Wide Web Conference (WWW 2010), pp. 1297–1300. ACM, New York (2010)

Modeling Design-Time Variability in Business Processes: Existing Support and Deficiencies

Inbal Mechrez and Iris Reinhartz-Berger

Information Systems Department,
University of Haifa, Haifa 31905, Israel
inbalma@gmail.com, iris@is.haifa.ac.il

Abstract. Recently the interest in managing families of business processes rather than individual processes has increased, mainly due to the need to maintain different variants of the same business process or similar business processes in the same organization. This led to the extension of different business process modeling languages (BPMLs) in order to support the representation of design-time variability, namely variability that is resolved when designing the particular business processes (the variants). However, the evaluation of these languages expressiveness is still in an inceptive stage. In particular, the abilities to express variable elements in different granularity levels and to guide variability in business process models have not been examined. To tackle this lack, we propose a two-dimensional framework which explicitly refers to granularity and guidance. We further examine how existing extensions of BPMLs support these dimensions, point on deficiencies in their expressiveness, and discuss the implications of those deficiencies through examples from a case study.

Keywords: Variability Modeling, Design Time Variability, Business Process Modeling, Configuration.

1 Introduction

Business processes have drawn much attention over the years [4]. They affect organization's performance, cost, and customer's satisfaction and are considered one of the key concepts to successful businesses. A common way to present the specification of business processes is through business process models which capture different aspects of business processes, such as their goals and constraints, their activities and flow, their events and resources, and the different organizational units or roles involved in their execution.

Various graphical languages have been proposed over the years to model business processes. These languages aim to bridge the gap between business process design and implementation, as well as to represent and communicate different aspects of business processes to various stakeholders. This is done in different ways: *imperative* business process modeling languages (BPMLs), for example, focus on *how* the process is executed (mainly, its activities and flow), while *declarative* BPMLs support the description of *what* should be done and not how it is done [25]. *Graph-based*

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 378–392, 2014.

BPMLs visually specify business processes as graphs, while *rule-based* BPMLs support abstracting the process logic into a set of rules [22]. There are BPMLs that mainly focus on *input/output flows*; others focus on *workflows* (namely, time ordering of activities); a third group concentrates on *agent* cooperation; and a fourth group is considered *state-based* [1].

Usually a single organization deals with a large number of business processes. The different business processes are not necessarily far apart from each other; they may be variants which are commonly considered as specializations of "abstract" business use cases [31]. Sometimes the existence of such variants indicates on large differences in the instances of the business process. This kind of variability is commonly referred to as *runtime variability* [39]. In other cases the need to handle variable aspects is raised during the process design phase and requires designing and managing variants of the same business process or business process part for different organizational units, market segments, or involved items. In this case, which is commonly referred to as *design time variability* [39], variability is resolved at design-time and not at run-time, potentially making the variants more suitable to the specific business process needs, but less flexible.

Another interest in design-time variability is raised by software companies that aim to develop COTS products or process-aware information systems (PAIS) [8] for a market segment that includes organizations that have similar core business processes. These software companies may benefit from treating the different business processes as a family, monitoring and analyzing process commonality and variability. The results of such analysis can be incorporated into the products development, yielding flexible products that can be adapted to meet the specific needs of a particular organization in that market segment.

In order to support design-time variability in business processes, several modeling languages have been suggested in the last decade. Most of these languages extend existing languages, and especially BPMN and EPC, with variability aids, e.g., [26, 28, 31], or suggest aids to specify the variability orthogonally to the business process models, e.g., [13, 40]. However, the evaluation of these languages expressiveness is still in an inceptive stage. In this paper, we propose a two-dimensional framework that refers to *granularity*, namely, the variable elements, and *guidance*, i.e., the creation of variants at design-time. We use this framework for evaluating the expressiveness of 22 languages that support design-time variability modeling in business processes.

The contribution of this paper is two folded. First, it provides a useful input for practitioners by pointing and discussing the deficiencies of the different languages and assisting in language selection. Second, the deficiencies are also of interest to researchers who wish to know what languages should be worked on and in what directions.

The rest of the paper is structured as follows. Section 2 reviews the relevant literature. Section 3 elaborates on the suggested framework. Section 4 reviews existing BPMLs that support design-time variability modeling and discusses their expressiveness based on the suggested framework. Section 5 presents and exemplifies the found deficiencies, as well as discusses their implications. Finally, Section 6 concludes and refers to future research.

2 Related Work

A few studies have already examined variability modeling in business processes. Torres et al. [38] present two approaches targeted at the representation of process families: the *behavioral approach* which derives a process variant by hiding and blocking elements, and the *structural approach* which applies a set of change operations on a base model in order to derive a process variant. The authors further compare two specific languages: C-EPC [28], which is a behavioral approach, and Provop [13], which is a structural approach. The comparison is done in terms of understandability of the produced process model.

Ayora et al. [2] propose an approach that refers to variability concepts as first order elements in business process models. This approach is evaluated with respect to C-EPC [28], PESOA [26], and Provop [13], using a set of criteria. Some of these criteria refer to variability concepts, including variation points, process fragments, process fragment context, process fragment relationships, language support regarding variability, process context regarding variability, and variation point resolution time. The other criteria define quality factors, such as flexibility, scalability, and understanding.

Vervuurt [40] defines nine criteria that need to be considered when evaluating business process variability modeling languages, including: (1) the ability to mark variable elements, (2) the support of change patterns, (3) the configuration rules that adapt process model, (4) visualization of configuration rules that adapt process models, (5) domain visualization and process model configuration, (6) domain and process configuration rules, (7) selective display, (8) correctness, and (9) consistency. All these criteria focus on configuration as the main mechanism for creating variants. Vervuurt further uses the nine criteria for comparing and evaluating specific modeling languages, namely: C-EPC, BPMN, and Extended EPC (E-EPC). Based on the comparison findings, alternative solutions to business process variability modeling problems are suggested: combining C-EPC with feature diagrams (Feature-EPC), extending C-EPC with Change-Oriented Versioning (COV-EPC), and utilizing Proteus Configuration Language (PCL-EPC).

La Rose et al. [17] review three approaches to capture variability in business process models: (1) configurable nodes, e.g., C-EPC, (2) hiding & blocking, which aim to represent choices in configurable process models independently of the language, e.g., Configuration in SAP WebFlow [10], and (3) annotation-based process variability which aim to “improve the customization of process-oriented software systems”, e.g., the study in [35]. They further claim that the existing languages do not provide sufficient support during the actual configuration of the generic or configurable process model, commonly termed the *reference model*. Thus they suggest independent representations of the variability that can be used to complement these approaches: questionnaires models, feature diagrams, and adaptive mechanisms.

Weidmann et al. [41] specifically refer to the variability scope, but only within BPMN 2.0, concluding that events, activities, gateways, sequence & message flows, and pools & lanes can have variable attributes. They further compare four approaches to variability modeling in business processes, namely Provop, PESOA, Process Configuration (ProCon) and Multi-Perspectives Variants (MultPers). The comparison, which is based on five criteria that focus on variability and dependency visualization in process models, leads to the conclusion that the interaction of the user with the process model is missing. Thus, an approach for Adaptive Business process modeling in the Internet of Services (ABIS) is presented. This approach enables business users to create their own process variants using process templates and process fragments.

Discussing techniques that deal with the management of process model variants, Dijkman et al. [7] distinguish between techniques that use a single consolidated model to capture the process variants and techniques that keep the process variants separate. The first group of techniques mainly utilizes variation points to distinguish between the common and variable parts. Representing variability in this kind of techniques may rely on configuring nodes (e.g., [11], [13], [18], [31]), attaching parameters to nodes or marking nodes with stereotypes (e.g., [35]), assigning cardinalities to arcs and nodes (e.g., [29]), or using aspect-oriented principles (e.g., [21]). The second group of techniques "leaves the various variants separate, but provides an infrastructure to identify and keep track of their commonalities in order to maintain consistency across variants when updating them" [7]. This can be done, for example, by utilizing the inheritance mechanism, using version control techniques, or identifying behavioural relations between process variants.

The above studies examine and compare a few extensions of BPMLs that support design-time variability modeling. However, these studies treat the business process models as a whole, without separately referring to different business process elements and to the way they vary. As variability may be present in different granularity levels, it is important to know the variability of which elements is supported by a certain BPML. In addition, the support that BPMLs provide to (re)use variable elements in specific business processes is not sufficiently analyzed in those studies, which mainly concentrate on a single mechanism – configuration, and do not examine the relationships between the process elements and the utilized mechanisms.

3 The Suggested Evaluation Framework

To tackle the aforementioned limitations, our framework refers to two dimensions for evaluating design-time variability in business process models: *granularity*, which refers to the variable business process elements, and *guidance*, which refers to the mechanisms to create variable elements (i.e., variants).

3.1 Granularity Dimension

Curtis et al. [6] refer to four perspectives of business processes: functional, behavioral, organizational, and informational. These perspectives are also mentioned in List and Korherr's metamodel [21], which was inspired by ARIS [34]. These perspectives, which are briefly reviewed below, are of high relevance to many BPMLs, which are classified as imperative. Thus, we set them and their high-level elements as the values of the granularity dimension.

The ***functional perspective*** represents *what* process elements are being performed [6]. The main elements in this perspective are atomic tasks and composite activities. ***Atomic tasks*** (also termed process elements or process steps) are functional units of a process that have no externally visible substructure. ***Composite activities***, on the other hand, represent major units of work that need to be performed in order to achieve the objective of a process. Composite activities are commonly described as sets of partially ordered tasks.

The ***behavioral perspective*** represents *when* activities are performed as well as aspects of *how* they are performed [6]. In particular, the behavioral perspective describes the order in which the different activities are executed (***control flows***) and when process elements are performed (i.e., ***sequence flows***). Moreover, ***data flows*** are used to connect atomic tasks with information resources (such as data, artifacts, and products) [16].

The ***organizational perspective*** describes the organization structure and, in particular, *where* and *by whom* (which agents) process elements are performed. Three types of process participants are commonly mentioned [21]: (1) an ***organizational unit***, which is a group of people organized for some purpose; (2) a ***role***, which is a group of process elements exhibiting a set of specific skills or qualifications and assigned to an agent; and (3) ***software***, e.g., applications and services, which automatically performs process elements.

The ***informational perspective*** represents the information and data produced or manipulated by a process and their interrelationships [6]. The informational perspective describes *which* information is involved in the business process, *how* it is represented, and *how* it is propagated among different activities. The elements of the informational perspective are primarily divided into resources and events: an ***event*** may trigger an activity or a task, whereas a ***resource*** is an entity to be produced or consumed by an atomic task, e.g., data, products, and artifacts.

Table 1 summarizes the granularity dimension in terms of perspectives, relevant questions, and high-level elements.

Table 1. The granularity dimension

Perspective	Relevant Questions	High-Level Elements
Functional	- *What* process elements are being performed?	- Atomic tasks - Composite activities
Behavioral	- *When* are process elements performed? - *How* are process elements performed?	- Sequence flows - Control flows - Data flows
Organizational	- *What* is the organization structure? - *Where* and *by whom* are the process elements performed?	- Organization units - Roles - Software
Informational	- Which information is involved? - How is it represented? - How is it propagated among different process elements?	- Resources - Events

Note, however, that there are a few business process elements, mentioned in the literature, which cannot be naturally classified into one of the four perspectives. These elements are either more abstract than the elements in the four perspectives (e.g., goals, soft-goals, and domains), or are needed for evaluation or classification purposes (e.g., process types, measures, and dimensions). List and Korherr [21] classify these elements under a fifth perspective called business process context. Although these elements are important for the completeness of the process models, their number and nature made us currently leave them out of the framework scope.

3.2 Guidance Dimension

The second dimension refers to the ways variability of a business process family can be resolved in order to create specific business processes. These ways are commonly termed *variability mechanisms* or *reuse mechanisms* in the area of reference modeling and business process families [5, 35]. Table 2 lists four common mechanisms, which were established as the values of the guidance dimension, their descriptions, and related terms.

Table 2. The guidance dimension

Variability mechanism	Related terms	Description
Configuration	Inclusion, exclusion, selection, blocking, hiding, deletion	Enables selecting process elements for inclusion
Inheritance	Specialization, encapsulation, uses	Enables specializing process elements
Parameterization	Parameters, values	Enables customizing process elements by assigning values to parameters
Extension	Addition, insertion	Enables attaching several variants (process elements) at a certain point at the same time

Configuration and parameterization are classified in [35] as basic variability mechanisms, as they are standalone and do not require any other variability mechanisms or new model design. Inheritance and extension, on the other hand, are variability mechanisms derived by restriction. Nevertheless, all the four are common variability mechanisms in business process modeling.

4 BPMLs That Support Design-Time Variability Modeling

Searching for BPMLs that have been suggested to model design-time variability in business processes, we found 22 such languages published since 2005. All of them are graph-based languages and most of them are imperative. 7 languages are based on BPMN and 6 on EPC. A few languages are based on other BPMLs: YAWL (2), UML Activity Diagrams (AD) (2), UML State machines (1), EWF-nets (1), Petri-nets (1), goal models (1), and SAP WebFlow (1). Most of the languages (20 out of 22) extend the base notation and introduce a single (unique) model that captures both commonality and variability. This kind of languages is commonly called *annotation-based* as variability is annotated on the base model. Two languages distinguish and keep the base model separate from the variability model. This kind of languages is termed *composition* as it proposes ways to combine or compose the two separately handled models, the base and the variability models.

Table 3 details, for each extension of BPML, the variability of which perspectives and high level process elements is supported and how, namely with which variability mechanisms. The grayed rows are languages that follow a composition-based approach (all others are annotation-based languages). As can be seen, variability is not uniformly supported with respect to the granularity and guidance dimensions: there are neglected perspectives, neglected elements, neglected variability mechanisms, and neglected combinations. We next discuss these deficiencies and exemplify their implications with examples from a case study.

5 Deficiencies in Business Process Variability Modeling

Conducting the evaluation of the reviewed extensions of BPMLs according to the granularity and guidance dimensions, we can find several deficiencies with respect to design-time variability modeling in business processes. To examine whether these deficiencies indicate real limitations, we conducted a case study for examining the variability of procurement processes in two organizations: a university library and an industrial company dealing with defense electronics. We collected data on the procurement processes and their variability through interviews, observations, and existing documents. We qualitatively analyzed the data and classified each variability type according to the two suggested dimensions. Due to space limitations, we briefly discuss here each deficiency and exemplify the implications with examples from the case study.

5.1 Deficiencies with Respect to the Granularity Dimension

Neglected Perspectives. Business processes may differ in what they are doing (the functional perspective), how and when they are doing that (the behavioral perspective), where and by whom they are doing that (the organizational perspective), and which information is required and in what way (the informational perspective). The business processes in a certain organization may vary only in specific perspectives and not in all of them. For example, procurement processes may vary in when and how they are performed and not in what they are doing, where and by whom they are doing that, and which information is required and in what way. In this case the expressiveness of variability modeling in the behavioral perspective is important, requiring a BPML whose expressiveness in this category is high.

As can be seen in Table 3, the most handled perspectives are the functional and the behavioral ones. These perspectives are the most prominent in "regular" BPMLs and here we see that they remain prominent when dealing with design-time variability. Variability modeling in the informational perspective, on the other hand, is supported to some extent. Several studies refer to variability in different data-related resources, such as data storage, objects, inputs, and outputs. Variability of events is partially handled in four studies. A possible reason for this low support may be that events, as opposed to functional and behavioral units and data-related elements, are considered external and independent of the organization (i.e., often the organization cannot directly affect the events of its business processes) [28]. Thus, events are individually handled and their variability is not commonly modeled.

Table 3. Design-time variability modeling in existing extensions of BPML*

Category	Study	Functional Perspective		Behavioral Perspective			Organizational Perspective			Info. Perspective	
		atomic tasks	composite activities	control flows	sequence flows	data flows	org. units	roles	software	events	resources
BPMN-based	PESOA for BPMN [26]		c e i p		p	p					p
	DeCo [33]	c i						c i			c i
	ADOM-BPMN [29]	c i		c i		c i					c i
	BPFM [23]	c									
	AO4BPMN 2.0 [24]		c i							c i	
	ABIS [41]	c	c	c	c	c					
	Provop [13]	c e i	c e i	c e		c e					
EPC-based	C-EPC [28], [31]	c		c							
	PCL-EPC [40]	c		c	c					c	
	ADOM-EPC [30]	c i		c i						c i	
	c-iEPC [18]			c				c i			c i
	COV-EPC [40]	c e		c						c e	
	Feature-EPC [40]	c		c							
Others	C-EWF [11] (EWF nets)	c		c							
	Liaskon et al. [20] (goal models)		c	c			c				c
	Protos2CPN [9] (Petri nets)	c									c
	Conf. in SAP WebFlow [10]			c							
	PESOA for UML-AD [26]	c e i p		p		p					c p
	Razavian & Khosravi [27] (UML-AD)	c		c		c					
	PESOA for state machines [26]			p	p						c e i p
	C-YAWL [12] (YAWL)			c							
	CoSeNet [36] (YAWL)			c	c						
# studies supporting the variability type		14	5	17	5	6	1	2	0	5	7

* Support of: c – configuration, i – inheritance, e – extension, p – parameterization

Variability modeling in the organizational perspective is partially handled in only three studies, where variability of roles or organizational units is handled. The reason for this low support may be that the reviewed studies selected to extend languages that do not focus on the organizational perspective, such as EPC and BPMN, and did not adopt more holistic methods, such as ARIS [34].

As an example for the need to improve the expressiveness in the neglected perspectives, Table 4 lists three examples taken from our case study. In the first two examples the variability is in the organization perspective: the budget can be controlled by a role or by a software system, and received items can be inspected by different roles. This type of variability cannot be handled in existing BPMLs as configurable control flows connect functional elements and not organizational ones and inheritance cannot simply utilized when different elements types (e.g., a role and software in the first example) are involved. In the third example, the variability is in the informational perspective. One can claim that this kind of variability can be specified using parameterization. However, currently business rules need to be associated to the parameter in order to constrain the values it can receive at design-time (and not at run-time).

Neglected Elements. Analyzing the variability in the different perspectives, we observed that not all elements in the same perspective are similarly handled. The degree of support for the different elements is once again important as organizations may face variability in certain elements, e.g., business processes that involve many events and event handlers. In this case, using BPMLs that support variability in the informational perspective will not necessarily help, as those BPML may concentrate on resource variability (and not on events).

Table 4. Examples of variability related to neglected perspectives

Case title	Case description	Organization	Comments
Budget control	Budget can be controlled by the finance department that monitors and alerts on excess expenditures or by an automatic alert software system	Industrial company	Variability in the organizational perspective
Received Items Inspection	Received item inspection can be done by a warehouseman or by any worker qualified by the warehouseman	Industrial company	Variability in the organizational perspective
Delivery date overdue	The system warns on delivery date overdue; the number of "acceptable" overdue days varies, depending on the organization policies	Industrial company, university library	Variability in the informational perspective

As can be seen in Table 3, the most neglected perspective naturally also yields the most neglected elements. However, in the functional and behavioral perspectives, the variability of composite activities, sequence flows and data flows is neglected. We speculate that the reasons for this lack of support are that composite activities are perceived as aids to support scalability and thus their variability is not supported as much as the variability of the building blocks (the atomic tasks); sequence flows are mainly used to connect elements and their variability is hard to be grasped and modeled; and data flows are secondary elements in process models. Furthermore, variability in data flows may be percolated to variability in resources (i.e., in the informational perspective) and vice versa. In the informational perspective, events and resources are similarly neglected. Finally, in the organizational perspective, the variability of software elements is completely neglected, maybe since they are considered in business process modeling as “black boxes”.

Table 5 lists three examples related to neglected elements found in our case study: variability in software, sequence flows, and resources.

Table 5. Examples of variability related to neglected elements

Case title	Case description	Organization	Comments
Purchase order generation	A purchase order can be generated automatically by a purchasing module of an ERP system or automatically by an autonomous purchasing system	Industrial company, University library	Variability in software (org. perspective)
Shipment order	Purchase order can be produced before shipment or after shipment (push supply)	University library	Variability in sequence flows (behavioral perspective)
Types of invoices	A supplier invoices can be hard-copy or electronic	Industrial company	Variability in resources (inf. perspective)

The above types of variability can be handled in existing BPMLs by utilizing inheritance, but such a treatment introduces abstract elements to the model – the “super” elements in the inheritance, which complicate the models and may negatively affect comprehension. Furthermore, inheritance of behavioral elements is not well supported in existing BPMLs and sometimes requires splitting models or percolating the variability to connectors.

Neglecting Cross-perspective or Cross-element Variability. Most BPMLs support variability within the same kind of elements. Only a few BPMLs refer to variability that goes beyond the boundaries of a single element type or perspective. These BPMLs commonly define placeholder elements that can be replaced by different elements from the same perspective or from different perspectives. This possibility is mainly utilized for replacing control flows and sequence flows, atomic tasks and composite activities, and atomic tasks and sequence flows. The other combinations are (almost) completely neglected. As an example to the need to represent variability of different elements, which potentially belong to different perspectives, consider a case of inventory assessment. Assessing the inventory may be a complicated function in a certain organization, justifying its representation as a composite activity that includes tasks for counting the actual amounts, writing them down, comparing them to the expected amounts, resolving differences, and so on. The same process may be very simple in an organization which checks its inventory continuously by means of a cycle count, requiring only generation of a printed report. Moreover, inventory assessment in one organization may be an internal function, calling for its representation in the functional perspective. A different organization may use JIT (Just In Time) method in which the supplier manages the inventory and supplies the products whenever they are needed. In this case the supplier is an external entity that triggers events that may cause the activation of different functional units when occur. In these cases it is important whether the BPML supports variability that goes beyond the boundaries of a specific high-level element or a specific perspective.

5.2 Deficiencies with Respect to the Guidance Dimension

Neglected Variability Mechanisms. The nature of the mechanisms makes them suitable to different types of variability. Configuration, for example, is suitable to situations where all the variants are explicitly modeled and the selection of the appropriate variant needs to be guided. Extension and inheritance, on the other hand, enable additional design of the variants. Finally, parameterization requires generalization of the variants and properly using parameters when necessary.

As can be seen in Table 3, variability in business processes is mainly supported in terms of configuration. All the 22 reviewed BPMLs support configuration, which is relatively easy to utilize. Furthermore, in 12 of the languages configuration is the only utilized mechanism. To support configuration, the languages usually supply means for specifying optional elements and selection conditions. Less than half of the reviewed BPMLs support inheritance, while parameterization and extension are far away neglected. A possible reason for this may be that parameterization requires extra generalization effort (done only in the PESOA project) and extension is too lenient and less guided. The case entitled "delivery date overdue" in Table 4 exemplifies the need for parameterization at design-time, while the case entitled "received items inspection" in that table exemplifies the need for extension.

Neglected Granularity-Guidance Combinations. Examining the granularity-guidance combinations, we found that the most commonly used mechanism in all perspectives is configuration, while inheritance is commonly used, in addition to configuration, in the functional and informational perspectives. Extension and parameterization are lowly used in the functional, behavioral, and informational perspectives. These findings may be attributed to the nature of the mechanisms: inheritance of tasks and resources is known in other modeling areas, such as object-oriented modeling; extension is found with respect to functionality (e.g., extension points in use case diagrams); and parameterization is mainly known with respect to data and information. We found evidence to the need of the different variability mechanisms in the various perspectives. The case entitled "purchase order generation" in Table 5, for instance, exemplifies the need for inheritance in the organizational perspective.

6 Conclusions and Future Directions

Analyzing variability of business processes is important within an organization and between similar organizations. The main way to present the outcome of such analysis is through variability models which can be incorporated into or presented orthogonally to the business process models. We examined the expressiveness of different BPMLs that support design-time variability modeling with respect to two dimensions: *granularity*, which refers to four perspectives and their elements, and *guidance*, including four variability mechanisms. We found that variability in the functional and behavioral perspectives is extensively handled in the modeling level, although variability of composite activities, sequence flows and data flows is quite neglected. Variability in the informational perspective is supported to some extent, leaving aside important elements, such as events. Variability modeling in the organizational perspective is far away neglected. We further found that configuration is the most utilized variability mechanism in business processes, but some languages support extension, inheritance, and parameterization for creating process variants mainly in the functional and information perspectives. All languages concentrate on variability within the same element kinds, neglecting possible variability between different types of elements that may belong to the same or different perspectives.

It is important to consider the current study under the following limitations. First, we reviewed modeling languages in the field of business processes. We could extend the scope of review to studies that deal with variability in databases and organizations. This way we could increase the expressiveness in the informational and organizational perspectives. However, incorporating such languages into BPMLs is not trivial and may increase complexity (potentially decreasing comprehension). Second, most modeling languages reviewed in the current study are workflow-oriented. This is because most "regular" BPMLs are workflow-oriented [19]. However, business process modeling approaches that capture and refine business goals also exist. The study in [20], which is included in our review, is a goal-oriented language that explicitly refers to design-time variability in business processes. Third, we included in our study only graphical languages that extend existing BPML. In particular, textual and formal languages as well as proprietary languages were not included.

In the future, we plan to provide concrete suggestions for improving the expressiveness of variability modeling in BPMLs. In particular, we will provide suggestions for supporting neglected perspectives, elements, mechanisms, and combinations. We further plan to empirically evaluate the influence of these suggestions on the usability of different BPML extensions for variability modeling and their comprehensibility. Finally, we intend to explore additional dimensions and refine the current dimensions, e.g., by examining additional variability mechanisms and referring to low-level process elements.

References

1. Andersson, B., Bider, I., Johannesson, P., Perjons, E.: Towards a formal definition of goal-oriented business process patterns. Business Process Management Journal 11(6), 650–662 (2005)
2. Ayora, C., Torres, V., Pelechano, V.: Variability management in business process models. Technical Report 17, PROS – UPV (2012)
3. Bayer, J., Flege, O., Gacek, C.: Creating product line architectures. In: van der Linden, F.J. (ed.) IW-SAPF 2000. LNCS, vol. 1951, pp. 210–216. Springer, Heidelberg (2000)
4. 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)

5. Brocke, J.: Design Principles for Reference Modelling: Reusing Information Models by Means of Aggregation, Specialisation, Instantiation, and Analogy. Reference Modeling for Business Systems Analysis, 47–75 (2007)
6. Curtis, B., Kellner, M., Over, J.: Process Modeling. Communication of the ACM 35(9), 75–90 (1992)
7. Dijkman, R., La Rosa, M., Reijers, H.: Managing large collections of business process models — current techniques and challenges. Computers in Industry 63(2), 91–97 (2012)
8. Dumas, M., Van der Aalst, W.M.P., Ter Hofstede, A.H.M.: Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons (2005)
9. Gottschalk, F., van der Aalst, W.M.P., Verbeek, H.M.W.: Protos2cpn: Using colored Petri Nets for configuring and testing business processes. International Journal on Software Tools for Technology Transfer 10(1), 95–110 (2007)
10. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: SAP WebFlow Made Configurable: Unifying Workflow Templates into a Configurable Model. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 262–270. Springer, Heidelberg (2007)
11. Gottschalk, F., Van Der Aalst, W.M.P., Jansen-Vullers, M.H., Rosa, M.L.: Configurable workflow models. International Journal on Cooperative Information Systems 17(2), 177–221 (2008)
12. 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)
13. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: The Provop approach. Journal of Software Maintenance and Evolution: Research and Practice 22(6/7), 519–546 (2010)
14. Istoan, P., Klein, J., Perouin, G., Jézéquel., J.M.: A metamodel-based classification of variability modeling approaches. In: VARY International Workshop Affiliated with MODELS 2011, pp. 23–32 (2011)
15. Korherr, B.: Business Process Modelling Languages, Goals, and Variabilities. PhD Thesis, Vienna University of Technology (2008)
16. Korherr, B., List, B.: Extending the EPC and the BPMN with Business Process Goals and Performance Measures. In: The 9th International Conference on Enterprise Information Systems (ICEIS 2007), pp. 287–294. ACM/SIGMIS and AAAI (2007)
17. La Rosa, M., Dumas, M., ter Hofstede, A.: Modelling Business Process Variability for Design-Time Configuration. In: Cardoso, J., van der Aalst, W.M.P. (eds.) Handbook of Research on Business Process Modeling. Information Science Reference, pp. 204–228 (2008)
18. La Rosa, M., Dumas, M., ter Hofstede, A., Mendling, J.: Configurable Multi-Perspective Business Process Models. Information Systems 36(2), 313–340 (2011)
19. 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)
20. Liaskon, S., Lapouchnian, A., Yu, Y., Yu, E., Mylopoulos, J.: On Goal-based Variability Acquisition and Analysis. In: The 14th IEEE International Conference Requirements Engineering (RE 2006), pp. 92–96 (2006)

21. List, B., Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages. In: The 21st ACM Symposium on Applied Computing (SAC 2006), pp. 1532–1539 (2006)
22. 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)
23. Park, J.: A Modeling Approach for Business Processes Based on Variability. In: 9th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 211–218 (2011)
24. Patiniotakis, I., Papageorgiou, N., Verginadis, Y., Apostolou, D., Mentzas, G.: An Aspect Oriented Approach for Implementing Situational Driven Adaptation of BPMN2.0 Workflows. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 414–425. Springer, Heidelberg (2013)
25. Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A.: Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 383–394. Springer, Heidelberg (2012)
26. Puhlmann, F., Schnieders, A., Weiland, J., Weske, M.: Variability Mechanisms for Process Models. PESOA-Report TR 17/2005, Process Family Engineering in Service-Oriented Applications, PESOA (2005)
27. Razavian, M., Khosravi, R.: Modeling Variability in Business Process Models using UML. In: 5th International Conference on Information Technology: New Generations (ITNG 2008), pp. 82–87 (2008)
28. Recker, J., Rosemann, M., van der Aalst, W.M.P., Mendling, J.: On the Syntax of Reference Model Configuration. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 497–511. Springer, Heidelberg (2006)
29. Reinhartz-Berger, I., Soffer, P., Sturm, A.: Organizational reference models: Supporting an adequate design of local business processes. International Journal of Business Process Integration and Management 4(2), 134–149 (2009)
30. Reinhartz-Berger, I., Soffer, P., Sturm, A.: Extending the adaptability of reference models. IEEE Transactions on Systems, Man, and Cybernetics, Part A 40(5), 1045–1056 (2010)
31. Rosemann, M., van der Aalst, W.: A Configurable Reference Modelling Language. Information Systems 32(1), 1–23 (2007)
32. Russell, N., ter Hofstede, A.H.M., Edmond, D.: Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane (2004), http://is.tm.tue.nl/research/patterns/documentation.htm
33. Rychkova, I., Nurcan, S.: Towards Adaptability and Control for Knowledge-Intensive Business Processes: Declarative Configurable Process Specifications. In: The 44th Annual Hawaii International Conference on System Sciences, pp. 1–10 (2011)
34. Scheer, A.W.: ARIS – Business Process Modeling, 3rd edn. Springer (2000)
35. Schnieders, A., Puhlmann, F.: Variability Mechanisms in E-Business Process Families. In: The 9th International Conference on Business Information Systems (BIS 2006), pp. 583–601 (2006)
36. Schunselaar, D.M.M., Verbeek, E., van der Aalst, W.M.P., Raijers, H.A.: Creating Sound and Reversible Configurable Process Models Using CoSeNets. In: Abramowicz, W., Kriksciuniene, D., Sakalauskas, V. (eds.) BIS 2012. LNBIP, vol. 117, pp. 24–35. Springer, Heidelberg (2012)

37. Shahzad, K., Johannesson, P.: An Evaluation of Process Warehousing Approaches for Business Process Analysis. In: The 5th ACM-EOMAS in Conjunction with CAiSE 2009, pp. 1–14 (2009)
38. Torres, V., Zugal, S., Weber, B., Reichert, M., Ayora, C., Pelechano, V.: A qualitative comparison of approaches supporting business process variability. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 560–572. Springer, Heidelberg (2013)
39. Van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA 2001), pp. 45–54 (2001)
40. Vervuurt, M.: Modeling Business Process Variability. Master Thesis. University of Twent (2007)
41. Weidmann, M., Koetter, F., Kintz, M., Schleicher, D., Mietzner, R.: Adaptive Business Process Modeling in the Internet of Services (ABIS). In: ICIW 2011, pp. 29–34 (2011)

Linguistic Consistency of Goal Models

Fabian Pittke[1], Benjamin Nagel[2], Gregor Engels[2], and Jan Mendling[1]

[1] Wirtschaftsuniversität Wien
Augasse 2-6, A-1090 Vienna, Austria
fabian.pittke@wu.ac.at, jan.mendling@wu.ac.at
[2] s-lab- Software Quality Lab, University of Paderborn
Zukunftsmeile 1, 33102 Paderborn, Germany
bnagel@s-lab.upb.de, engels@s-lab.upb.de

Abstract. Goal models are used for the elicitation and specification of strategic requirements in early phases of the software engineering lifecycle. By explicitly modeling requirements on a strategic level, these goals provide input for the derivation of operational software specifications. An unambiguous and consistent definition of the goals is the prerequisite for this derivation. Addressing this challenge, this paper presents an analytic approach for the automatic detection of linguistic inconsistencies in goal models. By providing syntactical and semantic consistency conditions, we support requirements engineers by improving the overall quality of goal-oriented requirements specifications. To demonstrate the applicability of our approach, we apply it to three case studies taken from literature using the implemented tool support.

Keywords: Goal models, inconsistency detection, linguistic consistency.

1 Introduction

Goal-oriented requirements engineering has emerged as a paradigm for the elicitation and specification of strategic requirements in early phases of the software lifecycle. Goal models enable the iterative definition of goals on different levels of abstraction and their decomposition to concrete operations that need to be performed to achieve the stated goals. The elicitated goals and operations are used as foundation for the following phases of the software engineering process. Hence, quality and consistency of the goal models have a direct impact on future development stages.

The definition of goals involves several stakeholders with different roles and backgrounds which typically leads to uncontrolled and inconsistent goal descriptions. This is further complicated by the use of natural language to describe goals. To mitigate this problem, goal model notations provide means to formally specify goals in terms of state definitions expressed e.g. in linear temporal logic (LTL). Yet, many stakeholders feel unfamiliar with formal notations [8] that current goal models typically include. Instead, they use natural language or semi-formal visualization techniques for the communication, discussion, and the derivation of system specification artifacts, e.g. business process models [19,18] or UML

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 393–407, 2014.

diagrams [32]. Since these techniques are prone to language errors and inconsistencies, there is a strong need to assure the consistency of natural language in goal models.

In order to ensure the consistency of goal models with regard to natural language, one has to consider two different aspects. First, goal model elements are connected by hierarchical decomposition relations that represent different levels of abstraction. Consistency with respect to these relations implies that one goal is an appropriate decomposition of its super goal. Missing guidelines and the high-level of flexibility for the specification of goal models increase the threat of gaps in terms of inconsistent terminology between goals in decomposition relations. Second, the natural language itself has to be considered as it introduces inconsistencies, such as overloaded and synonymous concepts in goal descriptions. In consequence, ambiguous concepts arise and complicate the following communication activities between business stakeholder and requirements engineers [17].

Addressing these research challenges, we contribute an analytic approach to validate the consistency of the natural language fragments that are used goal model elements. A syntactical check evaluates the consistency of goals following the decomposition relations among them. To ensure the unambiguity of goal models, homonyms and synonyms are identified by a semantic consistency check. Moreover, we further analyze the decomposition logic and provide means to ensure the correctness of the decomposition. To evaluate the capabilities of our approach, we challenge it against three goal models from literature differing in size, domain, and the use of natural language.

The remainder of this paper is structured as follows. Section 2 introduces the foundations for our approach and discusses related work on goal model consistency. Our approach on consistency validation of goal models is presented in Section 3. In order to demonstrate the applicability and validity of our approach, Section 4 gives an overview of the performed evaluation including three case studies taken from literature. Finally, Section 5 concludes this paper and provides an outline of future work.

2 Problem Statement and Related Work

In this section, we further elaborate the problem at hand and discuss related work. Thus, Section 2.1 states the problem based on an example and related work on goal model consistency is described and analyzed in Section 2.2.

2.1 The Problem of Linguistic Inconsistencies

In prior research, different aspects of goal model consistency have been addressed. In particular, formal definitions of goal models and related properties can be verified using approaches that rely on temporal logic and other reasoning techniques. For example, these techniques make use of formal goal model definitions to handle conflicts [34] and obstacles [35]. In this way, these techniques ensure that formalized goal models are consistent. However, these techniques require a very detailed description of goal models, e.g., resources or actors, and are not capable to ensure the consistency of natural language or semi-formal goal modeling techniques.

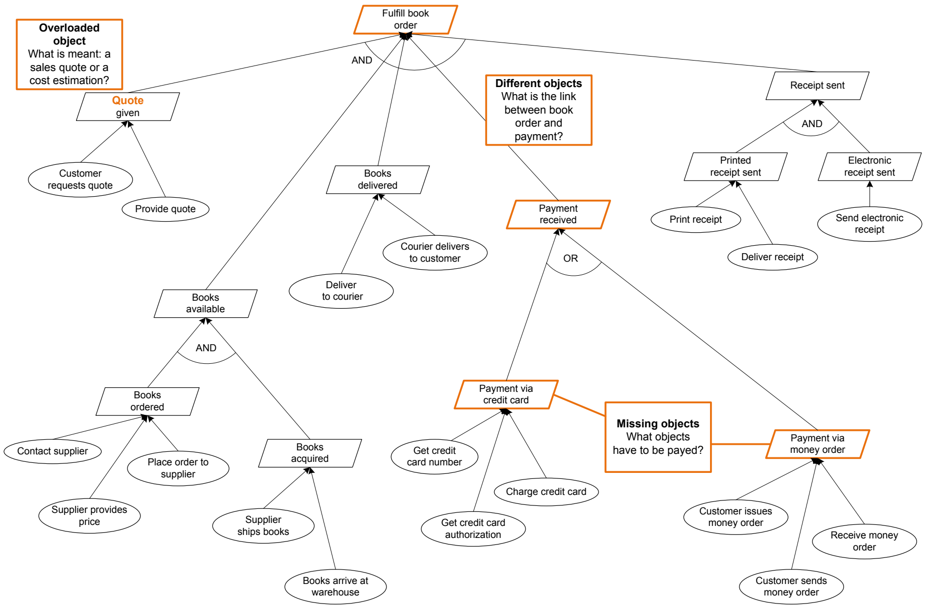


Fig. 1. *Fulfill book order* Goal Model [22]

In order to illustrate this problem, we provide an example of such a specification from the literature [22]. We used the KAOS goal modeling notation [9,10] for the visualization as it provides language-support for the elicitation and specification of goals and requirements in early phases of the software lifecycle. Figure 1 depicts the goals that have to be fulfilled with regard to a book order. To do so, four subgoals have to be fulfilled that are further decomposed. For example, *Books available* is decomposed to *Books ordered* and *Books acquired*. Moreover, goals can also be achieved by alternative subgoals. For example, the goal *Payment received* is fulfilled by either achieving the goal *Payment via credit card* or *Payment via money order*. By performing the iterative decomposition, goals are getting more concrete until they can be operationalized and executed. For example, the goal *Printed receipt sent* is achieved by performing the operations *Print receipt* and *Deliver receipt*.

Figure 1 also shows inconsistencies that are grounded in the use of natural language. We observe the first inconsistency formed by the goals *Payment via credit card* and *Payment via money order*. Both goals specify that a payment needs to be done. However, they lack information about the specific object that has to be payed. This information, i.e. the book order, is only given implicitly by the context. As a second inconsistency, we observe that the goal *Payment received* and the top goal *Fulfill book order* deal with completely different objects. While the latter one deals with the receipt of a payment, the other goal is concerned with the fulfillment of a book order. Possibly, using the label *Book order payment received* is more concise. The third inconsistency is shown in the goal *Quote given* and its operations. In this case, the word *quote* might relate to a sales quote or a cost estimation. As the meaning is not obvious from the context, the word quote is a perfect example of a homonym. Thus, in order to ensure the consistency of goal models with regard their natural language labels, we require model verification and validation techniques.

2.2 Related Work on Goal Model Consistency

The consistency of goal models is an integral part of software engineering as it assures the correctness of the future software artifact [2]. For that purpose, verification and validation techniques have been developed [5]. While *verification* addresses the general properties and rules of a conceptual model, *validation* addresses the consistency of the model with respect to the universe of discourse. Verification can be achieved by algorithmic analyses on the conceptual models, while validation also requires the consultation and discussion of conceptual models by stakeholders.

There are several approaches to support the validity and consistency assurance of goal models. The works by van Lamsweerde [34,35] present techniques for conflict management, especially with regard to divergence detection and resolution, and obstacle handling. These approaches provide value for validating the consistency of KAOS goal models with respect to the formal definitions, e.g. a CTL definition of a desired state on a given context model. In [12], Fuxman et al. propose a model-checking approach for the analysis of i*/Tropos goal models. It provides an extended notation that is used to derive a set of LTL constraints. Using these constraints different consistency checks can be performed. In contrast to our approach, this technique focuses on the verification properties instead of ensuring consistency of the definition of the individual goals. The extended Tropos notation presented in [14] addresses the specific aspect of security requirements. By using a formalized representation of security and trust requirements their fulfillment is verified in an automated manner. Consistency with respect to privacy and security constraints is also addressed in [7,6]. By applying a planning approach, it constructively derives different design alternatives. A comprehensive overview of goal model consistency techniques is given in the survey by Horkoff and Yu [16].

As the related work in the field of goal model consistency shows, there is a considerable amount of approaches that verifies the formal definition of goals with temporal logic and other reasoning approaches after the goals have been aligned with business stakeholders. However, it is not sufficient to solely rely on formal approaches, as goal models also contain natural language text elements that are prone to errors and that contribute to the overall consistency of goal models. The potential of analyzing the natural language fragments in goal models for consistency checking has not been taken into account so far. Therefore, we define several consistency conditions that ensure the syntactic and the semantic correctness of natural language goal models.

3 Conceptual Approach

In this section, we propose several conditions to check the consistency of goal models based on linguistic aspects. First, Section 3.1 defines the preliminaries for our approach before the Sections 3.2 and 3.3 introduce the linguistic consistency conditions of goal models.

3.1 Preliminaries

In general, we can consider a goal model as a set of goals G, a set of operations O, as well as hierarchic relations $D \subseteq (G \cup O) \times (G \cup O)$ among these goals. We have already outlined that these hierarchy relations can be expressed as OR-decompositions or AND-decompositions respectively. Based on the work of Antón [4], goals are expressed in terms of actions as well as objects on which these actions are applied. In many cases, the actions contain a temporal constraint that implies a timely sequence when the goal has to be fulfilled. Considering the goal *printed receipt sent* of our example in Figure 1, we can identify the action *to send* as well as the object *printed receipt.* Furthermore, we also find a temporal constraint expressed in the past participle of the action. Accordingly, this goal is fulfilled after both operations have been performed, i.e. the receipt has been printed and delivered. Similarly, Prat [28] utilizes functional grammars to formalize goals consisting of a verbal predicates and several dependable arguments, such as the agent, the recipient, and the object given to the recipient. Analogously, the goal *printed receipt sent* consists of the verbal predicate *to send* and the object *printed receipt.* Based on these insights, we consider a goal as being labeled with a verb g^a that describes a distinct action and a noun g^o that describes a distinct object. To extract these components, we can use available techniques, such as [20], to automatically annotate goals with these components as they follow similar labeling styles as process models [24].

Further, we define the following conventions to formulate the linguistic consistency conditions:

Definition 1. *Given a goal model $\mathcal{G}$ and a goal $g' \in G$ that is decomposed into several goals or operations. Then, we define:*

- *The set of all subordinate goals of g': $SG^{g'} = \{g \in G \mid (g', g) \in D\}$.*
- *The set of all subordinate operations of g': $SO^{g'} = \{o \in O \mid (g', o) \in D\}$.*
- *The set of all elements that are part of a decomposition of g':* $DC^{g'} = SG^{g'} \bigcup SO^{g'}$.

3.2 Syntactic Consistency Conditions

This section will introduce consistency conditions that check a goal model from the syntactical dimension of natural language. In this context, we check if a goal or operation contains all necessary components or if a goal decomposition is consistent based on the object for which the goal has been decomposed.

Component Consistency. As outlined by Antón [4], a goal specifies an action as well as an object on which the action is performed. Accordingly, we can check the decomposition of a goal with respect to the completeness of the components, i.e. if all goals actually specify an action and an object. The rationale is that a goal that does not contain all components is underspecified and may cause confusion when communicating the goals in software development. For example, consider the operation *Deliver to courier* from the example. Obviously, the operation properly specifies an action but misses an object that has to be delivered. To check goal models for this inconsistency, we define the component consistency of a goal docompositon as follows:

Definition 2. *(Component Consistency). Let $DC^{g'}$ be a goal decomposition of a goal $g' \in G$. $DC^{g'}$ is consistent with respect to the components iff* $\forall g \in DC^{g'} : \; g^a \neq \emptyset \wedge g^o \neq \emptyset$

Component Stringency. One essential characteristic of goal modeling is the step-wise decomposition of goals [31,3,4]. The decomposition process involves the creation of a logical subgrouping of goals until an operational definition is possible. While decomposing goals, one has to ensure that decomposed goals are still related to the super goal. In the example, the goal *fulfill book order* is decomposed into four goals. Among them, the sub goal *payment received* requires the receipt of a payment before the book order is considered as fulfilled. However, we observe that the object *book order* in the top goal is replaced by the object *payment* and thus lacks information about the original object to be paid. Thus, we define that the decomposition is not stringent with respect to the components. Accordingly, we define the component stringency:

Definition 3. *(Component Stringency). Let $DC^{g'}$ be a goal decomposition of a goal $g' \in G$. $DC^{g'}$ is stringent with respect to the components iff* $\forall g \in DC^{g'} : \; g'^o = g^o$.[1]

3.3 Semantic Consistency Checking

In this section, we introduce conditions that analyze the semantics of a goal model in detail. Hence, we consider aspects like the ambiguity of goals or the consistency of their decomposition.

Homonym and Synonym Consistency. Goal models are used to close the communication gap of domain experts and system analysts [8]. As they are

[1] Note that it is not meaningful to check the component stringency for actions since a specific object needs to be processed requiring different types of actions.

created by several domain experts and stakeholders, the unambiguity of the specified concepts has to be preserved. However, this unambiguity often suffers from linguistic ambiguities such as homonymy (a word has more than one meaning) or synonymy (different words have the same meaning) [11]. As a synonym example, we consider the goal *Books ordered* and the operation *Place order to supplier* in the illustrative example. We see that the actions *to order* and *to place* clearly intend to lodge the book order to the selected supplier. For the homonym case, we could interpret the object *book order* in the top goal *fulfill book order* as an order for reading books on the one hand or as reservation order on the other hand. Based on the definitions of Deissenboeck [11], we can formalize the semantic consistency conditions for synonyms and homonyms as follows:

Definition 4. *(Synonym Consistency). Let* $Senses_D$ *denote a function that retrieves all word senses of a given word from a dictionary D. Further, let* $DC^{g'}$ *be a goal decomposition of a goal* $g' \in G$. $DC^{g'}$ *is consistent with respect to synonym usage iff* $\forall g_1, g_2 \in DC^{g'}$:
$((Senses_D(g_1^a) \cap Senses_D(g_2^a) = \emptyset) \wedge (Senses_D((g_1^o) \cap Senses_D(g_2^o) = \emptyset))$

Definition 5. *(Homonym Consistency). Let* $Senses_D$ *denote a function that retrieves all word senses of a given word from a dictionary D. Further, let* $DC^{g'}$ *be a goal decomposition of a goal* $g' \in G$. $DC^{g'}$ *is consistent with respect to homonym usage iff* $\forall g \in DC^{g'} : (|Senses_D(g^a)| = 1 \wedge |Senses_D(g^o)| = 1)$.

Goal-Logic Consistency. The literature shows that there is a considerable amount of approaches that exploit the underlying logical structure of goal models by using formal predicates and first order logic [16]. We take a complementary direction and assess the logical consistency based on semantic relations between the goals and their components. In particular, we can analyze the semantic closeness of the sub goals with respect to the decomposition type of the super goal. As example, consider the sub goals *payment via credit card* and *payment via money order* of the OR-decomposed super goal *payment received*. As the payment via credit card is conceptually exclusive to the payment via money order, the semantic closeness of these goals is expected to be small. Accordingly, we require for the goals of an OR-decomposition that the semantic closeness is smaller than a given threshold τ_{or}. In contrast to that, the AND-decomposition constitutes that several sub goals need to be achieved in order to fulfill the super goal and thus, that they are part of the super goal. Thus, we do not expect parts of a greater whole to be strongly connected with each other rather than moderately related. For example, the operations *Get credit card number* and *Get credit card authorization* are part of the super goal *Payment via credit card*. Although the objects *credit card number* and *credit card authorization* are not notably close to each other, we still require a moderate connection between these since they are part of the object *credit card*. Therefore, we require the goals of an AND-decomposition to fall within a specified range of thresholds τ_{and}^{min} and τ_{and}^{max}. Finally, to measure the semantic closeness of the goal components, several semantic measures can be used, such as the measure by Lesk [21], Wu and Palmer

Table 1. Details of the Test Goal Models

Characteristic	**BOF** [22]	**CR** [27]	**SES** [30]
No. of Goals	12	16	21
No. of Operations	18	13	0
No. of Decompositions	12	13	6
No. of Unique Actions	18	14	5
No. of Unique Objects	14	17	18

[36], Resnik [29], or Lin [23]. Accordingly, we can formalize the logically-semantic consistency as follows:

Definition 6. *(AND-Consistency). Let sim be a function that calculated the closeness of two goals, τ_{and}^{min} and τ_{and}^{max} be semantic closeness thresholds for an AND-decomposition of goals. Further, let $DC^{g'}$ be a goal decomposition of a goal g'. $DC^{g'}$ is consistent with respect to the AND-logic iff*

$$\tau_{and}^{min} \leq \sum_{g_1, g_2 \in DC^{g'}} \frac{sim(g_1^a, g_2^a) + sim(g_1^o, g_2^o)}{|DC^{g'}|} \leq \tau_{and}^{max}$$

Definition 7. *(OR-Consistency). Let sim be a function that calculated the closeness of two goals and τ_{or} be a semantic closeness threshold for an OR-decomposition of goals. Further, let $DC^{g'}$ be a goal decomposition of a goal g'. $DC^{g'}$ is consistent with respect to the OR-logic iff*

$$\sum_{g_1, g_2 \in DC^{g'}} \frac{sim(g_1^a, g_2^a) + sim(g_1^o, g_2^o)}{|DC^{g'}|} \leq \tau_{or}$$

4 Evaluation

In order to demonstrate the capabilities of the presented linguistic consistency conditions, we challenge them against real-world goal models. More specifically, we employ these conditions to three goal models from literature varying in size and domain. Thus, we aim to learn whether our proposed consistency conditions are capable of finding linguistic inconsistencies in these models. Accordingly, we begin with a short description of the evaluation setup (Section 4.1) and present the results for each goal model (Section 4.2) afterwards. Finally, we discuss the results in Section 4.3.

4.1 Setup

For our evaluation scenario, we reviewed the literature for available goal models and selected three goal models with varying size, domain and expected quality of goal labeling. Table 1 summarizes the main characteristics of these goal models. Our data set includes:

- *Book order fulfillment* (BFO): The goal model presented by Liaskos et al. [22] was already introduced in Section 2.1 and specifies all necessary goals to fulfill a book order. The goal model encompasses 12 goals that are all decomposed into sub goals and operations. For the latter, the goal model inhibits 18 operations. In general, we can distinguish 18 unique actions and 14 unique objects which indicates a certain consistency and standardization of specifying goals.
- *Cleaning Robot* (CR): The scenario depicted in [27] specifies goals and requirements a cleaning robot has to satisfy when cleaning a room from dust and other objects. It comprises 16 goals and 13 operations and has thus a similar size as the book order fulfillment. Again, all 16 goals are decomposed into smaller sub goals and operations. This goal model includes 14 unique actions and 17 unique business objects.
- *Safe Elevator System* (SES): The safe elevator system scenario [30] is one of the special goal models we selected for the evaluation. In contrast to the other scenarios, it specifies 21 goals and, interestingly, no operations. Moreover, there are only six goal decompositions. The goal model itself contains 18 distinct objects, but only five distinct actions. We selected this goal model as it mainly specifies the elevator system by using its characteristic objects.

In order to evaluate the proposed linguistic consistency checks, we prototypically implemented them using Java 1.7. As the goal models were selected from literature, we transformed them into a processable input format that captures all the information from the graphical representation, i.e., the goals and operations along with their label, action, and object, as well as the decomposition structure, and type. In order to look up the different word senses for the semantic consistency conditions, we employ the lexical database WordNet version 3.0 [25,26] which captures the word senses as sets of synonym words (synsets). For the goal logic consistency, we set the thresholds τ_{or} to 0.5 and the thresholds for τ_{and}^{min} and τ_{and}^{max} to 0.3 and 0.5 respectively. We choose for these values as empirical research guidelines also argues that values of 0.5 or higher indicate a (moderate) relation between two concepts, while values between 0.3 and 0.5 indicate a weak relation [15]. Among the semantic similarity metrics, we selected the Lin measure as it correlates best with human judgment and thus is also capable to quantify weak and moderate relations [23].

4.2 Linguistic Consistency Results

The quantitative and qualitative consistency results for all of the three goal models are summarized in Table 2 and Table 3. From the numbers of the *compound consistency* conditions, we learn that all three goal models show these inconsistencies. Furthermore, we can see that the BOF model performs best according to this criterion since 6 out of 12 goal decompositions are specified with an action and an object. As an example, we list the goal *Books ordered* and the decomposed operations *Contact supplier*, *Supplier provides quote*, and *Place order to supplier*. We can observe that the goal and its decomposed operations all specify an action

Table 2. Results of Linguistic Consistency Checks

	Characteristic	BOF	CR	SES
Component Consistency	No. of checked goals	12	16	21
	No. of consistent goals	6	6	6
	No. of inconsistent goals	6	10	15
Component Stringency	No. of checked goals	12	16	21
	No. of consistent goals	4	5	16
	No. of inconsistent goals	8	11	5
Homonym Consistency	No. of checked goals	12	16	21
	No. of homonym consistent goals	0	1	11
	No. of homonym inconsistent goals	12	15	10
Synonym Consistency	No. of checked goals	12	16	21
	No. of synonym consistent goals	11	14	21
	No. of synonym inconsistent goals	1	2	0
Goal-Logic Consistency	No. of decompositions	12	13	6
	No. of AND-decompositions	11	11	6
	No. of AND-consistent decompositions	4	4	0
	No. of AND-inconsistent decompositions	7	7	6
	No. of OR-Decompositions	1	2	0
	No. of OR-consistent decompositions	0	0	0
	No. of OR-inconsistent decompositions	1	2	0

and an object and are consistent according to this condition. In contrast, the SES model has the worst performance with only 6 consistent goals out of 21. Having a closer look, we observe many goals only containing a single object without an action, such as the goal *No casualties* and its sub goals *Safe entrance and exit* and *Stay safe inside the cage* as depicted in Table 3. In this case, the super goal and the first sub goal only specify the object, but lack an action. Accordingly, this decomposition is considered to be inconsistent.

The results of the *component stringency* condition identifies the SES goal model as the most consistent as it consistently narrows down the objects in the goal decomposition. For this example, the approach found 16 of 21 goal decompositions to be consistent. To provide an example, we choose the goal *Station reachable* from CR model. This goal is decomposed into the operations *Find station* and *Get close to station.* Since the goal and both operations deal with the same object, the decomposition is reckoned as consistent. In case of the BOF model, the approach detected the highest number of stringency issues. Only 4 decompositions have been recognized as stringent which may indicate that an intermediate decomposition layer is actually missing. For example, the decomposition of the goal *Books delivered* fails to specify the object *book* for both sub goals, i.e. it is unclear what object is actually delivered to the courier or to the customer.

Table 3. Qualitative Results of Consistency Checks

Consistency Criterion	**Super Goal**	**Sub Goals**	**Consistent (y/n)**	**Explanation**
Component Consistency	No casualties	Safe entrance and exit Stay safe inside the cage	n	Action is missing in super goal and first sub goal
	Books ordered	Contact supplier Supplier provides price Place order to supplier	y	All goals contain an action and an object
Component Stringency	Books delivered	Deliver to courier Courier delivers books to customer	n	Object *book* missing in both sub goals
	Station reachable	Find station Get close to station	y	Object *station* present in all goals
Homonym Consistency	Dust reachable	Find dust Get close to dust	n	Homonyms found: find, dust
	Main crawler available	–	y	No Homonyms found
Synonym Consistency	Battery maintained	Observe battery level Station reachable Charge battery	n	Synonyms found: maintain, observe
	Books acquired	Supplier ships books Books arrive at warehouse	y	No Synonyms found
AND-Consistency	No casualties	Safe entrance and exit Safe stay inside the cage	n	Closeness: 0.064
	Payment via money order	Customer issues money order Customer sends money order Receive money order	y	Closeness: 0.444
OR-Consistency	Payment received	Payment via money order Payment via credit card	n	Closeness: 0.5

The *homonym consistency* detects a big amount of homonym inconsistencies for all three goal models, i.e. each goal decomposition at least contains one homonymous action or object. However, we have to qualify this observation as nearly every action or object has multiple word senses according to lexical databases. Yet we observe that the SES model scores best with respect to this criterion due to the fact that ambiguous actions are missing. We also find this pattern in the CR model for the goal *Main crawler available*. In this case, the approach does not recognize the main crawler as homonym and thus confirms the consistency with respect to homonyms. A counter example of the same model is the goal *Dust reachable* along with the operations *Find dust* and *Get close to dust*. In this case, the action *to find* and the object *dust* are recognized as homonyms.

We also evaluated the *synonym consistency* for each goal decomposition. In contrast to the homonym case, we could only identify minor issues of synonymy. In fact, only the BOF and the CR goal model contained synonym violations which are caused by synonym actions. We will consider the goal *Battery maintained* of the CR model as an example. This goal is subdivided into the operations *Observe battery level* and *Charge battery* as well as the goal *station reachable*. This decomposition is considered as inconsistent with respect to synonyms as the actions *to maintain* and *to observe* were recognized as synonyms.

In the end, we analyzed the *AND-consistency* as well as the *OR-consistency* of all three goal models. To do so, we had to exclude those goal decompositions that specified no or only one subgoal or operation as this criterion requires a pair-wise comparison of the subgoals and operations in the goal decomposition. The evaluation shows that, again, all three goal models suffer from this type of inconsistency. On a relative scale, the BOF model appears to be the most consistent as 4 of 11 decompositions satisfy this criterion. As examples, we consider the goals *No casualties* of the SES model and *Payment via money order* of the BOF model. For the first example, the super goal is decomposed into the subgoals *Safe entrance and exit* and *Safe stay inside the cage*. For this decomposition, the Lin measure calculates a closeness of 0.064 which reflects a hardly existing relation and might indicate an inconsistency with respect to the AND-decomposition. In the second example, the Lin measure evaluates to 0.444 which indicates a moderate relation. Accordingly, we recognize this decomposition as consistent with respect to the AND-decomposition. For the OR-decomposition, the results are even more striking. Exclusively, the approach detects inconsistencies for the CR and the BOF model which indicates that the subgoals are highly close to each other although the opposite was intended.

4.3 Discussion

Implications for Practice. The results of this paper have considerable implications for practice. Most importantly, the proposed techniques can be integrated into goal modeling tools. In such a context, they can point stakeholders to inconsistent goals who in turn can resolve them directly. This procedure will lead to consistent goal models in the beginning. For existing goal models, our consistency conditions can help stakeholders to effectively validate and consolidate the created goal models and to derive consistent requirements for software analysts.

Implications for Research. In goal-oriented software development, other goal model notations, such as i* [37] or Tropos [13], are used to formally specify system requirements. Analogously, goal models using these notations have to be consistent in order to avoid risks for the software development project and to improve the chances for a successful software deployment [1]. The consistency criteria of this paper are conceptualized as notation-independent and are thus also applicable to i*, Tropos and other goal modeling notations since they show similar linguistic characteristics as the KAOS goal models that were subject to this paper.

Limitations. The findings from this paper are subject to some limitations. In particular, we discuss limitations with respect to completeness of linguistic aspects and the representativeness of the evaluation samples.

The introduced criteria can hardly be seen as encapsulating all linguistic aspects of goal models. Therefore, we cannot say for sure that other linguistic aspects, for example the context in which the model was created and from which the goals were derived, might be worth investigating as well. However, we are confident that we tackled and motivated an interesting direction of goal model research and that we also proposed a set of linguistic consistency conditions that serve as a promising starting point.

Then, we only evaluated three goal models from literature are not statistically representative. Thus, we cannot completely rule out that other goal models or a collection of several goal models would yield different results. We tried to minimize this risk by taking a qualitative-oriented approach and theoretically sampling goal models from existing sources that vary along different dimensions such as domain, size, and standardization. The work in [33] states that industrial case studies can include several hundreds of goals. Although the applied case studies are based on models that include between 12 and 21 goal, we do not expect a negative impact on the general applicability. Hence, we are confident that the successful application of these criteria is not limited to a particular sampling of goal models.

5 Conclusion

In this paper, we presented a novel approach for assuring the linguistic consistency of goal models. Motivated by current research papers, we proposed five different consistency conditions that evaluate the syntactic and semantic dimension of the natural language fragments of goal models. The consistency conditions have been implemented prototypically and evaluated with three case studies from the literature with different characteristics. The evaluation demonstrates the capability of finding and explaining linguistic inconsistencies in goal models and stimulates further endeavors for research and practice.

In future research, we want to address the identified limitations of this paper. First, we plan to improve the consistency conditions with more sophisticated means of detecting synonym and homonym terminology that explicitly exploit the goal model context. Furthermore, we also want to integrate ontology technology to accurately evaluate the goal-logic consistency more precisely. Second, we intend to evaluate the conditions in user experiments. Such experiments will involve the ranking and the usefulness of the detected inconsistencies based on human judgment. For this purpose, we want to gain a cooperation partner from practice in order to further increase the amount of assessable goal models. Nevertheless, the research of this paper can be regarded as an important and complementary step towards the quality assurance of goal models.

References

1. Abran, A., Bourque, P., Dupuis, R., Moore, J.W., Tripp, L.L.: Guide to the Software Engineering Body of Knowledge - SWEBOK. IEEE Press (2004)
2. Adrion, W.R., Branstad, M.A., Cherniavsky, J.C.: Validation, verification, and testing of computer software. CSUR 1982 14(2), 159–192 (1982)
3. Alrajeh, D., Kramer, J., Russo, A., Uchitel, S.: Learning operational requirements from goal models. In: ICSE 2009, pp. 265–275. IEEE Computer Society (2009)
4. Anton, A.I.: Goal-based requirements analysis. In: Proc. of the Second Int. Conf. on Requirements Engineering, pp. 136–144 (1996)
5. Boehm, B.W.: Verifying and validating software requirements and design specifications. IEEE Software 1(1), 75–88 (1984)
6. Bryl, V., Giorgini, P., Mylopoulos, J.: Supporting requirements analysis in tropos: A planning-based approach. In: Ghose, A., Governatori, G., Sadananda, R. (eds.) PRIMA 2007. LNCS, vol. 5044, pp. 243–254. Springer, Heidelberg (2009)
7. Bryl, V., Massacci, F., Mylopoulos, J., Zannone, N.: Designing security requirements models through planning. In: Martinez, F.H., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 33–47. Springer, Heidelberg (2006)
8. Chakraborty, S., Sarker, S., Sarker, S.: An exploration into the process of requirements elicitation: A grounded approach. Journal of the Association for Information Systems 11(4), 212–249 (2010)
9. Dardenne, A., Fickas, S., van Lamsweerde, A.: Goal-directed concept acquisition in requirements elicitation. In: IWSSD 1991, pp. 14–21 (1991)
10. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20, 3–50 (1993)
11. Deissenboeck, F., Pizka, M.: Concise and consistent naming. Software Quality Journal 14(3), 261–282 (2006)
12. Fuxman, A., Pistore, M., Mylopoulos, J., Traverso, P.: Model checking early requirements specifications in tropos. In: Proceeding of the Fifth IEEE Int. Symposium on Requirements Engineering, pp. 174–181 (2001)
13. Giorgini, P., Kolp, M., Mylopoulos, J., Pistore, M.: The tropos methodology. In: Methodologies and Software Engineering for Agent Systems, pp. 89–106 (2004)
14. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Requirements engineering for trust management: Model, methodology, and reasoning. International Journal of Information Security 5(4), 257–274 (2006)
15. Hinkle, D.E., Wiersma, W., Jurs, S.G.: Applied statistics for the behavioral sciences (2003)
16. Horkoff, J., Yu, E.: Analyzing goal models: Different approaches and how to choose among them. In: Proc. of the ACM Symposium on Applied Computing, pp. 675–682. ACM (2011)

17. Kamsties, E.: Understanding ambiguity in requirements engineering. In: Engineering and Managing Software Requirements, pp. 245–266. Springer (2005)
18. Koliadis, G., Ghose, A.K.: Relating business process models to goal-oriented requirements models in KAOS. In: Hoffmann, A., Kang, B.-H., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 25–39. Springer, Heidelberg (2006)
19. Kueng, P., Kawalek, P.: Goal-based Business Process Models: Creation and Evaluation. Business Process Management Journal 3(1), 17–38 (1997)
20. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Information Systems 37(5), 443–459 (2012)
21. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Conference on Systems Documentation (1986), pp. 24–26. ACM (1986)
22. Liaskos, S., McIlraith, S., Sohrabi, S., Mylopoulos, J.: Integrating preferences into goal models for requirements engineering. In: RE 2010, pp. 135–144 (2010)
23. Lin, D.: An information-theoretic definition of similarity. In: MLDM 1998, ICML 1998, pp. 296–304. Morgan Kaufmann Publishers,
24. Mendling, J., Reijers, H.A., Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. Inf. Systems 35(4), 467–482 (2010)
25. Miller, G., Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
26. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
27. Nakagawa, H., Ohsuga, A., Honiden, S.: gocc: A configuration compiler for self-adaptive systems using goal-oriented requirements description. In: SEAMS 2011, pp. 40–49. ACM (2011)
28. Prat, N.: Goal formalization and classification for requirements engineering, fifteen years later. In: RCIS 2013, pp. 1–12 (2013)
29. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: IJCAI 1995, pp. 448–453 (1995)
30. Respect IT. A KAOS Tutorial, Objectiver (2007)
31. van Lamsweerde, A.: From system goals to software architecture. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 25–43. Springer, Heidelberg (2003)
32. van Lamsweerde, A.: Goal-oriented requirements engineering: From system objectives to UML models to precise software specifications. In: Proc. of the 25th Int. Conf. on Software Engineering, pp. 744–745. IEEE Computer Society (2003)
33. van Lamsweerde, A.: Goal-oriented Requirements Engineering: A Roundtrip from Research to Practice. In: Proceedings of the 12th IEEE International Requirements Engineering Conference, pp. 4–7 (2004)
34. van Lamsweerde, A., Darimont, R., Letier, E.: Managing Conflicts in Goal-Driven Requirements Engineering. IEEE Transactions on Software Engineering 24(11), 908–926 (1998)
35. van Lamsweerde, A., Letier, E.: Handling Obstacles in Goal-Oriented Requirements Engineering. IEEE Transactions on Software Engineering 26(10), 978–1005 (2000)
36. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL 1994, pp. 133–138. Association for Computational Linguistics (1994)
37. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proc. of the Third IEEE Int. Symposium on Requirements Engineering, pp. 226–235 (1997)

Devising DEMO Guidelines and Process Patterns and Validating Comprehensiveness and Conciseness

David Aveiro[1,2,3] and Duarte Pinto[1]

[1] Exact Sciences and Engineering Centre, University of Madeira, Caminho da Penteada 9020-105 Funchal, Portugal
[2] Madeira Interactive Technologies Institute, Caminho da Penteada, 9020-105 Funchal, Portugal
[3] Center for Organizational Design and Engineering, INESC-INOV Rua Alves Redol 9, 1000-029 Lisboa, Portugal
daveiro@uma.pt, duartenfpinto@gmail.com

Abstract. This case study paper presents DEMO models of a very complex process of urban construction licensing from a city hall. From our practical experience in this project, we elicit some guidelines and process patterns that may be useful to other similar projects and also guide DEMO modelers in similar scenarios of process complexity. From the metrics we got from this case study, we provide an empirical validation of DEMO's qualities of comprehensiveness and conciseness. Thanks to the nature of the transaction axiom, we managed to uncover hidden or neglected important process steps, not captured in the results of models previously obtained by the use of a flowchart approach.

Keywords: Enterprise engineering, DEMO, case study, guidelines, process pattern, validation.

1 Introduction

Enterprise Engineering aims to develop thorough theories, methods and tools to design, engineer and implement organizations. After decades of experience and progress in the discipline of software engineering, many software design guidelines and patterns have been elicited that guide software engineers in their work, making it more effective. Even though the Design and Engineering Methodology for Organizations (DEMO) has a set of proposed generic method steps and some proposed ways of working, as well as sound theories behind it, we claim that this is far from sufficient for a widespread adoption of DEMO. We consider that a good number of very complex real life DEMO projects, as well as important lessons learned – in the form of guidelines and process patterns – have to be presented to the scientific and practitioner communities, so that such lessons can be reused in other projects and then, the body of knowledge of these guidelines and patterns themselves, further improved. We envision a future where a good enterprise engineer will be a person with a high degree of knowledge in guidelines and process patterns that complement proposed methods.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 408–423, 2014.

This paper aims to be a relevant step in the path to that vision while presenting the DEMO models of a very complex process of urban construction licensing from a real-world city hall. From our practical experience in this project we elicit some guidelines and process patterns that may be very useful to other projects and also guide DEMO modelers in similar scenarios of process complexity. Furthermore, from metrics we got from this case study, we provide an empirical validation of DEMO's qualities of comprehensiveness and conciseness. Thanks to the nature of the transaction axiom, we managed to uncover hidden or neglected important process steps, not captured in the results of models previously obtained by the use of a flowchart approach.

Section 2 presents our Research method and problem. Next, in section 3, we present DEMO - Operation and Transaction Axioms. Section 4 has our Project steps, case models and description obtained from a series of meetings. Section 5 explores our results of Lessons learned and devising guidelines and process patterns. In section 6 we do our Validation of DEMO's conciseness and comprehensiveness and finally, in section 7, we present our Conclusions.

2 Research Method and Problem

On this section we present the research method used as well as the motivation behind this paper. A set of seven guidelines are proposed in [1] for understanding, executing, and evaluating research in Information Systems (IS). In order to assess how the design artifacts presented in this paper meets IS research standards we use the respective guidelines, as described below.

Guideline 1: Design as an Artifact - in this paper three main artifacts are presented: (1) DEMO models of a very complex real life process, (2) a set of guidelines and process patterns that were devised from this case and (3) a validation of the claimed DEMO qualities of comprehensiveness and conciseness.

Guideline 2: Problem Relevance - There is a great lack of guidelines and process patterns to complement already proposed Ways of Working of DEMO and this hinders a more widespread adoption of this method. Another problem that seems to impede such adoption is the lack of published large real life cases and convincing validations of the claimed qualities for this method.

Guideline 3: Design Evaluation - To evaluate the utility of the design artifact we applied the "*Case Study*" technique from Hevner's observational evaluation method. With a long process of analysis of the existing documentation, meetings with the city hall's collaborators, validations of the produced models, and production and analysis of certain metrics of this project, it was possible to reach and ground the conclusions presented in this paper.

Guideline 4: Research Contributions - The huge complexity of the modeled process is a rich source of knowledge that may be reused in similar contexts and also for facilitating widespread adoption of DEMO, as well, as a training example of a complex case. The guidelines and process patterns we identified seem to be useful and generic enough to be reused in other projects and contribute to the body of knowledge of DEMO Ways of Working. The first guideline aims to avoid the necessity of rolling back or canceling many useless c-acts and c-facts, by only starting sequential enclosed transactions after the previous one has been accepted. The second artifact is a pattern to be applied in DEMO's process model that helps us deal with cases of parallel join of and type, where one transaction needs to wait for multiple transactions possibly being executed in parallel. The third artifact can be considered as both a guideline and a pattern and aims to facilitate complex decision processes, by proposing the creation of a transaction that may be initiated in multiple points throughout the process if needed.

Guideline 5: Research Rigor - The process used follows a rigorous step-by-step logical reasoning, using the solid theoretical foundations from DEMO as properly explained throughout the whole document.

Guideline 6: Design as a Search Process - This paper has the advantage of having DEMO as base, which provides a set of coherent and solid definitions for many organizational concepts which constitute "laws" that help direct the construction of the artifacts. The artifacts themselves resulted from highly interactive process of many meetings with the organization's collaborators where we kept searching missing details of processes and also guidelines and patterns that could be useful and reused in similar contexts. Also by looking at particular metrics of our project efforts we managed to realize one of our aims: to validate some of DEMO's most important qualities.

Guideline 7: Communication of Research - To communicate our research and conclusions we are using this paper.

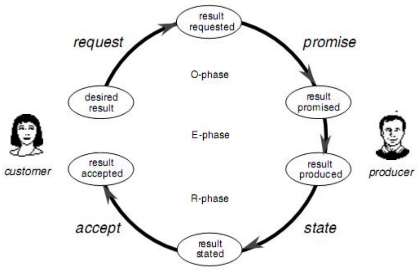


Fig. 1. Actors Interaction with Production and Coordination Worlds[2]

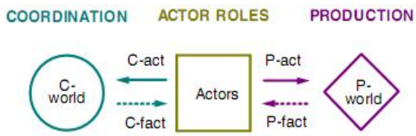


Fig. 2. Actors Interaction with Production and Coordination Worlds[2]

3 DEMO - Operation and Transaction Axioms

In the Ψ-theory [2] – on which DEMO is based – the operation axiom [3] states that, in organizations – that are considered systems – subjects perform two kinds of acts: production acts (P-acts) that have an effect in the production world and coordination acts (C-acts) that have an effect on the coordination world. Each of these worlds can be considered as the set of effects and/or facts produced by the acts of the system. Subjects are actors performing an actor role responsible for the execution of these acts. At any moment, these worlds are in a particular state specified by the C-facts and P-facts respectively occurred until that moment in time. When active, actors take the current state of the P-world and the C-world into account. C-facts serve as agenda for actors, which they constantly try to deal with. In other words, actors interact by means of creating and dealing with C-facts. This interaction between the actors and the worlds is illustrated in Figure 1. It depicts the operational principle of organizations where actors are committed to deal adequately with their agenda. The production acts contribute towards the organization's objectives by bringing about or delivering products and/or services to the organization's environment and coordination acts are the way actors enter into and comply with commitments towards achieving a certain production fact [4]. Examples of P-facts belonging to a pizzeria's P-world can be: *"Pizza #120 has been ordered"* or *"Pizza #233 has been delivered";* whilst examples of C-facts belonging to the pizzeria's C-world can be: the request of the production fact *"Pizza #120 has been ordered"* (calling the pizzeria and requesting a desired pizza) or the acceptance of the production fact *"Pizza #233 has been delivered"* (accepting the pizza brought by the delivery man).

According to the Ψ-theory's transaction axiom the coordination acts follow a certain path along a generic universal pattern called transaction [3]. The transaction pattern has three phases: (1) the order phase, were the initiating actor role of the transaction expresses his wishes in the shape of a request, and the executing actor role promises to produce the desired result; (2) the execution phase where the executing actor role produces in fact the desired result; and (3) the result phase, where the executing actor role states the produced result and the initiating actor role accepts that result, thus effectively concluding the transaction. This sequence is known as the basic transaction pattern, illustrated in Figure 1, and only considers the "happy case" where everything happens according to the expected outcomes. All these five mandatory steps must happen so that a new production fact is realized. In [4] we find the universal transaction pattern that also considers many other coordination acts, including cancellations and rejections that may happen at every step of the "happy path". Even though all transactions go through the four – social commitment – coordination acts of request, promise, state and accept, these may be performed tacitly, i.e. without any kind of explicit communication happening. This may happen due to the traditional "no news is good news" rule or pure forgetfulness which can lead to severe business breakdown. Thus the importance of always considering the full transaction pattern and the initiator and executor roles when designing organizations [4].

4 Project Steps, Case Models and Description

In figures 3, 4 and 5 we present the 3 parts of our actor transaction diagram and in figure 6 we present part of the process structure diagram of this case. In the text that follows we present the full case description of this process. This description can be considered as a general explanation of the operation of this process, structured around the final result of the modeled transactions. These final models were the result of a lengthy process of several meetings that took place with the involved stakeholders from the city hall, namely: a lawyer, an architect, an engineer and the city councilman. We started by realizing a Performa-Informa-Forma (PIF) and a Coordination-Actor-Production (CAP) analysis [3] of the flowcharts provided to us in our first meeting. These analysis gave origin to a first version of the Transaction Result Table (TRT) and of the Actor Transaction Diagram (ATD). These were used as a base for the second meeting where some corrections were made and new information was gathered and refinements were introduced into the models. During the iterations of this process frequently new information would be reminded either in the form of other flowcharts or process steps that were not written anywhere but were somehow following the national law and existed only in the minds of the city hall's collaborators. After several iterations of the previous steps, done until we got a relatively stable ATD, we produced the Process Structure Diagram (PSD) that aims to serve as a basis to configure a future workflow system to automate most of the executed work, currently mostly paper based. While validating this diagram, new information in the form of new process steps and new process step inter-dependencies would be found, which lead to the specification of even more transactions and new versions of the TRT, ATD and PSD. In our experience of this project we witnessed in practice the power of the operation and transaction axioms of DEMO. Compared to other modeling approaches, the fact that we keep asking to the interviewees about all the steps of each transaction and all the time clarifying who initiates and who executes each transaction allows us to uncover many hidden or tacit responsibilities and process steps. After this summary of our project steps, the full case description follows which will serve as a basis for the presentation of our contributions.

A citizen comes to the city hall and heads to the construction department desk and expresses the wish to acquire a license for a construction bringing the respective project's documents. The clerk initiates the procedure by creating a new process instance in the system and stamps the delivered documents with the date, the kind, the application number and the number of pages and then verifies the citizen's signature. The clerk assigns a process manager to this process instance and requests the citizen to pay the fee relating to the registration of a new process. Afterwards, the clerk delivers the documents to the process manager. He then initiates a preliminary analysis, verifying the

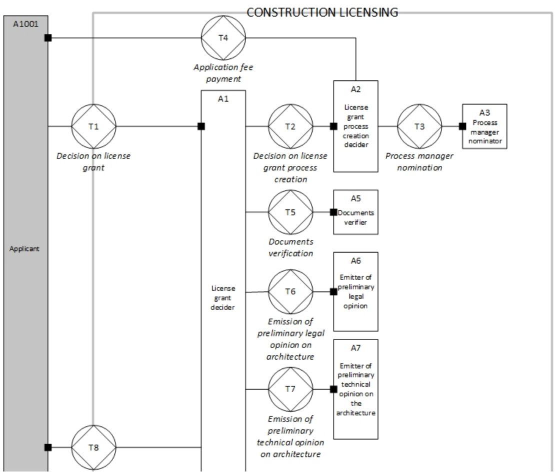


Fig. 3. Construction licensing ATD - part 1

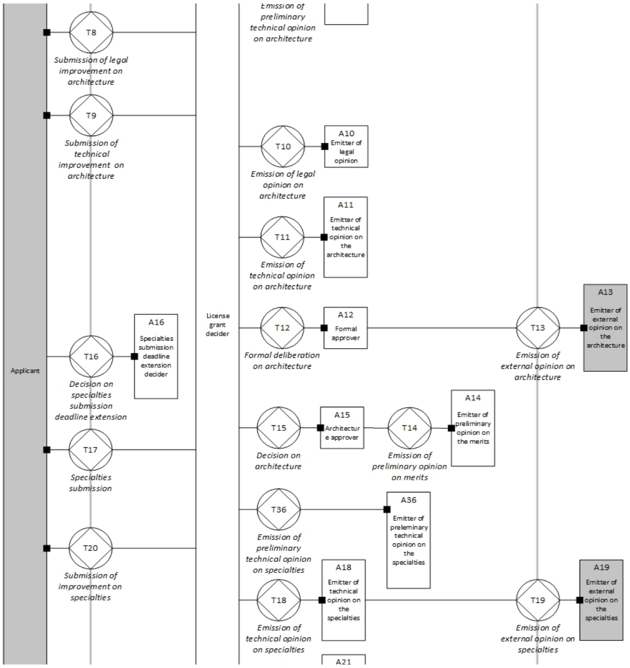


Fig. 4. Construction Licensing ATD - part 2

delivered documents and specifying in a check list the documents that were delivered and the ones that are missing. He then forwards all documents to the legal office.

The lawyer then makes a preliminary legal analysis of the documents and emits a preliminary juridical opinion where eventual missing legal documents are pointed out. The process is then forwarded to the architecture office where the architect analyzes the document and issues a preliminary opinion on the architecture project. If problems arise from these analysis, the citizen is notified by the city hall with an official letter requesting the submission of improvements in legal and/or technical aspects. After the eventual legal improvement, the lawyer may then issue a final legal opinion on the licensing request. After the eventual architecture improvement is submitted, the architect will issue the final technical opinion on the architecture. The chief of the urban and planning division checks the legal and architecture opinions issued about this process and assesses the necessity to ask for further external opinions on the matter. As soon as there is the emission of the external opinion on the architecture, the chief of division confirms all the opinions and the process is forwarded to the city councilman that makes a final appreciation of the architecture, assessing if all the administrative acts, either internal or external, are acceptable. If there is no need for the emission of a preliminary opinion on the final merit by the lawyer, the approval of the architecture will take place. Otherwise, it can be concluded that the project has no conditions to be executed and the process will come to an end, where an opinion is emitted that states the construction as infeasible. The process manager is then alerted to notify the

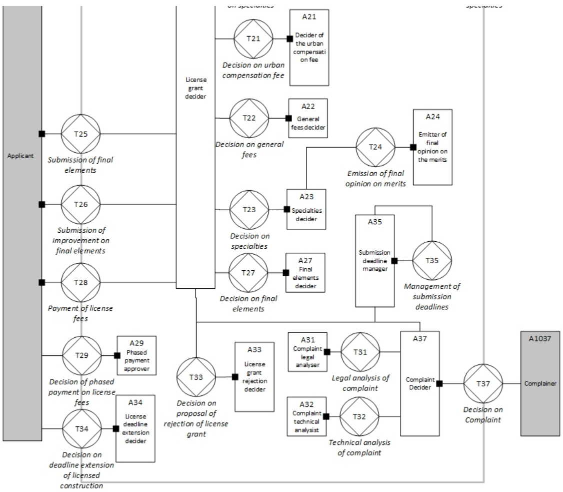


Fig. 5. Construction Licensing ATD - part 3

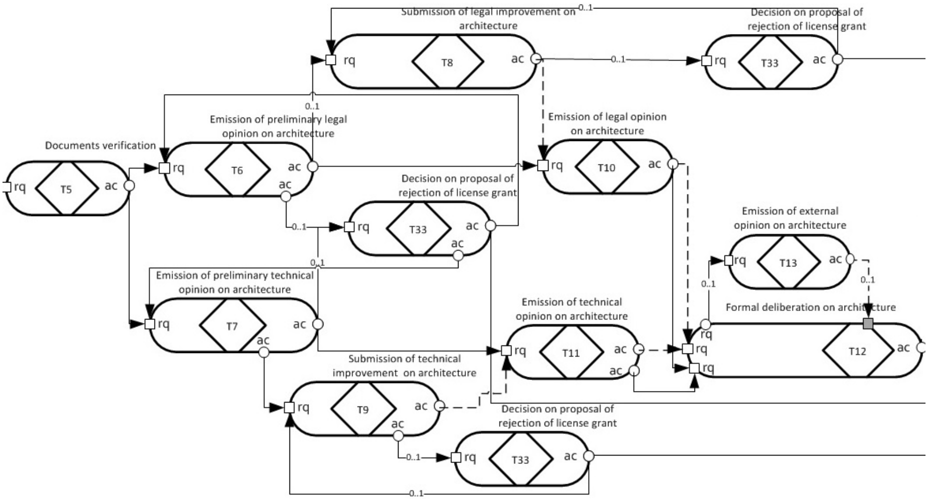


Fig. 6. Construction Licensing PSD (partial view)

citizen of the result of the municipal administrative acts that culminated in the rejection of the license. In the case of a positive decision, the citizen receives a notification of the approval of the architecture project together with a request to submit the specialties project, that is, the detailing of the several relevant technical designs for the complete construction project. If, by any reasons, the citizen does not deliver the specialties project within a 6 month time frame, he still can ask for a one time prorogation of the delivery date and for a period not longer then 3 months as long as he can supply proper grounding for such request. If the deadline is not respected, the city hall will terminate the process with a decision on rejecting the license. After the citizen submits the specialties project, the engineer of the technical services of construction department analyzes the content of the documents, and emits a technical opinion on the specialties. When there is the need to consult with external entities, the request of an external technical opinion on the specialties is made and after the reception of the external opinion, the engineer emits his final technical opinion. The process is then forwarded to the civil engineer in charge of the technical services of construction for analysis. The emission of the preliminary technical opinion over the specialties projects may conclude that there are aspects that need to be revised by the citizen. When this happens, the process manager, requests the citizen, in a notification, to submit an improvement on the specialties documents which, when delivered, is forwarded to the responsible civil engineer of the technical services of construction for a new appreciation. The civil engineer then emits his final opinion in the form of a legal opinion. This opinion can be positive if the project fulfills all the current legal obligations or negative if there is any objection. When everything is according to the law and requirements, the civil engineer, after emitting the opinion, includes a proposal on the fee to be paid by the citizen, and then the process is forwarded to the architecture office. The nominated architect is asked to decide on the urbanization compensation fees to be applied and that is followed by the decision on applicable taxes done by the technical coordinator. This information is then presented to the city councilman that will, in his turn, decide over the specialties taking into consideration the whole process. He approves the specialties if the process is according to the laws and rejects it if there are any flaws. If any legal questions arise and so that all ambiguities are answered, the city councilman may request another legal opinion on the merits of the project before his final decision. As soon as the lawyer emits his final appreciation of merit, the process is forwarded again to the city councilman so that he may decide over the specialties. After the decision and with proper authorization by the city councilman, the citizen is notified regarding the final appreciation of the specialties and is asked to submit to the city hall the final elements (documents and other details) indispensable for obtaining the construction license. When the citizen submits the final elements to the city hall, the city councilman verifies them and if any irregularity is detected he requests that the citizen submits an improvement on the final elements so he can remedy the process. When the city councilman receives the process including these final improvements, he evaluates the final elements and decides to approve them or not. If the decision is positive he will in turn decide in a positive way on the main decision of the granting of the construction license. The citizen is informed by a notification about the decision and on the fee of urban charges to be paid. If the citizen cannot afford the total amount of the urban charges at once he may request the city hall to approve a phased payment. This request is delivered to the process manager that forwards it to the city councilman. He then verifies the argumentation and makes the decision regarding the phased payment of the urban charges.

While the licensing process happens or even after it has been approved a third party with interest may go to the city hall examine ongoing processes, and, if considering that there is some harm, he or she may submit a written complaint in the urban division. The process manager will deliver the complaint of the opposing third party to the licensing office of the city councilman so that the licensing process or the construction itself may be halted, and the reasons that lead to the complaint may be analyzed. The complainant can add further information to the process and assist in the decision making of the licensing of the construction. The city councilman is informed immediately whenever a complaint of a construction process is made, and if needed, may ask the legal office if there is grounding or if it reports to questions of private rights to which the city hall has no jurisdiction. To that end the lawyer is asked to issue his legal opinion on the complaint. After issuing his legal opinion, the lawyer may also send the process to the technical office of architecture asking for a technical appreciation. After gathering the information regarding the groundings of the complaint the city councilman makes a final decision on the complaint.

When close to the stipulated end date conceded for the conclusion of the construction, if the contractor realizes that more time is needed, he informs the person responsible for the request, and this person goes to the city hall to request that the stipulated deadline for the conclusion of the construction may be extended. The city councilman analyzes the request and makes his decision after which the process manager informs the citizen. At any moment in the whole process whenever someone considers that the license must be rejected, a proposal for a decision on rejection is made and it is requested that the city councilman takes such decision. He will make a decision on this proposal and if the decision is to reject, then the license will be declined and such declination will be communicated to the citizen. If he decides not to reject then a request of the previous process step is made so that a new appreciation is made so that the process can continue where it was.

5 Lessons Learned and Devising Guidelines and Process Patterns

While modeling this very complex process that has nearly 40 transactions, we faced several instances of having to choose between different alternative ways to model certain process flows and inter-dependencies. Solutions or guidelines for handling and deciding on such alternatives cannot be found in currently proposed DEMO Ways of Working. We devised alternatives for certain modeling problems and took decisions which seemed the best and most elegant way to solve such problems. From this experience we produced the following set of guidelines and process patterns, proposed to be part of the DEMO Way-of-Working's knowledge base.

From the modeled process of our case, we see that actor role *license grant decider* has a pivotal role in the process, in the sense that it coordinates the execution of all of the transactions directly enclosed in transaction *T01 - license grant decision*. Now the question arose on how to specify the causal links that initiate each of the enclosed transactions. By following the guidelines described in DEMO Way of Working version 2 [3], one would have to specify that *when T01 is requested* then all enclosed transactions at the next level are also requested. The respective action rule would be something like: *when T01 is requested then T02 must be requested; T05 must be requested; T06 must be requested* etc. Then the action rules that handle the request of their respective transaction would have the form: *when T05 is requested and T02 is* *accepted then...* where the acceptance of T02 would be a conditional link between T02 accept and T05 request. From a pure conceptual point of view and in small processes like the ones on the examples of the library and the pizzeria in [3] (p. 192) such modeling option makes sense. But from a practical point of view, and taking in account that the Process Model is supposed to directly guide the design of a workflow process in a Workflow Management System [3] (p. 83), that option does not make sense. In our case, the action rule that deals with T01 request would lead to the request of around 10 other transactions. Being the case that, in many real life instances of this process, execution does not reach even half of the way when the license grant is declined. That would lead to the necessity of rolling back or canceling many useless c-acts and c-facts. The alternative option and guideline that we propose is that, *whenever there are transactions enclosed in a higher level transaction and these enclosed transactions are, by nature, executed in a sequential fashion, then only after the acceptance of the preceding transaction should we execute the act of requesting the next transaction*. This is the guideline that is followed throughout our case and visible in the PSD in Figure 6. Following this guideline also results in a simpler and cleaner diagram with less line clutter that would result of following the standard old-fashioned approach. This guideline can be seen as the application of LEAN principles [5] to enrich DEMO's body of knowledge as to reduce "process waste" and inefficiencies.

Another interesting problem we faced – and no publicly available case shows – is how to specify, in the Process Model, a flow situation of a *parallel join of and type*. In our case, when the architecture project is submitted, after the document verification transaction there is a *parallel fork* of the flow since the emission of legal opinion (T6, T8 and T10) can occur in parallel with the emission of technical opinion (T7, T9 and T11). However, the formal deliberation on architecture (T12) can only proceed if both the previous transactions (T10 and T11) have finished, i.e., have been accepted. So two mandatory conditional links have to connect the accept of the previous transactions with the request of the T12. And we also need two mandatory causal links linking these same c-acts/facts, since whichever transaction finishes first, the request of T12 will have to wait for the accept of the other, and when this accept is a fact, it will finally cause the advancement of the process. Concluding, a proposed process pattern is: in DEMO's Process Model, *one specifies a parallel join of and type of N transactions by linking the accept c-fact of these N transactions with the request c-act of the following transaction, both with a mandatory conditional and a mandatory causal link*. This pattern can be seen as a DEMO counterpart of the workflow pattern known – for both BPMN and UML's activity diagram – as "*Synchronization*" described in [6]. Our pattern is, nevertheless, an innovative contribution, as it was not clear or obvious how that could be done with DEMO. Although we don't find that case in our example, as logical induction, a *parallel join of type or* would be represented in a similar fashion but the *conditional links would be all of optional type*. This can be considered as the DEMO counterpart of the pattern "*exclusive choice*" from [6]. As both these patterns create some considerable clutter in the diagram we propose to the DEMO standard managers to consider the specification of special link kinds with specific symbols to denote these cases of *parallel join* of *and* and *or* types. One could argue that, due to the highly complex nature of the modeled decision process, a declarative workflow approach like [7] or [8] should be followed as to reduce flow clutter that our proposed guideline seems to imply. But DEMO's underlying theory considers an organization as a system with state changes affecting the world. On top of that, the causal and conditional links in the process model end up being conceptually equivalent to the automata structure of the declarative workflow approach which focuses on constrains and not on the many possible flows. So DEMO's quality of conciseness comes here into play. In the declarative workflow approach, the automata are a really concise way of representing the structure of constrains that represent the possible transition space of a process. DEMO's Process Model realizes the same in a more intuitive fashion, thanks to the notation used in PSD which reminds both BPMN and flowchart elements.

Another pattern we identify is that, in these cases where one is modeling a complex decision process (having as root transaction T01), a negative decision would be the execution of the decline act of T01 and the promise act will already constitute a positive decision. And during the execution of the process, in many possible points of the flow a sub-decision transaction may cause the decline of the global decision T01. Now, looking at our case, we see that, if in any point in the process, some actor role makes some negative decision or opinion, that will cause actor A01 to initiate T33 called *Decision on proposal of rejection of license grant*. Thus, the request of this transaction can be caused by the accept c-fact of many different transactions as we can see on the presented PSD. And a possible outcome is that the executor of T33 may decide that the license grant should not be rejected and the process should continue. In our modeling efforts the question arose: should each of these possible decisions on proposal of rejection be a separate transaction for each point it can happen or is it indeed the same transaction but requested in different points of the process? Since it's the same person/role that takes the decision it makes sense to become only one transaction. And also it is always actor A01 who requests T33 because some intermediary decision transaction decided the there should be a rejection. So A01 will have a very complex action rule that, according to the stage of the whole process, will have to probably repeat a request of some decision transaction that had lead to the proposal of the rejection. Concluding, another guideline and process pattern that we identify and may be generalized and reused in other projects is that *on complex decision processes with many transactions and sub-decisions one should consider specifying a transaction that can be requested whenever it's appropriate and that consists in a decision on the proposal of rejection of the global decision. Such decision transaction can then cause the decline of the main decision transaction or cause the repetition of a request, probably of the transaction that has led to the rejection proposal.* This may, at first sight, seem over-bureaucratic. But such transactions are really needed to clarify responsibilities and opinions of the participants in crucial decisions that involve huge amounts of resources normally allocated to these kind of construction processes. This third artifact ends up showing how the DEMO approach is indeed powerful as it naturally embeds the philosophy that any business process is a tree of transactions [3] and, consequently, a complex decision process will be a tree of decisions. Such a tree structure pattern has been identified in related research like the one found in [9] regarding decision modeling. Our DEMO based approach has the advantage of using the transaction pattern acts (decline and state) to naturally capture the two possible outcomes of a complex decision (license grant declined and license grant approved) and coherently relate them with other parts of the tree, with the possibility of “resuming” the decision process on the point before the proposal for rejection was issued.

6 Validation of DEMO's Conciseness and Comprehensiveness

An interesting outcome of this project was the fact that several key decision points in this whole process were not specified as tasks or decisions in the flowcharts but were hidden somewhere in the descriptions of the flowcharts or in the minds of some collaborator. In this section we present a table with a comparison of the tasks present in existing flowcharts for both the license grant process and the complaint process and their DEMO counterparts. We do the same for the roles, namely the responsible roles found in the flowcharts and the initiating and executing organizational functions (that are directly mapped to DEMO actor roles) we found. Later on in this section, we analyze this comparison and devise a set of metrics that serve as an empirical validation of DEMO's qualities of conciseness and comprehensiveness.

License Grant Process				
Flowchart Task	**Responsible Organizational Role**	**DEMO Transaction**	**Initiating Org. Function**	**Executing Org. Function**
1 Reception of documents; Registration and appointment of a process manager	Clerk	T1 - Decision on license grant (request)	Citizen	City councilman
		T2 - Decision on license grant process creation	Receptionist	City councilman
		T3 - Process manager nomination	Receptionist	Receptionist
		T4 - Application fee payment	Receptionist	Citizen
2 Verification of Architecture documents	Process Manager	T5 - Documents verification	Receptionist	Process manager
3 Preliminary analysis of the legal office	Lawyer	T6 - Emission of preliminary legal opinion on architecture	Process manager	Lawyer
4 Preliminary SAP Analysis: instruction, preliminary assessment	Architect || Process Manager	T7 - Emission of preliminary technical opinion on architecture	Process manager	Architect
(If it is contrary to the rules) 5 Order of outright rejection	City councilman || Process Manager	T1 - Decision on license grant (decline)	Citizen	City councilman
(if missing information) 6 Order perfecting	City councilman / Process Manager	T8 - Submission of legal improvement	Lawyer	Citizen
		T9 - Submission of technical improvement	Architect	Citizen

7 Request opinions to external entities	Process Manager	T13 - Emission of external opinion on architecture	Chief of urban planning division	External entity
8 Consideration of the architecture project	Architect || Lawyer	T10 - Emission of legal opinion on architecture	Lawyer	Lawyer
		T11 - Emission of technical opinion on architecture	Architect	Architect
9 Approval of opinions	Chief of the urban planning division	T12 Formal deliberation on architecture	Lawyer, Architect	Chief of urban planning division
(if unfavorable) 10 Order of dismissal	City councilman	T15 - Decision on architecture	Chief of urban planning division	City councilman
(if favorable) 10 Order of granting	City councilman	T15 - Decision on architecture	Chief of urban planning division	City councilman
11 Notification for submission of specialties project	Process Manager	T16 - Decision on specialties submission deadline extension	City councilman	Lawyer
		T17 - Specialties submission	City councilman	Citizen
12 Verification of specialties and enforceable terms	Civil engineer	T36 - Emission of preliminary technical opinion on specialties	Process Manager	Civil engineer
		T18 - Emission of technical opinion on specialties	Process Manager	Civil engineer
13 Necessary queries to external entities	Process Manager	T19 - Emission of external opinion on specialties	Civil engineer	External entity
14 Determination of deposits, fees for conducting, maintaining and strengthening the primary and secondary urban infrastructures, Compensation and Fees	Technical Coordinator	T21 - Decision on urban compensation fee	Civil engineer	Chief of urban planning division
		T22 Decision on general fees	Chief of urban planning division	Technical Coordinator
15 Final Decision	City councilman	T24 - Emission of final opinion on merits	City councilman	Lawyer
		T23 - Decision on specialities	Technical Coordinator	City councilman

Complaint process				
1. Receiving of the complaint; Registration and assignment of a process manager	Clerk	T37 - Decision on complaint (promise and infological and datalogical acts)	Citizen	City councilman
2. Determination of existence of process and attaching folder	Process Manager	T37 - Decision on complaint (infological and datalogical acts)	Citizen, Lawyer, Architect, City councilman	City councilman

3. Verification of the reason for complaint in the construction	Inspector	T38 - Analysis of complain on construction site	Citizen, Lawyer, Architect, City councilman	City councilman, Construction Inspector
4. Injunction assessment of the validity of the claim by legal office	Lawyer	T31 - Legal analysis of complaint	Citizen	Lawyer
(if denied) 5. Order notification and file the complaint in the archive	City councilman	T37 - Decision on complaint (state)	Citizen, Lawyer, Architect, City councilman	City councilman
6. Injunction assessment as to whether or not legalize the construction by SAP	Architect	T32 - Technical analysis of complaint	Citizen	Architect
7. Order to formalize project	City councilman / Chief of the urban planning division	T37 - Decision on complaint (state)	Citizen, Lawyer, Architect, City councilman	City councilman
8. Order of embargo	City councilman	T37 - Decision on complaint (state)	Citizen, Lawyer, Architect, City councilman	City councilman

As can be observed in the first table, some of the specified organizational roles responsible for flowchart tasks are not the same as the executing DEMO organizational functions, revealing the ambiguity of the flowchart approach and the much more precise DEMO approach taking in account the existence of the initiator and the executing actor roles which helps a lot to clarify responsibilities. Looking at both tables, there were 9 DEMO transactions that were specified in our project and were missing in the flowcharts, especially in respect to the final part of the license grant process, namely transactions: T14, T20, T25, T26, T27, T28, T29, T33, and T35. Furthermore, we find that 7 flowchart tasks correspond to two or more DEMO transactions (1 corresponds to 4). This means that these flowchart tasks are, due to their ambiguity, indeed hiding at least one of two ontological and human acts in each task. So we can consider that, in average, around 9 other DEMO transactions were also missing in the flowcharts, amounting to around 18 transactions missing in the original contents. In the flowchart relating to the complaint process, we witness a different issue: 5 of the flowchart steps are either ontological transaction steps or infological or datalogical acts of the same transaction. This contributes to show the conciseness quality of DEMO thanks to the aggregation of several ontological and human process steps in one DEMO transaction, thanks to the transaction axiom, and also the power of abstraction from implementation given by DEMO's distinction axiom.

In total, for these two inter-related processes of license grant and complaint, we found, in the given documentation, 23 flowchart tasks spread over diagrams contained in 7 A4 pages, in 4 documents. Due to a lack of clear semantics of a flowchart approach, direct interpretation of these flowcharts was either not easy or not possible. These flowcharts were accompanied by descriptions of the tasks contained in 21 A4 pages, also in 4 documents. These allowed the interpretation of the whole process but in an incomplete way, as several process and responsibility details were missing (but found by applying DEMO). Such interpretation was difficult because the descriptions had many implementation details and also many complex references to several articles of National law. From the total of 28 pages of content in diagrams and descriptions we could not have a succinct and crisp global view of these processes.

By using DEMO, all this information, as well missing process information not written anywhere, was concisely summarized in a set of 38 transactions presented in 2 A4 pages, in the ATD (actors and transactions) view, or in 2.5 A4 pages, in the PSD (transactions process) view. Thanks to the clear semantics of DEMO and the natural devising of more precise and unambiguous names for the transactions and actor roles, the interpretation of DEMO's diagrams is much clearer and more precise than with the flowchart approach. If just looking at transaction and actor role names is not enough, one can look at our case description centered around the specified transactions that explains the meaning of all such transactions as well as process flow and interdependencies. This description occupies just 2.5 A4 pages

Taking in account that 18 transactions were missing, we can consider that around half of the ontological process was not precisely described in the flowcharts. This evidence clearly validates DEMO's quality of comprehensiveness, stated in [3] as implying that "all relevant issues are covered, that the whole is complete". The process may still not be fully and completely specified, but it is quite impressive that the DEMO approach allowed to discover a "hidden" half of the process and complete it with the other half. Moreover, if that half of the process would have been specified with the flowchart approach, with the same amount of detail the rest of the process was, we assume that we would have around 60 pages of content. By providing a view of the process in 2.5 pages of the PSD, plus 2.5 pages of description (i.e., 5 pages total), we manage to get a reduction of around 90% from the complexity of the original materials while still providing very comprehensive and complete information. This impressive reduction in complexity also strongly validates DEMO's quality of conciseness, stated in [3] as implying that "no superfluous matters are contained in it, that the whole is compact and succinct".

7 Conclusions

The results presented in the previous two sections help us to conclude that it is possible and necessary that complex cases like this are communicated to the scientific and practitioner communities so that widespread adoption of DEMO and Enterprise Engineering becomes a reality. The knowledge provided in the case description and associated models is, by itself, a valuable contribution to inspire similar initiatives. We furthermore present guidelines and patterns that we devised from our experience and may be generalized and reused in similar scenarios. Such re-utilization is one of the future lines of future work we envision. The validation we provide, based in our project's metrics is also an important contribution to bring more ground and inspiration for DEMO's application. It is, however, based in a single case and on the impressions of the authors. So another future line of research would be to apply similar metrics and analysis to other complex cases similar to this one. Our next step in this project will be the implementation of a Workflow

Management System supporting this process. After the implementation we intend to realize qualitative and also quantitative validations of some results of this paper and other interesting results we expect to achieve in this enterprise engineering and DEMO project.

References

1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Management Information Systems Quarterly 28, 75–106 (2004)
2. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, Heidelberg (2006)
3. Dietz, J.L.G.: Is it PHI TAO PSI or Bullshit? Presented at the Methodologies for Enterprise Engineering Symposium, Delft (2009)
4. Dietz, J.L.G.: On the Nature of Business Rules. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) CIAO! 2008 and EOMAS 2008. LNBIP, vol. 10, pp. 1–15. Springer, Heidelberg (2008)
5. Hicks, B.J.: Lean information management: Understanding and eliminating waste. International Journal of Information Management 27, 233–249 (2007)
6. White, S.A.: Process modeling notations and workflow patterns. In: Workflow Handbook, pp. 265–294
7. Aalst, W.M.P., van der Pesic, M., Schonenberg, H.: Declarative workflows: Balancing be-tween flexibility and support. Comp. Sci. Res. Dev. 23, 99–113 (2009)
8. Pesic, M., Schonenberg, H., van der Aalst, W.: Declarative Workflow. In: Hofstede, A.H.M., Aalst, W.M.P., Adams, M., Russell, N. (eds.) Modern Business Process Automation,
   pp. 175–201. Springer, Heidelberg (2010)
9. De Roover, W., Vanthienen, J.: On the Relation between Decision Structures, Tables and Processes. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2011 Workshops. LNCS, vol. 7046, pp. 591–598. Springer, Heidelberg (2011)

A Design Science Perspective on Business Strategy Modeling

Constantinos Giannoulis and Jelena Zdravkovic

Department of Computer and Systems Sciences,
Stockholm University
Forum-16440, Kista, Sweden
{constantinos,jelenaz}@dsv.su.se
http://dsv.su.se

Abstract. An important topic in the modeling for IS development concerns quality of obtained models, especially when these models are to be used in global scopes, or as references. So far, a number of model quality frameworks have been established to assess relevant criteria such as completeness, clarity, modularity, or generality. In this study we take a look at how a research process contributes to the characteristics of a model produced during that process. For example: what should be observed; what research methods should be selected and how should they be applied; what kind of results should be expected; how they should be evaluated, etc. We report a result on this concern by presenting how we applied *Design Science Research* to model business strategy.

Keywords: Business Strategy, Modeling, UBSMM, Design Science.

1 Introduction

The study of Information Technology (IT) utilization in organizations [22] is concerned with both the technological and social systems, as well as by phenomena emerging upon their interaction [31]. According to [45], an Information System (IS) encompasses the interaction of technological elements and people engaged to collect, filter, process, create, and distribute data. Hence, research within IT revolves around three related fields: Computer Science, concerned with development and code, Software Engineering, focused on production and operationalization of software, and Information Systems (IS), concerned with the use of IT in organizations facing managerial and organizational challenges [22].

Within IS, Design Science Research (DSR) is a problem-solving paradigm rooted in engineering; it aims to resolve distinct wicked problems by innovative artifacts through a development and evaluation circle against criteria of utility within an operating context (social setting, environment, domain, etc.) [34, 23]. DSR defines a process for building the constructs of the innovative artifact, such as models, methods and instantiations; the artifact itself and its use; as well as the environment within which the artifact is meant to be used for solving the addressed problem [23]. After the seminal publication of Hevner et al [23], design

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 424–438, 2014.

science research has been gaining wide acceptance as an research paradigm [21]. So far, the use of DSR has been reported in system modeling (e.g [53]), in enterprise modeling (e.g. [46]) and enterprise architecture (e.g. [35]).

When aiming to IS development to technically operationalize certain domain, data and data operations, models have always been fundamental [6]. System modeling entails the use of models to conceptualize a realm and build IS, where many modeling perspectives exist with respect to the IS aspects meant to be described (e.g. behavioral, functional, structural, etc.) [29]. Our research concerns business strategy modeling and integration into a unified business strategy meta-model for improving the alignment linkage between the *Business* and *IT*.

The objective of this paper is to present the experience and results of applying the DSR paradigm for the development of the Unified Business Strategy Meta-Model (UBSMM). In particular, we present the research process undertaken and reason over the methodological choices made to achieve the research goals set for addressing the alignment linkage using UBSMM.

Differences in research assumptions influence a series of concerns. For example: what should be observed; what kind of questions should be asked around the problem; how these questions should be structured; what methods should be selected and how should they be applied; what kind of results should be expected; how should these be analyzed and interpreted.

The paper is structured as follows: section 2 presents an overview of our business strategy modeling effort; section 3 discusses the research paradigm and philosophical assumptions underpinning the work; section 4 presents the research process followed for business strategy modeling using the Design Science Method [26] along with the methodological choices for the development of UBSMM; section 5 holds a reflective discussion on our outcomes, and section 6 concludes the paper along with some directions for future research.

2 Modeling Business Strategy: UBSMM

Business strategy is the determination of long-term objectives and courses of action using resources to achieve them [8]. Formulating business strategy provides the ways to timely change strategic thrusts and strategic capabilities [1].

Pervading all sectors of organizations, Information technology (IT) has become a fundamental factor for business strategy enactment. IT comprises the essential information needed to build the information systems (IS) to execute, support and facilitate business operations for delivering offerings to customers.

The continuous emergence of technological advancements necessitates more than ever before, alignment of *Business* and *IT*. Business strategy should be understood and communicated to define the means required for its successful execution, also making clear for IT what business stakeholders need. The alignment linkage between business strategy and IS is essential for the coordination of strategic initiatives with IS, to setup the infrastructure, design the processes, and define the capabilities required to support business operations [47].

Despite this acknowledged importance of aligning strategic initiatives and plans with IS, the linkage suffers from shortcomings of existing approaches making even more difficult to grasp any view of strategic initiatives and facilitate the development of relevant IT solutions. Business strategy is typically linked to IS in an abstract way [33] or established business strategy formulations are often overlooked. When used, the linkage is heavily natural-language based, thus dependent on the specificities of the business strategy formulations and the IS models employed.

Our proposal to address these shortcomings has lead to development of the Unified Business Strategy Meta-Model (UBSMM) [18, 20], which integrates business strategy formulations within Strategic Management into a meta-model that enables linking with IS through model-level mappings. Such a model-centric proposal leverages characteristics of Model-Driven Development (MDD) such traceability [2], and also allows for the propagation and assessment of IS features and/or changes towards business strategy. With respect to the aforementioned shortcomings of current approaches UBSMM addresses two primary challenges: the a) domain modeled and b) its coverage:

a) Due to the ambiguity of business strategy formulations, typically natural language-based and accompanied with brief schematic representations, they are also ambiguous when compared to IS models that are build with well-defined syntax and semantics. This constitutes business strategy open to interpretation hindering common understanding and the linkage to IS.
b) The second challenge concerns domain coverage as there exist different perspectives of business strategy, which results in different formulations driven by different types of business strategy logic. Barney [3] identified three types of strategy-shaping logic upon the concept of competition in microeconomics, which he considered complementary to each other: the resource-based type, the industrial organization type, and the Schumpeterian (innovation) type. Similarly, more groupings of strategy-shaping logic exist, such as Mintzbergs ten school of thoughts [37], synthesized by defining strategy with five complementary ways; as a plan, as a plot, as a pattern, as a position, and as a perspective (the five Ps) [38] as well as using other base disciplines (i.e. psychology, political sociology, anthropology, etc.).

Overcoming these challenges and building UBSMM is based on iteratively integrating the conceptualizations of business strategy formulations. The fist UBSMM version has been required to at least aggregate the three complementary types from Barneys classification: Strategy Maps and Balanced Scorecards (SMBSC) [27] as an example of the resource-based type, the Value Configuration (VC), which consists of the Value Chain [44], the Value Shop and the Value Network [49] as an example of the industrial organization type, and Blue Ocean Strategy (BOS) [28] as an example of the innovation type.

This selection of business strategy formulations is not exclusive, thus other perspectives of business strategy can also be added and integrated to UBSMM, such as the ones of Mintzberg [37]. as well as future emergent ones. Figure 1 presents UBSMM as an aggregation of business strategy formulations (SMBSC-MM, VC-MM, BOS-MM), including others than can also be integrated, which as indicated from the Business Strategy Formulation MM). The integration of any business strategy formulation to UBSMM requires its conceptualization to undergo a similar schema integration process followed as the existing ones [18].

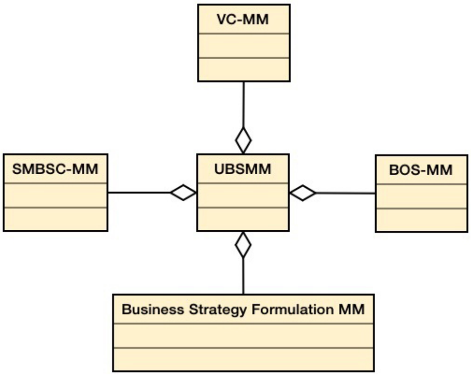


Fig. 1. UBSMM: an aggregation of Business Strategy formulation meta-models (MM)

3 Research Paradigm for Business Strategy Modeling

Typically, a research community shares a common set of beliefs and assumptions affecting the choice of research methods employed, namely a research paradigm, which shapes how its members perceive their discipline and consequently, how research methods are chosen [30, 26].

A research paradigm is characterized by philosophical assumptions expressed as concerns about reality (ontological), knowledge (epistemological), ways to examine reality for knowledge (methodological), and values (axiological). Within a discipline, these assumptions altogether position a researcher's belief system and view of the world towards the research problem being addressed, providing thus, rationale for the choice of the methods for actualizing the research process.

Ontological concerns focus on reality and the researcher's stance towards the nature of reality; what exists, what is derived [52] and [26]. Epistemological concerns focus on knowledge; how can people gain knowledge about the world, what does it depend on, how can one be sure of what they know [52, 26]. Methodological concerns focus on the appropriateness of the ways and procedures used to examine reality as well as the validity of the knowledge produced from them [52, 26]. Axiological concerns focus on people's values, collectively valuing what researchers hope to achieve and find, which makes a shared value system within a research community [52].

Research on the fit between strategy and IS models positions the research problem to the IS context. Vaishnavi and Kuechler name IS a *multi-paradigmatic community* [51], where different sets of practice define IS as a scientific discipline and researchers can take different stands following different paths when investigating IS research problems. Table 1 presents the dominant research paradigms within IS with respect to their philosophical assumptions.

Table 1. Dominant IS research paradigms

Paradigm	***Philosophical Assumptions***
Positivism [42]	**Ontology:** suggests that there exists a single reality regardless of people and their experiences. **Epistemology:** phenomena observed in the world can be explained through cause-effect relationships and are expected to embed explanation, prediction and control. Scientific knowledge allows for verification or falsification and the strive for generalizable results. **Methodology:** entails quantitative approaches aimed at providing objective and bias free knowledge. **Axiology:** entails striving for a universal truth supporting prediction of phenomena.
Interpretivism [42]	**Ontology:** argues that reality is constructed by people and their (inter)actions thus phenomena observed are dependent on their context along with people's subjectivity and through social interaction. **Epistemology:** truth is subjective with knowledge emerging from the active participation of the researcher in the phenomena investigated (social interaction). **Methodology:** qualitative approaches reinforce a participatory investigation of phenomena by engaging researchers in the social environment examined. **Axiology:** entails striving for understanding and describing including subjectivity acknowledgments affecting validity of results.
Social Constructivism [9]	**Ontology:** suggests that reality lies within the world people live and work, where subjective meanings of their experiences are developed. **Epistemology:** meanings are formed through interactions between people based on as many observers'/participants' views as possible of a situation examined, as well as through pre-existing norms and views. **Methodology:** entails participatory approaches to construct the meaning of a situation examined through social interaction, Focus is put on specific contexts where people operate to understand their historical and cultural settings. **Axiology:** focuses on making sense of meanings others have on a situation examined along with the researcher's own interpretation due to their background and experiences.
Pragmatism [9]	**Ontology:** suggests that truth is not bounded by any particular worldview or philosophy, rather what works for the situation examined. **Epistemology:** knowledge is gained based on examining the "what" and "how" with respect to the intended effects. **Methodology:** entails freedom of choice for multiple and mixed methods and techniques rather than subscribing to one, based on the needs of a situation examined. **Axiology:** suggests making sense of what works at the time and that is the truth.

Critical Realism [7]	**Ontology:** suggests the real world exists independently of our knowledge, beliefs, thoughts, perceptions etc. whether observable or not. **Epistemology:** knowledge is considered social and historical, where not all viewpoints must be equally valid, and exists in different types; physical, social, and conceptual. **Methodology:** entails a range of different research methods due to the different knowledge types and supports a mixed-methods research. **Axiology:** knowledge of reality is a result of social conditioning and, thus, cannot be understood independently of the social actors involved in the knowledge derivation process.

However, the emergence of DSR as a scientific study within IS has also emerged the idea of design science as a research paradigm [23, 52], though not widely accepted to cause a paradigm shift [30]. Nevertheless, for DSR, IS research paradigms can be combined in the same design science project, for example positivism and interpretivism [26].

This diverse utilization of research paradigms within design science is closer to the idea of a multi-methodological approach to IS research [41] or what is commonly refereed to as *pluralism*, which suggests that mixed method research designs are preferable to encompass real setting, social situations and research context [36]. Therefore, research paradigms with different philosophical assumptions can be utilized during each step of the research process influencing the selection of research methods employed [52, 26]. Particularly, ontological and epistemological views shift as a design science project progresses [52].

In the scope of this work, during the early steps of the research process the social constructivism perspective is relevant as it provides multiple reality experiences from multiple organization settings for the alignment linkage between business strategy and IS influencing both the practical implications of the problem as well as requirements put on the unified business strategy meta-model to be build. Moreover during the later steps of the process the positivist perspective becomes relevant as the unified business strategy meta-model becomes more stable and thus it is through observation that predictions can be made on the satisfaction of the requirements put on the artifact, which may lead to additional iterations of the design cycle. The pluralistic research paradigm followed in the development of UBSMM in the context of the alignment linkage between business strategy and IS is summarized in table 2 with respect to the philosophical groundings of *design science research* [52], influenced by [24].

Ontologically, design science research suggests that the state of reality is altered through the introduction of artifacts. However, there exists one single, stable underlying physical world whose laws constraint the various altered reality states during the artifacts' development. Epistemologically, knowledge is produced through the process of constructing and employing artifacts. Information on the artifact, its comprising components and their interactions, is considered true when artifacts behave as expected. Therefore, meanings are the utility provided and the functionality enabled with respect to the problem being addressed. Methodological concerns entail incremental artifact development and assessment with respect to the setting investigated. Axiologically, apart from the truth, researchers value control and creative adjustment of the setting investigated for the end result contributes to the body of knowledge with practical solutions or even partial and incomplete theories paving the way for further investigations.

Table 2. Business Strategy Modeling following a DSR paradigm

Basic Belief	***DSR [52]***	***Applied in UBSMM***
Ontology	Multiple, contextually situated alternative world-states. Socio-technologically enabled	Reality evolves as the alignment linkage is dependent on multiple alternative organizational settings as each organization is unique
Epistemology	*Knowing through making*: objectively constrained construction within a context. Iterative circumscription reveals meaning.	Knowing through making via iterative applications of the model-driven proposal revealing findings, which consequently lead into fine tuning of the proposal itself
Methodology	Developmental. Measure artifact impact on the composite system.	Reasoning through the design cycle actualizes the model-driven proposal for the alignment linkage in the development of a unified business strategy meta-model, whose impacts are assessed
Axiology	Control; creation progress (i.e. improvement); understanding.	Conceiving, incrementally creating and understanding the applicability of the unified business strategy meta-model in the context of the alignment linkage along with any socio-technological implications identified, constitutes valuable contribution

4 Business Strategy Modeling Using the Design Science Method

The scientific study and creation of artifacts in design science evolves iteratively and incrementally into a practical solution, through a generic design cycle [23]. Furthermore, the essential activities constituting a design science research project include: *explicating the problem; outlining the artifact and defining its design requirements; designing and developing the artifact; demonstrating; evaluating; and communicating the artifact* [23, 51, 22]. In our study we have adopted Johannesson and Perjons' *Design Science Method* (DSM) [26], which is a holistic problem solving approach through artifact development (Figure 2).

The DSM consists of an activity flow presented using IDEF0 (Figure 2), which is enriched with the research methods (upper part) and the knowledge base used for each activity (lower part). Therefore, for each activity, there exists some input which is transformed to some output using the knowledge base with respect to research methods. Similarly to the activities of other design research approaches [43, 51, 22], the activity flow of DSM is not temporal, rather it is based on input/output relationships between activities [26].

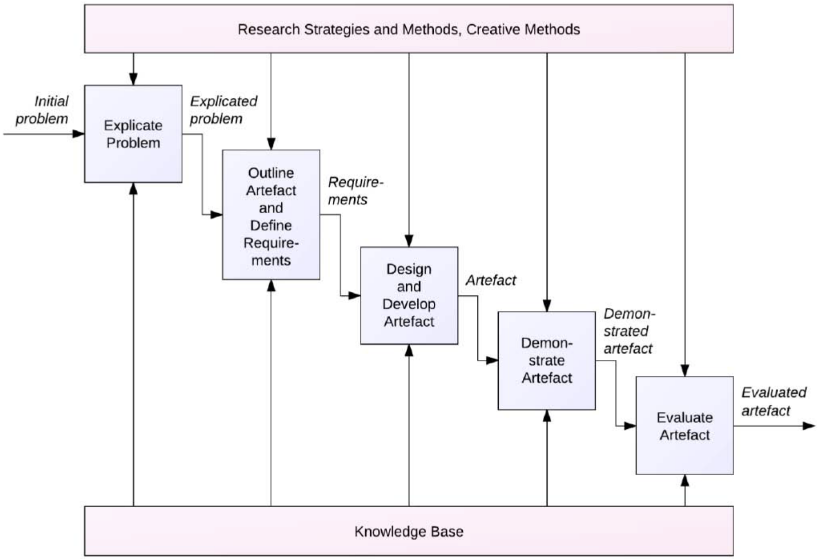


Fig. 2. The Design Science Method (adopted from [26])

4.1 Explicate Problem

The problem has been explicated through document studies as presented in section 2 as well as in [11–13, 10, 18] showing that business strategy is abstractly used when it comes to the linkage between strategic initiatives and IT solutions hindering alignment of the *Business* with *IT*. Resources used included literature addressing the overall problem of alignment, proposals addressing the alignment linkage and literature on types of IS models used.

In addition, an empirical study in the form of a self-administered online questionnaire targeting both business and IS practitioners has also been used to strengthen the problem identified and with an empirical basis [19, 17].

4.2 Outline Artifact and Define Requirements

The artifact is a unified business strategy meta-model (UBSMM) that integrates conceptualizations of business strategy formulations that can be mapped to IS models. It has been outlined based on literature and document studies of business strategy formulations reported in [11–13, 10, 15, 18, 16, 20], but also through theoretical analysis of usage scenarios for UBSMM [11, 18], through the aforementioned empirical study reporting on the use and acceptance of particular business strategy formulations as well as on the wide acceptance of different IS models by practitioners [19, 17]. These are presented as artifact requirements for UBSMM in Table 3 (Req. 1, Req. 2, Req. 5, and Req. 6).

Regarding model quality criteria, there exist varying perspectives (i.e. theory-based, experience-based, observation-based, consensus-based, and synthetical) resulting into many approaches, though no standard or consensus seems to exist as summarized in [39]. The selection of quality criteria has been based on the essential requirements of model correctness (Req. 4 in Table 3) and model completeness (Req. 3 in Table 3), as there exists empirical evidence suggesting they are the most influential factors of model quality for practitioners [40]. Document studies on schemata integration [4, 5] have also been used [18, 20].

Table 3. Artifact requirements for UBSMM

Req. 1	The business strategy formulations chosen to build UBSMM shall enable comprehensive coverage of business strategy with respect to Barney's types of strategy logic [3]; this will allow UBSMM to be linked with IS offering a comprehensive view on business strategy.
Req. 2	The integration of business strategy formulations shall follow a systematic process; this will allow for further enrichment and evolution of UBSMM to integrate emergent business strategy formulations in the future.
Req. 3	UBSMM shall be complete; this corresponds to model completeness with respect to the conceptualizations of business strategy formulations [32, 5, 48, 40], understandability [5, 40] and language adequacy [48].
Req. 4	UBSMM shall be correct; this corresponds to model correctness [5, 40], model validity [32], and model construction adequacy [48].
Req. 5	Each of the business strategy formulations integrated shall be derivable from UBSMM, which shall result into a conceptualization for each business strategy formulation in the form of a conceptual model; this allows for specializing UBSMM to conceptualizations for each business strategy formulation integrated, which consequently will allow instantiating the conceptualization into the business strategy of an organization.
Req. 6	Conceptualizations derived from UBSMM shall be mappable to IS models (i.e. RE, EM, and EA approaches), thus allow traceability of business strategy notions (objectives, intentions, etc.) to IS.

4.3 Design and Develop Artifact

Designing the artifact has been based on literature and document studies of the business strategy formulations that have been analyzed while outlining the artifact. Practical industrial applications of these formulations have also been considered. Conceptualizations for each business strategy formulation have been build using UML class diagrams. Moreover, literature in conceptual modeling and schemata integration has been used to define a development process for the artifact. The development process of UBSMM entails distinct phases that include from selecting business strategy formulations and building their conceptualizations to their integration into UBSMM. Based on the foundational work of [4] and [5] the four phases adopted are:

1. *Pre-Integration*; schemata to be integrated are selected and an integration strategy is decided.
2. *Schemata Comparison*; schemata are analyzed and compared for correspondences, conflicts and inter-schema properties.
3. *Schemata Conformance*; resolutions for conflicts are defined and modeling decisions are made upon correspondences and inter-schema-properties.
4. *Schemata Merging and Restructuring*; conflict resolutions are applied along with restructuring resulting into one schema.

During pre-integration, business strategy formulation schemata were selected, their conceptualizations were built as UML class diagrams, accompanied with constraints [12, 13, 16]. Following a binary strategy for the integration process, which allows for progressive and gradual unification of business strategy formulations [4, 5], UBSMM was built in two steps. The first step included integration of meta-models for SMBSC and VC, where all succeeding phases of the integration process were carried out resulting into a first version of UBSMM as presented in [18]. In a similar manner, the second step included integration of the derived first UBSMM version and the BOS meta-model, also following the succeeding phases of the integration process as discussed in [20]. This order of preference was based on literature indicating SMBSC and VC are well-established [10], also supported by results of empirical studies [19, 17].

For both steps, schemata were analyzed and compared to identify correspondences between concepts across business strategy formulations, naming conflicts and structural conflicts, as well as inter-schema properties [4, 5]. During schemata conformance, semantic relationships between concepts were identified with respect to conflicts, correspondences and inter-schema properties and resolutions were decided (i.e identical, equivalent, compatible and incompatible[4]). Finally, during the last phase, the conformed schemata were merged and restructuring occurred to accommodate conformance of resolutions into one schema.

The implementation of all phases is presented in [18] for SMBSC and VC resulting into a first version of UBSMM and again in [20] for the integration of BOS, which resulted into a complete UBSMM.

4.4 Demonstrate Artifact

Once developed, artifacts are used in instances of the problem they have been built to address [43]. Therefore, each business strategy formulation integrated to UBSMM has been demonstrated through experimentation, which included instantiating their conceptualizations using real world published applications. This entailed using the meta-models built for each business strategy formulations and a strategy from real published cases, as well as mappings to IS models used for system requirements.

For the former, the strategy map template, the value shop, and the strategy canvas have been used along with the original publications of the formulations [27], [49], and [28] respectively, which have been reported in [12], [13], and [16].

For the latter, mappings for the conceptualizations of Strategy Maps and Balanced Scorecards and Blue Ocean Strategy towards i* have been instantiated, which allowed the derivation of i* models from business strategy [15].

Experimentation with the aforementioned applications demonstrated that UBSMM integrated business strategy formulations reducing the risk of incorporating variances due to misinterpretation and also allowed for their mapping IS models such as i* which is used in requirements engineering.

4.5 Evaluate Artifact

UBSMM has been evaluated with respect to the requirements defined in section 4.2, as summarized in table 4. For *Req. 1* and *5* theoretical analysis has been used to build informed arguments for their satisfaction. For *Req. 3* and *4* experiments have been used to report on their satisfaction. Whereas for *Req. 2* and *6*, both experiments and theoretical analysis have been used.

Table 4. Requirements evaluation for UBSMM

Req. 1	Business strategy literature from strategic management has been analyzed and informed arguments have been built for using SMBSC, VC, and BOS. The reasoning that supports this argument is based on the construction of the artifact [26]. UBSMM has been constructed based on the conceptualizations of business strategy formulations that are representative of the three types of strategy shaping logic suggested in [3]. Thus, providing comprehensive coverage of business strategy notions [11, 18, 20, 17].
Req. 2	The schema integration process adopted is well-documented and allows the continuous and integral integration of more business strategy formulations to UBSMM in a systematic manner [18, 20].
Req.3&4	Experiments using real-world published cases have been conducted; ABB Industrie AG for SMBSC [12], the Norwegian police for VC[13], and Southwest Airlines for BOS[16]. Additional experiments have been conducted for SMBSC involving the real strategy map for education in a Swedish higher education institute [14], as well as the use of the SMBSC meta-model to capture consumer values for a shopping mall [50]. Concepts from the original business strategy formulations have been modeled and instantiated using the aforementioned cases. For SMBSC and VC, the meta-models have been implemented in semantic languages such as OWL creating instances with respect to the cases modeled but also to allow for formal evaluation of concepts and associations modeled. Model constraints were also formalized and model constructs were instantiated one by one.
Req. 5	Constraints defined for UBSMM allowed to derive conceptualizations of each of the three integrated business strategy formulations in the form of a conceptual models [20].
Req. 6	Experiments have been used for mappings to IS models used towards RE [15, 14], while informed arguments have been built for mappings to IS models towards EA [18] and EM [20].

Overall, with respect to generic DSR criteria: validity, utility, quality and efficacy [23, 21], experiments have shown that UBSMM functions as intended; it captures business strategy for each of the three business strategy formulations integrated and can be conceptually related to RE, EA, and EM approaches through mappings. In terms of utility, the experimental application of UBSMM in [15, 50, 14] has shown how to establish a bidirectional linkage between business strategy and the IS requirements' model derived. In terms of quality, UBSMM has fulfilled requirements on completeness and correctness (Req. 3 and 4) but it has also shown ease of understanding through the experiment in [14], which is relevant to pragmatic quality in [29]. The idea of using such models for establishing and strengthening the alignment linkage has been positively received by practitioners [19, 17], which is indicative of the approach' efficacy.

5 Discussion

The foundation of the adopted DSR paradigm has been used to guide the production and communication of a new knowledge artifact that is relevant for a global practice. Creation of generalizable knowledge has further required the use of rigorous research strategies and methods along the research process. As very important, the applied Design Science Method does not prescribe a sequential way of working. The activities (see Figure 2) are logical and not temporal groupings of work, i.d. as explained in section 4. The relationships between activities are solely of the input-output type, hence, the development process is iterative, capable of absorbing complex and changing requirements for the artifact, both directly as well as through changing environment. Moreover,the DSM implemented in this work is consistent with characteristics of both the artifact and process of other DSR strategies, as in [25].

During the research process multifold uses of UBSMM have emerged: a) iterative integration of the conceptualizations of existing and future strategy formulations to facilitate formal mappings to IS models, b) a reference model to synchronize or integrate business strategies across business of an organization, or of the partners in a multi-organizational constellation; c) a single point for mapping to IS models practiced across various business units/organizations; d) a pivot model for organizations to assess their business strategy considering a different type of strategy-shaping logic (resource-based, competition-based, innovation-based), or to explore potential strategic shifts, for example from resource- to innovation-based considering implications on IS.

As for limitations of the used research paradigm and the process, an obvious one is a lack of the techniques and the tools to support development of the artifact as it is the case with system development tools. Another limitation concerns the extent of evaluation of the artifact DSR does not offer prescriptions on how to evaluate artifacts differentiating in terms of their scope, adoption time, way, duration and change of use, etc. Hence the evaluation of UBSMM is currently limited, as indicated in section 4.

6 Concluding Remarks

In this study we have presented the application of the DSR paradigm followed for modeling business strategy in the development of the Unified Business Strategy Meta-model (UBSMM). Within the scope of our proposal the selection of the DSR paradigm was motivated with respect to the philosophical assumptions underpinning business strategy modeling and the need for a pluralistic paradigm.

The outcome of our work can serve as prescriptive knowledge for future business strategy modeling efforts. It puts forward a set of requirements for business strategy modeling addressing domain coverage (Req.1 and 5), progressive evolution through integration (Req. 2), model quality (Req. 3 and 4), and linkage to IS (Req. 6).

At the same time, the design science perspective followed and applied contributes to the body of knowledge with a set of paradigmatic research assumptions for business strategy modeling including ontological, epistemological, methodological and axiological assumptions. Such differences in research assumptions influence a series of concerns that frame the research agenda: what is it to be observed, what questions shall be asked around the problem and how, what methods should be selected and how should they be applied, what kind of results should be expected, how should these be analyzed and interpreted.

References

1. Ansoff, H.I.: Corporate strategy: Business policy for growth and expansion. McGraw-Hill Book (1965)
2. Atkinson, C., Kuhne, T.: Model-driven development: A metamodeling foundation. IEEE Software 20(5), 36–41 (2003)
3. Barney, J.B.: Types of competition and the theory of strategy: Toward an integrative framework. Academy of management review 11(4), 791–800 (1986)
4. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys (CSUR) 18(4), 323–364 (1986)
5. Boman, M., Bubenko Jr., J.A., Johannesson, P., Wangler, B.: Conceptual modelling. Prentice-Hall, Inc. (1997)
6. Bubenko Jr., J.A.: From information algebra to enterprise modelling and ontologies—A historical perspective on modelling for information systems. In: Conceptual Modelling in Information Systems Engineering, pp. 1–18. Springer (2007)
7. Carlsson, S.A.: The potential of critical realism in is research. In: Information Systems Theory, pp. 281–304. Springer (2012)
8. Chandler, A.D.: Strategy and structure, vol. 4. MIT press, Cambridge (1962)
9. Creswell, J.W.: Qualitative inquiry and research design: Choosing among five approaches. Sage (2012)
10. Giannoulis, C.: Modeling Business Strategy for Business-IT Alignment. Licentiate, Stockholm University (2011)
11. Giannoulis, C., Petit, M., Zdravkovic, J.: Towards a unified business strategy language: A meta-model of strategy maps. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 205–216. Springer, Heidelberg (2010)

12. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling business strategy: A meta-model of strategy maps and balanced scorecards. In: IEEE 2011 5th International Conference on Research Challenges in Information Science (RCIS), pp. 1–6 (2011)
13. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling competition-driven business strategy for business it alignment. In: Salinesi, C., Pastor, O. (eds.) CAiSE Workshops 2011. LNBIP, vol. 83, pp. 16–28. Springer, Heidelberg (2011)
14. Giannoulis, C., Svee, E.O., Zdravkovic, J.: Capturing consumer preference in system requirements through business strategy. International Journal of Information System Modeling and Design (IJISMD) 4(4) (2013)
15. Giannoulis, C., Zdravkovic, J.: Modeling strategy maps and balanced scorecards using i. In: iStar, pp. 90–95 (2011)
16. Giannoulis, C., Zdravkovic, J.: Linking strategic innovation to requirements: A look into blue ocean strategy. In: Proceedings of Practice of Enterprise Modeling (PoEM 2012), CEUR, vol. 933, pp. 118–128 (2012)
17. Giannoulis, C., Zdravkovic, J.: Model-centric strategy2is linkage: Model-centric strategy2is linkage: An empirical study. In: 16th IEEE Conference on Business Informatics (submitted, 2014)
18. Giannoulis, C., Zdravkovic, J., Petit, M.: Model-driven strategic awareness: From a unified business strategy meta-model (UBSMM) to enterprise architecture. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 255–269. Springer, Heidelberg (2012)
19. Giannoulis, C., Zdravkovic, J., Petit, M.: Model-centric strategy-it alignment: An empirical study in progress. In: Franch, X., Soffer, P. (eds.) CAiSE Workshops 2013. LNBIP, vol. 148, pp. 146–155. Springer, Heidelberg (2013)
20. Giannoulis, C., Zikra, I., Bergholtz, M., Zdravkovic, J., Stirna, J., Johannesson, P.: A comparative analysis of enterprise modeling approaches for modeling business strategy. In: Short Papers in Proceedings of Practice of Enterprise Modeling (PoEM 2013), CEUR (2013)
21. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS Quarterly 37(2), 337–355 (2013)
22. Hevner, A., Chatterjee, S.: Design Research in Information Systems: Theory and Practice. Integrated Series in Information Systems, vol. 22. Springer (2010)
23. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
24. Iivari, J.: A paradigmatic analysis of information systems as a design science. Scandinavian Journal of Information Systems 19(2) (2007)
25. Iivari, J.: Distinguishing and contrasting two strategies for design science research. European Journal of Information Systems (2014)
26. Johannesson, P., Perjons, E.: A Design Science Primer. CreateSpace (2012)
27. Kaplan, R., Norton, D.: Strategy maps: Converting intangible assets into tangible outcomes. Harvard Business Press (2004)
28. Kim, W.C., Mauborgne, R.: Blue Ocean Strategy. Harvard Business Press (2004)
29. Krogstie, J.: Model-based development and evolution of information systems: A Quality Approach. Springer (2012)
30. Kuhn, T.S.: The structure of scientific revolutions (50th anniversary edition: 2012) edn. University of Chicago press (1962)
31. Lee, A.: Editorial. MIS Quarterly 25(1), iii–vii (2001)
32. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE Software 11(2), 42–49 (1994)

33. Luftman, J., Kempaiah, R., Nash, E.: Key issues for it executives 2004. MIS Quarterly Executive 4(2), 269–285 (2005)
34. March, S., Smith, G.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
35. Meyer, M., Helfert, M., Donnellan, B., Kenneally, J.: Applying design science research for enterprise architecture business value assessments. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 108–121. Springer, Heidelberg (2012)
36. Mingers, J.: Combining is research methods: Towards a pluralist methodology. Information Systems Research 12(3), 240–259 (2001)
37. Mintzberg, H., Ahlstrand, B., Lampel, J.: Strategy Safari: A Guided Tour Through The Wilds of Strategic Mangament. The Free Press (1998)
38. Mintzberg, H.: The strategy concept i: Five ps for strategy. California Management Review 3, 11–23 (1987)
39. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. Data & Knowledge Engineering 55(3), 243–276 (2005)
40. Moody, D.L., Shanks, G.G.: Improving the quality of data models: Empirical validation of a quality management framework. Information Systems 28(6), 619–650 (2003)
41. Nunamaker Jr, J.F., Chen, M.: Systems development in information systems research. In: Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences, 1990, vol. 3, pp. 631–640. IEEE (1990)
42. Orlikowski, W.J., Baroudi, J.J.: Studying information technology in organizations: Research approaches and assumptions. IS Research 2(1), 1–28 (1991)
43. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
44. Porter, M.E.: Competitive advantage: Creating and sustaining superior performance. Free Press (1985)
45. Pyster, A., Olwell, D. (eds.): Guide to the Systems Engineering Body of Knowledge (SEBoK), version 1.1.2. The Trustees of the Stevens Institute of Technology, Hoboken (2013)
46. Rittgen, P.: Collaborative modeling–A design science approach (2009)
47. Ross, J.W., Weill, P., Robertson, D.C.: Enterprise architecture as strategy: Creating a foundation for business execution. Harvard Business Press (2006)
48. Schuette, R., Rotthowe, T.: The guidelines of modeling–An approach to enhance the quality in information models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 240–254. Springer, Heidelberg (1998)
49. Stabell, C.B., Fjeldstad, Ø.D.: Configuring value for competitive advantage: On chains, shops, and networks. Strategic Management Journal 19(5), 413–437 (1998)
50. 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)
51. Vaishnavi, V.K., Kuechler Jr., W.: Design science research methods and patterns: Innovating information and communication technology. CRC Press (2007)
52. Vaisvani, V., Kuechler, J.W.: Design research in information systems (2004)
53. Wieringa, R., Heerkens, H.: Design science, engineering science and requirements engineering. In: 16th IEEE International Requirements Engineering, RE 2008, pp. 310–313 (2008)

Automated Enterprise-Level Analysis of ArchiMate Models

Andres Ramos, Paola Gomez, Mario Sánchez, and Jorge Villalobos

Systems and Computing Engineering Department
Universidad de los Andes, Bogotá Colombia
{am.ramos260,pa.gomez398,
mar-san1,jvillalo}@uniandes.edu.co

Abstract. Around the world, Enterprise Architecture (EA) practices are been formed in large and medium companies that see in IT either a competitive advantage or a requirement for survival. These EA practices produce models that conceptualize the enterprise, and are commonly used only for communication purposes. Using these models also for analysis purposes is desirable, but this is hard to do because of the complexity and size of these models. Automated analysis tools seem to be adequate mechanisms to solve this issue, but currently there is a problem of mismatch between the information available in the models, and the information that the automated mechanisms require. To address this, this paper proposes a characterization of analysis functions, which makes explicit the information that each one requires to be executable (among other things). Furthermore, the paper presents *ArchiAnalysis,* an extensible tool for applying analysis functions over ArchiMate models.

Keywords: Enterprise Architecture, Enterprise Modeling, Automated Analysis, ArchiMate, Relations, Analysis tools.

1 Introduction

Enterprise modeling languages can be classified as predictive, descriptive, or a mixture of both [15,16]. Predictive languages are used for estimating future states of the reality; descriptive languages are used for acquiring understanding of said reality. Typically, these are graphical languages because this facilitates communication and increases understanding [7]. A prominent example of this is ArchiMate [2], a modeling graphical language for enterprise architecture that has recently started to become a de-facto standard for the practice. In the enterprise environment, enterprise architects commonly use ArchiMate and other similar languages to document and communicate the enterprise state. This is called Enterprise Modeling (EM) and it is a prior step to model-based analysis, which supports diagnosis, optimization, and decision-making processes across many areas of an organization [3], [6,7,8].

The current problem that has been identified is that tools based on ArchiMate and similar notations are focused more on the visualization and modeling aspects than in the analysis of the resulting models. Therefore, analysts have to perform their jobs in

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 439–453, 2014.

a mostly manual fashion, and only get technological support from tools that are meant for different purposes, such as spreadsheets' processors. To further complicate matters, the size and complexity of enterprise models increases every day. Progressively, enterprise models have become more detailed and have incorporated other domains within a company, and thus have more elements and relationships. Using views and viewpoints [9,10] as a strategy to lower complexity is a common practice, but it encloses the risk of losing the holistic, global view of the enterprise which is so highly praised in the enterprise architecture practice.

To solve this problem, automated analysis mechanisms should be used. However, automated mechanisms have an important limitation: they are not flexible with respect to the information that they require. If some information (attributes in an element or relation, relations, or elements) is not present, the mechanism will not work. Furthermore, different analysis methods will have different information requirements, which may not be part of modeling languages. Therefore, a two-part strategy is necessary: on the one hand, it is necessary to make explicit the information requirements ("*semantic conditions*" [1]) of each automated analysis method; on the other hand, it is necessary to have tools that support the extension and specialization of modeling languages, including the addition of attributes on elements and relations [2], [3], [11].

The work presented in this paper has two parts. Firstly, there is a characterization of analysis function embodied in the analysis domains used by Lankhorst [8]. This characterization can be used to describe and guide the design of analysis functions with the ultimate goal of automating them. Furthermore, this characterization is guiding the creation of a catalog of analysis functions, which is briefly presented in the paper. Secondly, the paper presents *ArchiAnalysis*, an extensible tool built on top of Archi [12] to enables the automated execution of analysis functions over ArchiMate models. Conceptually, the creation of additional analysis functions should follow the guidelines posed by the characterization.

The rest of the paper is structured as follows. Section 2 discusses the concept of automated model analysis, presents the characterization of analysis functions, and illustrates it with two fully defined functions. Then, Section 3 presents *ArchiAnalysis* from a high-level point of view, which is followed by a showcase of its capacities applied to the ArchiSurance [13] case study. Finally, Section 5 discusses some of the relevant related work, and Section 6 concludes the paper.

2 Analysis

Analysis transforms *mere facts* into *reasoned facts* in order to provide the information needed to solve or to resolve a problem using solid arguments [4]. Particularly, in the EA context, analysis processes extract relevant information from the enterprise models in order to provide solid, and structured information that will be valuable for making business and IT decisions.

Figure 1 depicts the normal flow of a human-based analysis process in this context. The analysis process starts with the identification of *what* is the problem to solve or the concern to address. Next, using the EA model and other relevant information of the enterprise, an analyst defines the process to obtain the information that IT and Business decision-makers need. This analysis process is cyclic: analysts interact repeatedly with the EA model, querying and enriching it, until an acceptable result is obtained [7]. Finally, the process produces outcomes such as findings, recommendations, or assessments that satisfy the initial analysis requirements.

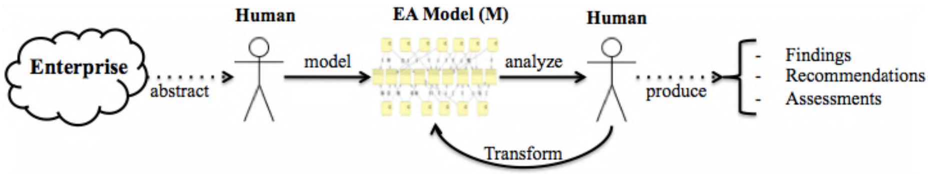


Fig. 1. human-based analysis process

2.1 Automated Analysis

Nowadays, the analysis problem resides in identifying methods and models that can transform data into reliable and comprehensible knowledge [5]. Automated model analysis is described as the process of extracting and manipulating data contained in the model, using automated mechanisms [14], to support decision-making processes. In the EA context, automated analysis transforms the model and creates additional information that enriches the model. Then, the analysts use this information to produce more accurate results [18].

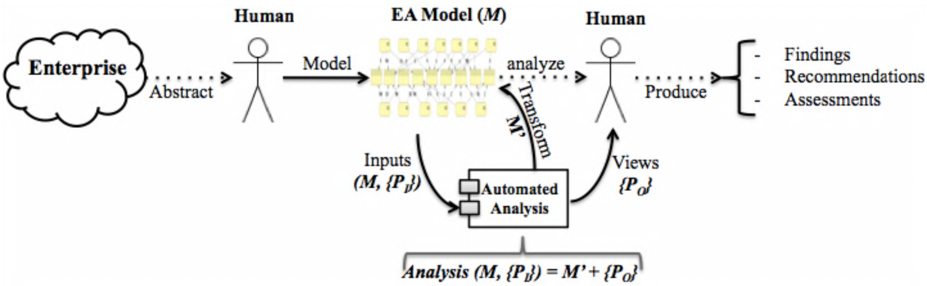


Fig. 2. human-based analysis supported with an automated analysis tool

Figure 2 illustrates automated analysis and highlights its similarities and differences with the purely manual process. In addition, figure 2 depicts an *Analysis function* structure, which is the backbone of the present work. In the figure, an EA model M, additional model attributes and context variables P_I are the inputs of some *Analysis function*. Then, the *Analysis function* processes the EA model and produces: 1) an enriched model M' and 2) a set of new data P_O.

It is important to consider that automated analysis depends on the modeling language, and how it presents the model information. An example of this is illustrated in Figure 3. Figure 3a shows a model where *Policy Data Management* and *Claims Data*

Management systems use a *Financial Application* service. Two relevant questions that this model should answer are: 1) Are the *Policy Data Management* and *Claims Data Management* systems able to consume the *Financial Application* service via SOAP web services? 2) The *Financial Application* service supports all the transactions per minute requested from the *Policy Data Management* and *Claims Data Management* systems? To answer these questions, it is evident that the model should contain more specific information. In Figure 3b, the same model was complemented with some additional information, the supported protocols of each application, which corresponds with the P_I of the *Analysis function* presented in Figure 2. Now, with this inputs, the *Analysis function* process could automatically answer question number 1. Figure 3c presents the same model, this time enriched with information regarding the volume of transactions supported by each application, which was added to answer the second question.

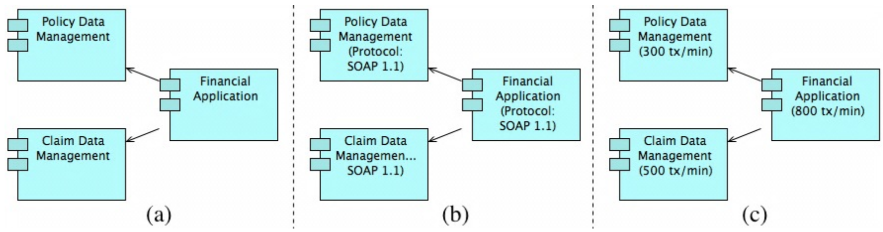


Fig. 3. information required for perform automated analysis of an enterprise model

In this small example, it can be seen that even simple analyses require model languages with mechanisms to manage and change attributes at will. Furthermore, it can be seen that the selection of attributes should not depend on the language selected to model the reality, but on the analyses and questions that the model should help to answer.

2.2 Analysis Function Characterization

In order to support the aforementioned point, it is necessary to know which information is required by each analysis function. If said information is found, the function can be applied. Otherwise, a different function should be selected, or the model should be enriched with missing information. A problem that we have identified is that analysis functions or methods often lack an explicit specification of the information and structures that they require to work. Thus, it is not easy to realize if a model is suited or not to support a particular analysis function, which in turn limits the possibilities for having automated analysis.

To address this issue, we now propose a structure to characterize analysis functions. This structure requires, for each analysis function, the following bits of information: *name*, *description*, *dimension*, *type*, *layer*, *entities and relations*, *structural attributes* and *algorithm*.

Name refers to the unique name of an analysis function, which must be unique.

Description is a textual description of the analysis function, including the scope of the analysis, the context where this analysis could be performed, and the stakeholders that may require it.

Dimension (inspired by Lankhorst [8]) creates a function classification into four groups: quantitative, functional, analytical or simulation. The first dimension includes functions that answer quantitative questions. Functions in the second dimension are used to determine whether or not a model meets with a functional requirement, or for validating correctness of the model. There are two approaches of functional analysis: impact of change or gap analysis. The simulation dimension is related with a model execution where the inputs could be established dynamically, and the results could change every time the model is executed. Finally, the analytical dimension refers to functions that deliver unique and reproducible results providing to the architect a first insight of the architecture behavior.

The element *Type* characterizes the analysis functions with respect to the concern they address. Table 1 presents a list of some analysis types and the dimensions where they have relevance.

Table 1. Analysis functions types

Dimension	Type	Dimension	Type
Quantitative	Performance	**Functional (Structural / Dynamical)**	Impact of change
	Optimization		Alignment
	Impact of change		Coherence
	Capacity planning		Correctness
	Cost		Conformance
	Availability		Gap
	Trade-off		Graph structure
	HR		Counting
			Process

The *Layer* element of the characterization structure is inspired on ArchiMate's layers. Each analysis function can be classified with respect to the *Layer*, or *Layers*, that it addresses: Business, Application, Technology, Motivation, and Implementation and Migration.

With respect to the problem of automating model analysis, the most important element of the characterization structure is the one called *Entities and relations.* In this apart, for each function it is necessary to describe the information it requires to be present in the model (mandatory entities and relations between them). This characterization element is very important because the information provided can be used to guarantee that the model has the information required to successfully apply the analysis function.

Similarly, the *Structural attributes* element describes the attributes that an analysis function requires on each entity type and relation.

Finally, the *Algorithm* element describes how the analysis function extracts and processes information from a model in order to obtain some analysis results.

Table 2. Analysis function catalog sample

	Type	Id	Name
Quantitative	Performance	QPR001	Infrastructure services workload
		QPR002	Application services processing and response time
	Impact of change	QIC001	Measure impact of modify (increase/decrease) infrastructure resources
	Capacity planning	QCP001	Estimate solution storage volumetric
		QCP002	Estimate solution network volumetric
	HR	QHR001	Human Resource workload at business process level
Functional	Impact of change	FIC001	Remove an architecture component
	Alignment	FAG001	Business-Application Alignment
		FAG002	Business-Technology Alignment
	Coherence	FCH001	Every business active structure has at less one direct/derived assignment
	Correctness	FCO001	Data security compliance at transport level
	Process	FPR001	Data/Information vs. Application
		FPR002	Process responsibility assignment

2.3 A Catalog of Analysis Functions

Based on the available bibliography, as well as the study of existing tools, we are currently compiling a catalog of analysis functions. To structure this catalog, each analysis function is characterized, and the combination of its *Dimension*, *Type* and *Layer* is being used to determine its placement on the catalog. Currently, we have identified over one hundred analysis functions to classify.[1] Table 2 presents an extract composed by 13 analysis functions with their respective *Name*, *Id*, *Type* and *Dimension*. The simulation and analytical *Dimension* are not currently strongly represented in our catalog because behavioral analyses are not the current focus of our work.

We now present two analysis functions fully characterized to illustrate how information is presented in the catalog. These analysis functions are *Business-Application Alignment* (FAG001) and *Human Resource workload at business process level* (QHR001). Table 3 shows the function FAG001. It is based on the specific types of some relations, and on specialized ArchiMate elements.

[1] http://backus1.uniandes.edu.co/~enar/dokuwiki/doku.php?id=archianalysis#analysis_function_catalog

Table 3. Analysis function FAG001 - Business-Application Alignment

ID: FAG001 **Dimension:** Functional **Type:** Alignment
Name: Business-Application Alignment **Layer:** Business/Application

Description: This function covers a particular analysis for the EA Requirement of Business and IT Alignment. Through it, it is possible to appreciate whether each IT service of the organization (at an application level not technological) supports at least one-business service/capability.
Due to the high density of services (Business and IT) in an organization, this analysis function should be considered over the entire architecture enterprise model.

Entities and relations: Three main relationships types between Business and Application must be considered:

1. *Used by relation*: a) between application service and the different types of business behavior elements, and b) between application interface and business role.
2. *Realization relation*: from a data object to a business object in order to indicate that business object has an electronic representation corresponding to a data object.
3. *Assignment relation*: a) between application component and business process, function, or interaction, and b) between application interface and business service in order to indicate that, for example, a business processes or business services are completely automated. In case that a business process, function, or interaction is not completely automated but only supported by an application component is expressed with a *Used by relationship*.

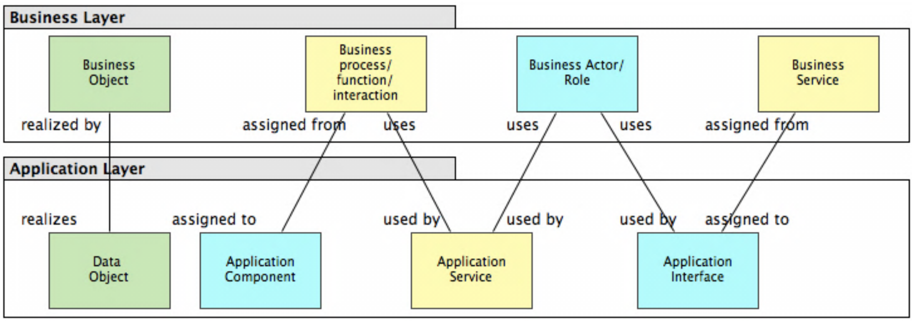


Fig. 4 Business-Application Alignment

Structural Attributes: No structural attributes are required for this analysis, only the entities types and the three relations mentioned in the previous section are required:

— Relations (R): Used by (u_R), Realization (r_R), Assignment (a_R)
— Business elements (B): Business object (o_B), Business process/function/interaction (b_B), Business actor/role (a_B), Business service (s_B)
— Application elements (A): Data object (o_A), Application component (c_A), Application service (s_A), Application Interface (i_A)

Algorithm: The first step is to create a function able to calculate the derived relation *tdr* which applies between an application element *A* and a business element *B*, *tdr(a,b)*. The result of this function will be the type of the derived relation (u_R, r_R or a_R), or *none* whether there is not any derived relation that complies the required structural chain. Then, the function must evaluate the following analyses rules and establish whether each application component complies with all requirements given bellow; otherwise, it is determined as a "*misaligned*" application layer element:

- Every application service s_A must be used by (u_R) at less one business behavioral element s_B.
- Every application interface i_A must be used by (u_R) at less one business role a_B
- Every Data object o_A must realises (r_R) a Business object o_B
- Every application component c_A must be assigned to (a_R) or used by (u_R) a Business process/function/interaction b_B

The second analysis function to be treated in this section is presented in Table 4. This function (QHR001) presents a case where it is necessary the *Business Process* specialization into a new entity named *Business Sub-process* in order to get different information from each entity. Therefore, this function requires the model *M* and the additional information P_I which is provided by each model element whose can be different because depending on it type. The outcome of this function, as will be illustrated in Section 4, is a modified model *M'* and a set of new information P_O calculated as part of the function algorithm.

Table 4. Analysis function QHR001 - Human Resource workload at business process level

ID: QHR001	**Dimension:** Quantitative	**Type:** HR
Name: Human Resource workload at business process level		**Layer:** Business

Description: This analytical method aims identify human resources (employees) that are overloaded or idle according to the responsibilities assigned to any process, the contribution percentage and the involvement percentage.

The result of this analysis permits to organize resources allocation, involvement percentage, re-assign responsibilities and take other decisions around the human resource capacity.

The output of the analysis will be described in the following scale:

- 100%< workload — Overwork
- 100%>= workload >80% — Appropriate
- 80%>= workload >50% — Review
- 50%>= workload > 0% — Idle

Entities and relations: To describe this method clearly, we assume that any *Business Process* aggregated into other *Business Process* will be called *Business Sub-process*, and these will be the behavioral unit where an active structure element can performs a behavior. This function will treated and define the active structure element in a Business Role level. Through decision will be possible have enough detail to calculate the workload of any human resource assigned to this Role (whether more detail is required, the entire organizational structure need to be modeled and actors would be the active structure unit)

In terms of organization structure, the *Business Actors* will be treated as *Organizational Units* instead of persons assigned to a *Business Role*. Finally, this function considers that a Business Role could be assigned to one or many Sub-processes. In this case, each assignment must have an involvement % and a contribution %.

Structural Attributes:

- For any business process P, a Daily arriving Instances $f_{P.}$
- For any business sup-process S, a % of arriving instances l_S, (after an split caused by a junction the % of instances arriving to each sub-process will be divided e.g. 50% of the claims are paid and the other 50% are rejected). And the average daily time spent per each instance t_S expressed in minutes.
- For any business role R, a number of employees n_R are assigned to this role.
- For any assignment relation A, a % of involvement i_A of the role in the sub-process. And a % of contribution c_A of the role in the sub-process (a sub-process must be full filed by one or many roles and each one plays a specific function)

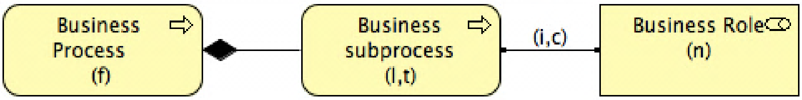


Fig. 5. Human Resource workload

Algorithm: As this function is classified as a quantitative analysis function, it performs some calculations based on 8 hours as a working day:

- it_A (hours): Daily involvement hours one employee. $it_A = i_A \, x \, d$
- ir_A (hours): Daily involvement hours per role. $ir_A = it_A \, x \, n_R$
- ns_S: # of instances arriving per sub-process. $ns_S = f_P \, x \, l_S$
- bt_S (hours): Total hours per daily instances batch. $bt_S = ns_S \, x \, t_S$
- cr_A (hours): Contribution hours of the role per sub-process. $cr_A = bt_S \, x \, c_A$
- y_A: % workload x role & sub-process. $y_A = ir_A \, x \, cr_A$

3 Implementation of Automated Analysis Functions

The characterization of analysis functions presented in the previous section is a necessary step in the direction of supporting automated analysis functions. However, to support the actual implementation of the algorithms enclosed in every analysis function, it is necessary to have some base framework to serve as a starting point. Otherwise, implementing a single analysis function would be so costly and be unfeasible. Probably, it would be even more expensive than performing the analysis manually.

To address this issue, we designed a conceptual framework to support analysis functions over enterprise models. For validation purposes, this framework was implemented on top of Archi [12], a well-known and open source editor for ArchiMate models, and was tested on the ArchiSurance scenario that is used to showcase ArchiMate in the specification of the language. Nevertheless, the framework is generic, in the sense that it does not dependent on a specific metamodel, and that it does not depend on a specific technology. Its only technological requirement is having the possibility of adding attributes to elements and relations, even when they are not part of the original metamodel.

The implementation of the framework on top of Archi is called *ArchiAnalysis* and it was implemented as an Eclipse plug-in [21]. Since ArchiAnalysis was specifically made for Archi, it relies on the model and diagram management technology underlying Archi (EMF - Eclipse Modeling Framework [17], and GEF - Graphical Editing Framework [22]). Additionally, *ArchiAnalysis* relies on Eclipse's extension mechanisms to support the configuration of any number of analysis functions. Therefore, the development of a new analysis function only requires the implementation of the actual processing code (using some libraries provided by the framework), the compilation and encapsulation of the code, and the deployment of the package as an Eclipse plug-in. The analysis functions described previously have already been implemented in *ArchiAnalysis* using the procedure that was just described.

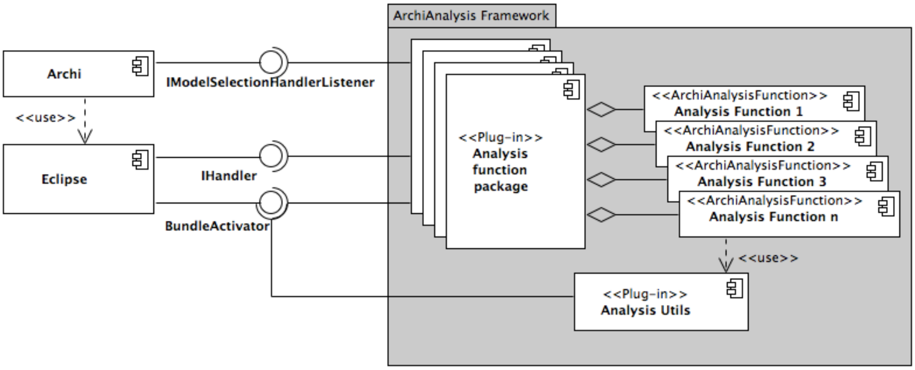
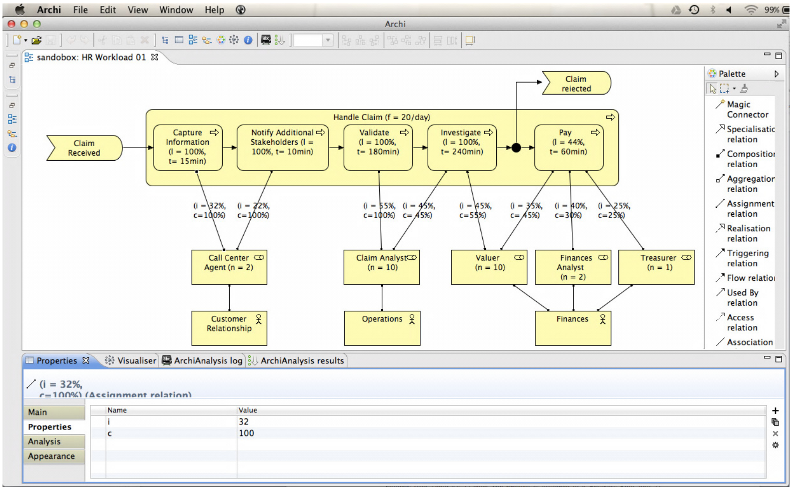


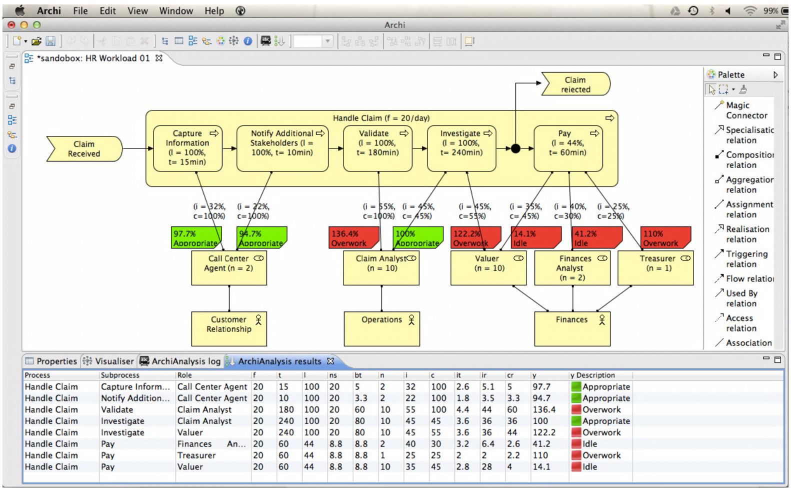
Fig. 6. ArchiAnalysis framework components model

Figure 6 shows a high level view of *ArchiAnalysis* structure. The right hand side of the figure shows the framework and its two main components: *ArchiAnalysisFunction* and *Analysis Utils*. The *ArchiAnalysisFunction* is an abstract component that must be specialized by each analysis function in order to define the way to connect with Archi models. In addition, this component uses the *Analysis Utils* which offers several operations for model manipulation. Each concrete *ArchiAnalysisFunction* is implemented and deployed as an independent Eclipse plug-in.

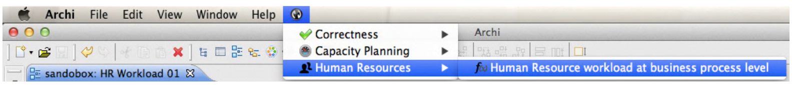
The left side of the figure shows Archi and Eclipse. Archi exposes the current working model through the interface *IModelSelectionHandlerListener*. Similarly, Eclipse exposes the *IHandler* interface to be consumed by each analysis function in order to gain user interface control. As a result, these set of interfaces provided to *ArchiAnalysis* the possibility to get the model and its attributes in order to process them according with the analysis function defined and deliver results transforming the model or deploying data in different views.



(a) Model before analysis execution



(b) Model after analysis execution



(c) ArchiAnalysis menu inside Archi tool bar

Fig. 7. ArchiSurance Handle Claim Process and ArchiAnalysis execution screenshots

4 ArchiAnalysis at Use: QHR001 Applied to ArchiSurance

As an example of ArchiAnalysis at use, we present the implementation of *ArchiAnalysis Function* QHR001 discussed in Section 2.3 (Table 4). This function, called *Human Resource workload at business process level*, identifies the employees that are overloaded or idle based on their responsibilities over some processes, their contribution percentage, and their involvement percentage. This function was applied on the ArchiSurance case study that comes pre-loaded in Archi.

In the first place, Figure 7.a shows the *Handle Claim* process, which was modeled with Archi, already enriched with the additional information that the analysis function requires (see Table 4): 1) each *Sup-process* is *assigned* to a *Business Role,* and 2) each element has some properties set according to the next table.

Table 5 Handle Claim Process - properties values for sub-processes and roles

Sub-process (*S*)		f_P	t_S	l_S
S_1	Capture Information	20	15	100%
S_2	Notify Additional Stakeholders	20	10	100%
S_3	Validate	20	180	100%
S_4	Investigate	20	240	100%
S_5	Pay	20	60	44%

Business Role (R)		n_R
R_1	Call Center Agent	2
R_2	Claim Analyst	10
R_3	Valuer	10
R_4	Finances Analyst	2
R_5	Treasurer	1

The results of applying the analysis functions can be graphically presented in the Archi canvas. Figure 7.b illustrates this: 1) the % of workload of each relation is set as an attribute, and 2) a different color is related in order to distinguish between the possible outputs (red for *Overwork* or Idle, yellow for *Review* and green for *Appropriate* workload).

ArchiAnalysis can also deliver results as a spreadsheet. Figure 8 shows the results given by the analysis function QHR001. These results are based on the business role and sub-process assignment where all attributes, intermediate variables, and the final result of the analysis are presented.

Properties | Visualiser | ArchiAnalysis log | ArchiAnalysis results

Process	Subprocess	Role	f	t	l	ns	bt	n	i	c	it	ir	cr	y	y Description
Handle Claim	Capture Inform...	Call Center Agent	20	15	100	20	5	2	32	100	2.6	5.1	5	97.7	Appropriate
Handle Claim	Notify Addition...	Call Center Agent	20	10	100	20	3.3	2	22	100	1.8	3.5	3.3	94.7	Appropriate
Handle Claim	Validate	Claim Analyst	20	180	100	20	60	10	55	100	4.4	44	60	136.4	Overwork
Handle Claim	Investigate	Claim Analyst	20	240	100	20	80	10	45	45	3.6	36	36	100	Appropriate
Handle Claim	Investigate	Valuer	20	240	100	20	80	10	45	55	3.6	36	44	122.2	Overwork
Handle Claim	Pay	Finances An...	20	60	44	8.8	8.8	2	40	30	3.2	6.4	2.6	41.2	Idle
Handle Claim	Pay	Treasurer	20	60	44	8.8	8.8	1	25	25	2	2	2.2	110	Overwork
Handle Claim	Pay	Valuer	20	60	44	8.8	8.8	10	35	45	2.8	28	4	14.1	Idle

Fig. 8. Analysis function AM004: Human Resource workload results

Finally, Figure 7.c shows the exclusive *ArchiAnalysis* menu that displays the analysis functions currently deployed. The selection of one function in the menu validates if the model accomplishes with the required semantic conditions and then executes the analysis.

5 Related Work

In this section we briefly present some previous works that are related to our own.. In the first place, Lankhorst [8] and Iacob [24] propose in their approaches the incorporation of information in all model elements including entities and relations for analysis purposes. They perform quantitative analysis of ArchiMate models and use attributes in the entities and relations as input of their analysis function. In our proposal, we use the attributes in a similar way; nevertheless, we define a structured way to establish the necessary model information to support automated analysis functions.

In the second place, relating to automated analysis, Benavides [14], [23] and Sunkle [20] explore about the automated analysis over enterprise-level models. They use the relations as structural or topological elements. In contrast, our analysis method uses relations in a dynamic way as data source elements in the analysis execution.

Related to the use of enterprise models as decision-making support instruments, more than just communication tools, Johnson [15], Sunkle [19] and Kohlhammer [5] propose interesting works. Johnson [15] proposes P^2AMF as Predictive, Probabilistic Architecture Modeling Framework which is used to predict the properties of the system-to-be in order to take better design decisions during design phase. Sunkle [19] who argues that enterprise models are required for describe the enterprise as well as prescribe courses of action in the face of change, presents an approach that uses specialized models focused on decision making. Kohlhammer [5] proposes to combine visualization techniques and analytic algorithms to enable human experts to guide the decision making process.

Finally, regarding to automated-analysis tools, we will compare against two close tools to our own. Johnson [6] presents a tool for analysis of enterprise architecture scenarios. This tool guides the development of enterprise architecture models and provides a quality measure of the modeled architecture. Naranjo [7] proposes PRIMROSe, which is a visual-analysis framework and tool. PRIMROSe suggests an Enterprise Models analysis made with non-destructive functions that select and decorate an analytical abstraction.

6 Conclusions

This paper has addressed the problem of performing automated enterprise analysis and the need for frameworks and approaches to support said analysis using the information contained in enterprise models. The main obstacles that we have found for supporting that automation are the lack of precise definitions about the information requirements of each analysis function.

To address this issue, this paper proposes a characterization of analysis functions, which specifically requires a specification of the information that each one needs. Furthermore, this characterization also requires a specification of the algorithms that should support each analysis function. With this information it should be possible to build tools that automate the analysis functions. To validate the characterization, we are currently building a catalog of *Analysis functions*.

On the other hand, we presented *ArchiAnalysis* as a tool and framework where it is possible to implement and run automated analysis functions. This tool, which was built on top of Archi, provides the means to perform automated analysis on top of ArchiMate models, and also serves to validate the proposed characterization of the functions.

References

1. Harel, D., Rumpe, B.: Modeling Languages: Syntax, Semantics and All That Stu. (2000)
2. The Open Group.: ArchiMate 2.1 Specification. Van Haren Publishing (2013)
3. Karsai, G., Nordstrom, G., Ledeczi, A., Sztipanovits, J.: Specifying graphical modeling systems using constraint-based meta models. In: IEEE International Symposium on Computer-Aided Control System Design, CACSD 2000, pp. 89–94. IEEE (2000)
4. Byrne, P.H.: Analysis and science in Aristotle. SUNY Press (1997)
5. Kohlhammer, J., May, T., Hoffmann, M.: Visual analytics for the strategic decision making process. In: GeoSpatial Visual Analytics, pp. 299–310. Springer, Netherlands (2009)
6. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A tool for enterprise architecture analysis. In: 11th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2007, p. 142. IEEE (2007)
7. Naranjo, D., Sánchez, M., Villalobos, J.: PRIMROSe - A Tool for Enterprise Architecture Analysis and Diagnosis. In: 16th International Conference on Enterprise Information Systems, Lisbon, Portugal (2014)
8. Lankhorst, M.: Enterprise architecture at work: Modelling, communication and analysis. Springer (2013)
9. Jen, L.R., Lee, Y.J.: Working group. IEEE recommended practice for architectural description of software-intensive systems. In: IEEE Architecture (2000)
10. Rozanski, N., Woods, E.: Software systems architecture: Working with stakeholders using viewpoints and perspectives. Addison-Wesley (2011)
11. Cardelli, L., Matthes, F., Abadi, M.: Extensible Grammars for Language Specialization. In: DBPL, pp. 11-31 (1993)
12. Archi, `http://www.archimatetool.com`
13. Jonkers, H., Band, I., Quartel, D.: The ArchiSurance Case Study. White paper. The Open Group (Spring 2012)
14. Benavides, D., Segura, S., Ruiz-Cortés, A.: Automated analysis of feature models 20 years later: A literature review. Information Systems 35(6), 615–636 (2010)
15. Johnson, P., Ullberg, J., Buschle, M., Franke, U., Shahzad, K.: P^2AMF: Predictive, Probabilistic Architecture Modeling Framework. In: van Sinderen, M., Oude Luttighuis, P., Folmer, E., Bosems, S. (eds.) IWEI 2013. LNBIP, vol. 144, pp. 104–117. Springer, Heidelberg (2013)
16. Basili, V., Briand, L., Condon, S., Kim, Y.M., Melo, W.L., Valett, J.D.: Understanding and predicting the process of software maintenance release. In: Proceedings of the 18th International Conference on Software Engineering, pp. 464–474. IEEE Computer Society (1996)
17. Steinberg, D., Budinsky, F., Merks, E., Paternostro, M.: EMF:Eclipse Modeling Framework, 2nd edn. Addison-Wesley (2009)
18. Biehl, M.: Literature study on model transformations. Royal Institute of Technology, Tech. Rep. ISRN/KTH/MMK (2010)

19. Sunkle, S., Kulkarni, V., Rathod, H.: (Multi-) Modeling Enterprises for Better Decisions (2013)
20. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Analyzing Enterprise Models Using Enterprise Architecture-based Ontology. In: Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P. (eds.) MODELS 2013. LNCS, vol. 8107, pp. 622–638. Springer, Heidelberg (2013)
21. Clayberg, E., Rubel, D.: Eclipse Plugins, 3rd edn. Addison-Wesley (2008)
22. Rubel, D., Wren, J., Clayberg, E.: The Eclipse Graphical Editing Framework (GEF). Addison-Wesley Professional (2011)
23. Benavides, D., Trinidad, P., Ruiz-Cortés, A.: Automated reasoning on feature models. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 491–503. Springer, Heidelberg (2005)
24. Iacob, M.E., Jonkers, H.: Quantitative analysis of enterprise architectures. In: Interoperability of Enterprise Software and Applications, pp. 239–252. Springer, London (2006)

A Conceptual Framework for Time Distortion Analysis in Method Components

Fredrik Karlsson[1], Fredrik Linander[1,2], and Fabian von Schéele[1,3]

[1] CERIS, Department of Informatics, Örebro University School of Business, Örebro, Sweden
{fredrik.karlsson,fredrik.linander,fabian.vonscheele}@oru.se
[2] Saab Dynamics AB, Karlskoga, Sweden
fredrik.linander@saabgroup.com
[3] Linnaues University, Växjö, Sweden
fabian.vonscheele@lnu.se

Abstract. The "software crisis" is still a prevailing problem to many organizations despite existence of advanced systems engineering methods, techniques for project planning and method engineering; systems engineering project still struggle to deliver on time and budget, and with sufficient quality. Existing research stresses that time leakage has a lever effect on economic outcome, which is not addressed in the abovementioned approaches. As part of an on-going research project we therefore extend existing method engineering concept to include time distortion analysis. This allows for analysis of resource use (productivity) in execution of method components. It has the potential to act as a) a tool for improving the execution of systems engineering processes, or b) criteria for selecting method parts to improve the systems engineering processes.

Keywords: Method components, method engineering, time distortion, systems engineering method.

1 Introduction

In all business operations, productivity is a key performance in order to stay competitive and to reach the goal of sustainable profitability. This implies control of a combination of performance measures such as quality, delivery, finance and personnel, i.e. system engineering process management. Traditionally most companies developing complex system – systems that embed software into hardware – have focused their improvement work on productivity and measurements in the area of physical (operational) processes. Not much work has been focused on the systems engineering or administrative (transactional) processes or to the interaction between the transactional and operational processes. Furthermore, little attention has been put to problems related to time, such as time distortion in processes, or to the mechanisms between econometry, time distortion and process management. However, earlier research [e.g. 1] has shown the coupling influence of transactional processes on operational processes. Also, it has been demonstrated [2] that there are strong lever mechanisms between time distortion and economic key ratios. This is especially interesting as well as

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 454–463, 2014.

challenging, since the Standish Group CHAOS report [e.g. 3] showed that the software is an ever-increasing part of industry products. Simultaneously, the same report [3] also presented statistics concerning a severe deterioration in the efficiency of projects that involve systems engineering.

Consequently, it may be suggested that transactional processes, such as systems engineering, increase in importance, while the precision of measuring and managing them still are at loss. Notwithstanding project management tools, modern advanced systems engineering methods [e.g. 4, 5, 6], situational method engineering methods [e.g. 7, 8, 9] and computerized-aided method engineering tools [e.g. 10, 11] it is evident that the efforts indicate an aggravation of target achievement in projects with respect to project time schedule. Furthermore, time distortion and its lever effect on economic key ratios [2] is still not a part of the systems engineering management approaches previously mentioned. Consequently, what was once coined as the "software crisis" [12] in the late 1960's is still a prevailing problem to many organizations.

Against this backdrop we elaborate on a conceptual framework that combines situational method engineering concepts and time distortion analysis techniques to enable time leakage analysis in systems engineering of complex systems. The development of the conceptual framework is part of a three-year action research project aiming at increasing the efficiency of the systems engineering processes in industry. The overall aim of the research project is to develop a method and a computer-based decision support system for process management based on time distortion analysis.

The rest of the paper is structured as follows. In the next section we take a closer look at situational method engineering in general, and method components in particular. The third section outlines the research design. In the fourth section we present the conceptual framework. The fifth section contains a short illustrative example. Finally, we end the paper with a concluding discussion where we address implications for research and practice as well as future research.

2 Related Research

State-of-the art research on systems engineering process management can be divided into, at least, three schools, that largely pursue their own agendas without many cross-references. First there is the method engineering research [13], which focuses on the construction of systems engineering methods, and has contributed extensively to today's wisdom on such methods. Brinkkemper [14] has defined method engineering as the 'discipline to design, construct and adapt methods, techniques and tools for the development of information systems.' Furthermore, in recent years a subfield of method engineering has been established that focuses on situational method engineering [15]. The work in this sub-field has attracted much attention since it nowadays is an established fact that there is no such thing as a one-size-fits all method [16].

In order to design, construct or adapt software engineering methods much effort has been invested into understanding the concept of systems engineering methods. Several techniques for modelling of methods have been introduced, such as method fragments [9], method chunks [17], method components [18], process components

[15], and GOPRR [10]. Many of these techniques are very similar [19]. However, one difference is that some of these techniques, such as GOPRR, have been designed specifically for automating methods into computer-aided software engineering, while others have been designed to create situational systems engineering methods without turning them into computerized tools [20]. The strength of method engineering research is its attention to how methods are structured. Less attention has been devoted to how systems engineering methods are actually used in practice, and how these techniques can be used to reengineer existing systems engineering processes, to identify areas of process improvements.

The “method-in-action research” focuses specifically on how methods are used, or rather enacted, in practice [e.g. 20, 21, 22]. This school emphasizes personal aspects and studies how methods are used to enact personal beliefs and goals and they have contributed extensively to our understanding of method use. However, they often miss the difficulties involving how a method can be managed and adapted. Instead they are more interested in finding possible problems with actual method usage and they give less attention to if the method really is consistent with what they want to achieve in the long run. The results from this school often describe problems during method usage. Hence, these theories offer limited practical support when it comes to selection of method parts, i.e. that is when to decide whether or not to implement a ‘best practice’.

The third school is software process improvement, which aims to assess systems engineering organizations' capabilities to work with systems engineering. The field of software process improvement has delivered a number of important tools for this kind of work, such as the CMM [23], CMMI [24], and ISO/IEC 15504 [25]. These types of models focus on assessments of organisations’ capabilities with regard to maturity steps, for example going from a managed process to a predictable process. Hence, much attention is given to identify generic process attributes that can be evaluated on a scale of achievement. In addition, to reach certain maturity levels, such as Level 4 or above by CMMI quantitative process management must be used. Of course, in the extensive research made, many examples of metrics can be found such as defect rate, defect counts, project productivity and schedule adherence [e.g. 26, 27, 28]. Nevertheless, no approach focusing on time distortion, as suggested by von Schéele and Haftor [2], has been investigated. This model presents a metric that influences processes, projects and economy in a curve linear way. While the passage of time itself is considered as being a linear parameter, the error in time assessment, which is the time distortion, can be demonstrated to influence economy in a curve linear way. Thus, errors in processes, projects, and economy are calculated with respect to the curve linear mechanisms, in order so support improved predictability of target outcome.

3 Research Design

The conceptual framework presented in this paper is the result from an on-going action research project. The framework constitutes an important building block in reaching the overall project aim, to develop a method and a computer-based decision support system for systems engineering process management; it will act as the blueprint for implementing the method and the computerized tool. The research project is a collaborative project between Örebro University, Linnaeus University, Saab Dynamics and BAE Systems.

The research method adopted can be described as multigrounded [29, 30] action research (MGAR). MGAR consists of an interplay between three grounding processes: 'internal grounding,' 'external grounding,' and 'empirical grounding.' Internal grounding means reconstructing and articulating a priori knowledge and defining the concepts used and their interrelationships. The important contribution of this process is the conceptual framework presented in this paper. This model shall be free from ambiguities and with concepts that are anchored in explicit design goals. External grounding is concerned with relationships between the developed knowledge and other knowledge of a theoretical character. This means building the conceptual framework on existing wisdom about method engineering and time distortion analysis in process management and that we do not contradict relevant previous studies. Empirical grounding emphasizes the importance of applying the proposed design in practice to validate the concepts and their relationships. In our case, this involves gaining experience from future use of the conceptual framework in action cases [31].

Our implementation of MGAR follows the traditional 'canonical' action research method, which has cycles of diagnosing, action planning, action taking, evaluating, and specifying learning [32]. The project will comprise a number of such action research cycles where action cases [31] will be performed at Saab Dynamics and BAE Systems. An action case involves competent practitioners in collaborative design and evaluation efforts. Problems are analysed and design decisions taken by researchers and practitioners together to improve the design of the conceptual framework, and the future method and a computer-based decision support system for systems engineering process management.

The research results, presented in this paper, focus on the internal and external grounding of the project. It covers the diagnosing and action planning of the first MGAR cycle.

4 The Conceptual Framework

The conceptual framework builds on the method component concept and time distortion analysis. A systems engineering method, and hence an engineering process, consist of a set of method components. A method component is, according to Karlsson and Wistrand [18] "a self-contained part of a systems development method expressing the transformation of one or several artefacts into a defined target artefact and the rationale for such a transformation."

A method component consists of method elements of five different types: concept, notation, artefact, action, and actor role. Hence, the method element itself is an abstract class. The most central method element is the artefact, since there is always one deliverable from each method component. Moreover, artefacts are what connect method components; artefacts are consumed during actions that produce the deliverable(s).

Each action tells project members what tasks to perform during a project. During these actions a set of concepts is needed to describe the problem domain (and of the method itself). For instance, class diagrams use concepts such as class, attribute, and association. Hence, the concepts direct the engineers' attention to certain aspects of the problem domain while other parts are placed in the background. Third, the results are captured and represented using notation, such as Unified Modelling Language. Fourth, actor roles describe who are to carry out or participate in the actions that the method component describes. For example, that a requirements engineer has the responsibility to gather requirements from the users.

The method component includes an additional two classes – goals and values – that describe the rationale of the component. Method elements in a method component are included for reasons; these reasons are inherited from the method of which the method component is part. Goals describe what can be achieved with a method element, and are anchored in the values of the method creator; values tell why these goals are important to achieve.

Of the method elements in a method component, actions are the ones related to time; they consume time. Thus, it is where process efficiency can be measured in order to trace time distortion. Consider here shortly the formalisation of the concept time distortion [2]. In general terms, the definition of time distortion is set to be the ratio between the cognitive, or psychic, time (tc) and the physical, or clock, time (tp). Consequently, cognitive time distortion may be formalized in the following way:

$$\text{Cognitive time distortion} = tc \,/\, tp$$

In this, "tc" is the mental, or cognitive, time assessment made by an individual human being while "tp" is the physical time of the corresponding time duration, as measured by a clock. In appraising time distortion, it is required that the psychic and the physical time have the same frame of reference, and that they address the same event. Thus, "frame of reference" and "event" signify in our case the sum of actions in a method component. Time distortion is a relative measure, and can be interpreted as the difference, here the error, that occurs when a human individual assesses a time-duration in relation to its corresponding physical time. Therefore more specifically, time distortion, here denoted as "τi", is defined here as the ratio between the psychic time, "tc", and the physical time, "tp", of a certain event "i", hence formally:

$$\text{Time distortion: } \tau i = (tc \,/\, tp)\, i \qquad (1)$$

Specifically, for time distortion in a process, " time for the sum of actions "i" in a certain method component, while "tc" denotes the cognitive time assessment of the same actions. From the definition in equation (1) follows that time distortion, "τi" is limited by [0…L], where "L" is a large number, and that a value of "τi" corresponding to unity signifies the total conformity between psychic and physical time. Thus, a $\tau i = 1$ signifies a perfect compliance between the cognitive and physical time in activity "i". Anticipating the forthcoming elaboration below, we wish to highlight that the nature of this correspondence constitutes a key erroneous assumption made within present managerial and economic conceptions.

As this discussion shows, it is important to acknowledge the fact that time distortion is measured as the deviation from the planned time. This means that we need to trace both the plan and the execution. Method components, as described in existing research [e.g. 7, 11, 18], are used to structure existing methods as described in textbooks or as documented in organizations. Consequently, it is a plan and can be compared with "espoused theory" [33] – an ideal established by the organization's method creators "to explain or justify a given pattern of activity". Time distortion occurs due to deviations from the espoused theory, either from how the method component should be carried out or from an incorrect estimate of the time needed to complete the method component. Hence, for each project there exists one or several instantiations of the method components that may deviate from the espoused theory. In order to enable time distortion analysis, it is necessary to acknowledge that method components also exists as "theory-in-use", or "the performance of that pattern of activity" [33]. Conceptually it means an extension of the method component concept; it means that method components exist both as espoused theory and as several instances of theory-in-use.

Of course, the other method elements are not unrelated to time. They have an indirect affect on time consumption. The use of different sets of concepts and notation affect the time used, as well as the actor that are involved (or excluded) during a set of actions. Hence, these method elements play an important part in explaining time distortion.

5 A Theoretical Example

Productivity is an output/input measure informing about degree of change in a certain variable. The variable can be referred to in terms of hours, monetary units (MU), kilos etc. Consider now a simple systems engineering process with sequential linked method components as in Table 1 below.

Table 1. Example of time distortion in fictive systems engineering process

Espoused theory		Theory-in-use				
Method component	Planned time (h)	Method component	Actual time (h)	Difference (h)	Time distortion	ε
1. User stories	10	1. User stories	15	5	1,5	0,45
2. Prioritize User stories	5	2. Prioritize User stories	4	-1	0,8	0,23
3. Estimate User stories	4	3. Estimate User stories	4	0	1	0,18
4. Iteration Plan	3	4. Iteration Plan	2	-1	0,66	0,14
Sum	22		25	3	1,13	1

In this example, we consider costs as well as revenues linked to a predefined *fixed* budget. As the fifth column in Table 1 shows, the method components differ in planned time (espoused theory) and actual time (theory-in-use) and we wish to express this difference in terms of changed productivity. A common logic is to argue that there has been an increase in the labour time (with 3 hours) to accomplish the planned target of the process, while this does not apply to the revenues (the fixed price mechanism). Assume a price per hour to customer corresponding to 1000 MU/hour, and a salary cost per hour corresponding to 600 MU/hour. Example Calculus A:

Input: Profit = 1000 * 22 – 600 * 22 = 8 800

Output: Profit = 1000 * 22 – 600 * 25 = 7 000

Productivity: 7000/8800 = 0,80

Now, applying the time distortion on the same problem, we refer to von Schéele and Haftor [2] and their elaboration on time distortion and profit. Consider the same process as before, and assume that time distortion is identical on costs and on revenues. Set the lever effect of the time distortion to correspond to that of a fixed price contract (inverted mechanism). The time distorted profit can, with reference to von Schéele and Haftor [2], be written as:

$$\pi(\tau) = p\, t_{vol} \Sigma \big((\varepsilon/\tau) - (v_p\, \varepsilon / \tau)\big)_i \qquad (2)$$

Here, "$\pi(\tau)$" is time distorted profit, "p" is price per hour, "t_{vol}" is total budgeted time of process, "ε"$_i$ stands for planned fraction of time of action "i" in the method components (part of the systems engineering process), and "v_p" stands for the relative difference between salary costs per hour and customer price per hour. Applying equation (1) for the time distortion and use the figures in Table 1 gives example Calculus B:

Input Profit: $\pi(\tau)$ = 1000 * 22 $\big($(0,45/1 – 0,6 * 0,45/1) + (0,23/1 – 0,6 * 0,23/1) + (0,18/1 – 0,6 * 0,18) + (0,14/1 – 0,6 * 0,14/1)$\big)$

Input Profit: $\pi(\tau)$ = 1000 * 22 * 0,4 = 8 800

Output Profit: $\pi(\tau)$ = 1000 * 22 * $\big($(0,45/1,5 – 0,6 * 0,45/1,5) + (0,23/0,8 – 0,6 * 0,23/0,8) + (0,18/1 – 0,6 * 0,18/1) + (0,14/0,66 – 0,6 * 0,14/0,66)$\big)$

Output Profit: $\pi(\tau)$ = 1000 * 22 *0,392 = 8 620

Productivity: 8 620/8 800 = 0,98

The conclusion from this illustrative example is: time distortion analysis indicates that the productivity decreases with only somewhat 2% to 0,98. How can Calculus A differ so much from Calculus B? First, the time distortion analysis of method components does not treat the time gain on Method component 2 and 4 in Table 1 above, to level off in the same way as was assumed in Calculus A. In Calculus A, it is assumed that there is a *linear* trade off between the disparate method components; for example, that one hour lost in in method component may be recaptured in another method component. Instead, time distortion analysis makes corrections of each method component with respect to their relative weight (ε_i) and their relative *non-linear* error in time distortion. Second, time distortion analysis considers *both* revenues and costs when analysing time distortion. Thus, the perspective is a holistic one, which is frequently argued for in systems science. Third, time distortion analysis opens up for assessing the productivity of hours as output or input. In that case, there is a pronounced lever effect of the productivity – a counter intuitive outcome not yet accounted for in economy. Finally, the here demonstrated analysis is partly similar to time-driven activity based costing [34], however, the time distortion analysis acknowledge lever effects of time due to contractual mode.

6 Concluding Discussion

Existing research on systems engineering process management does not provide support for time distortion analysis. The aim of this paper was therefore to elaborate on a conceptual framework that combines method engineering concepts and time distortion analysis techniques to enable time leakage analysis in systems engineering of complex systems.

6.1 Implications for Research and Practice

The proposed framework contributes to existing research on systems engineering process management, which has so far paid little attention to time distortion analysis. Our framework is a direct extension of the method component concept [18]; it means a contribution to this theory. However, as Ågerfalk et al. [19] argued, many of the existing method engineering concepts that are used to structure systems engineering methods are very similar. Hence, our attempt to extend the method component concept is important because if this extension proves useful in practice it is possible to make similar extensions to other method engineering concepts, such as method fragments [9], method chunks [17], and process components [15]. The framework also complement software process improvement research, such as CMM [23], CMMI [24], and ISO/IEC 15504 [25]. The framework can be used to measure an organisations improvement on using specific method components; measurements on time distortions can be created for the method components, and can be compared across projects.

The implications for practice are so far limited since the conceptual framework has not been evaluated in real project settings. However, the framework has the potential to be a useful tool for tracing time distortion in method components and act as a staring point for either a) improving the execution of specific method components, or b) as criteria for selecting other method components to improve the systems engineering process.

6.2 Future Research

The conceptual framework opens up future research opportunities. First we would like to evaluate the framework as a vehicle for systems engineering process management.

This would suggest using the framework to analyse time distortion in systems engineering processes in order to assess its usefulness for the process management. We plan to carry out this assessment together with Saab Dynamics and BAE Systems in the on-going action research project. Second, we want to compare the time distortion results from this industry with analysis of systems engineering processes in other types of systems engineering organisations. This would not only give us interesting results concerning the use of the framework, but also comparative results concerning similar method components, which would be of interest as benchmarks for the participating companies. Third, we want to develop a computerized-tool support for this kind of analysis, since we suspect that this type of analysis generate a large amount of data. Consequently, for the framework to be useful in industry there is a need to make the analysis efficient.

References

1. Murgau, A., Johansson, B., Pejryd, L.: A Study on the Interaction between Physical and Information Flows in Manufacturing Systems. In: CIRP 38th ISMS 2005, Florianopolis Brazil (2005)
2. von Schéele, F., Haftor, D.: Consequences of Cognitive Time Distortion on the Performance of Economic Organisations. Journal of Systems Research and Systems Science (accepted)
3. Standish Group, Chaos report 2012 (2012)
4. Kruchten, P.: The rational unified process: An introduction. Addison-Wesley object technology series, p. 255. Addison-Wesley. xiv, Reading (2005)
5. Beck, K.: Extreme Programming explained: Embrace change, p. 190. Addison-Wesley, Reading (2000)
6. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall, Upper Saddle River (2001)
7. Karlsson, F., Ågerfalk, P.J.: Towards structured flexibility in information systems development: Devising a method for method configuration. Journal of Database Management 20(3), 51–75 (2009)
8. Ralyté, J., Rolland, C.: An Assembly Process Model for Method Engineering. In: Dittrich, K.R., Geppert, A., Norrie, M. C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 267–283. Springer, Heidelberg (2001)
9. Harmsen, F.: Situational Method Engineering. University of Twente, Utrecht (1997)
10. Kelly, S., Lyytinen, K., Rossi, M.: MetaEDIT+ A Fully Configurable Multi-User And Multi-Tool CASE and CAME Environment. In: Constantopoulos, P., Vassiliou, Y., Mylopoulos, J. (eds.) CAiSE 1996. LNCS, vol. 1080, pp. 1–21. Springer, Heidelberg (1996)
11. Karlsson, F., Ågerfalk, P.: MC Sandbox: Devising a tool for method-user-centered method configuration. Information and Software Technology 54(5), 501–516 (2012)
12. Brooks, F.P.: The Mythical Man-Month. Addison-Wesley, Reading (1995)
13. Kumar, K., Welke, R.: Methodology Engineering: A proposal for situation specific methodology construction. In: Cotterman, W.W., Seen, J.A. (eds.) Challenges and Strategies for Research in Systems Development, pp. 257–269. John Wiley & Sons, Washington (1992)
14. Brinkkemper, S.: Method engineering: Engineering of information systems development methods and tools. Information and Software Technology 38, 275–280 (1996)

15. Henderson-Sellers, B., Ralyté, J.: Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science 16(3), 424–478 (2010)
16. Henderson-Sellers, B., Serour, M.K.: Creating a Dual-Agility Method: The Value of Method Engineering. Journal of Database Management 16(4), 1–23 (2005)
17. Rolland, C., Plihon, V., Ralyté, J.: Specifying the Reuse Context of Scenario Method Chunks. In: Pernici, B., Thanos, C. (eds.) CAiSE 1998. LNCS, vol. 1413, pp. 191–218. Springer, Heidelberg (1998)
18. Karlsson, F., Wistrand, K.: Combining method engineering with activity theory: Theoretical grounding of the method component concept. European Journal of Information Systems 15(1), 82–90 (2006)
19. Ågerfalk, P.J., et al.: Modularization Constructs in Method Engineering: Towards Common Ground? In: Situational Method Engineering: Fundamentals and Experiences, Geneva, Switzerland. Springer (2007)
20. Avison, D.E., Fitzgerald, G.: Where now for development methodologies? Association for Computing Machinery. Communications of the ACM 46(1), 78 (2003)
21. Introna, L.D., Whitley, E.A.: Against Methodism: Exploring the limits of method. Information Technology & People 10(1), 31–45 (1997)
22. Russo, N.L., Stolterman, E.: Exploring the assumptions underlying information systems methodologies: Their impact on past, present and future ISM research. Information Technology & People 13(4), 313–327 (2000)
23. Herbsleb, J., et al.: Software quality and the Capability Maturity Model. Communications of the ACM 40(6), 30–40 (1997)
24. Software Enginnering Institute, Understanding and Levaraging a Supplier's CMMI Efforts: A Guidebook for Acquirers. Carnegie Mellon University (2007)
25. van Loon, H.: Process Assessment and ISO/Iec 15504: A Reference Book. Springer (2004)
26. Gaikovina Kula, R., Fushida, K., Kawaguchi, S., Iida, H.: Analysis of bug fixing processes using program slicing metrics. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 32–46. Springer, Heidelberg (2010)
27. Paulish, D.J., Carleton, A.D.: Case studies of software-process-improvement measurement. Computer 27(9), 50–57 (1994)
28. Siok, M.F., Tian, J.: Benchmarking Embedded Software Development Project Performance. In: 2011 IEEE 13the International Symposium on High-Assurance Systems Engineering, pp. 277–284 (2011)
29. Lind, M., Goldkuhl, G.: How to develop a Multi-Grounded Theory: The Evolution of a Business Process Theory. Australian Journal of Information Systems 13(2), 69–85 (2006)
30. Karlsson, F., Ågerfalk, P.J.: Multi-Grounded Action Research in Method Engineering: The MMC Case. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences. IFIP, vol. 244, pp. 19–32. Springer, Boston (2007)
31. Braa, K., Vidgen, R.: Interpretation, intervention, and reduction in the organizational laboratory: A framework for in-context information system research. Accounting, Management and Information Technologies 9(1), 25–47 (1999)
32. Baskerville, R., Wood-Harper, A.T.: Diversity in information systems action research methods. European Journal of Information Systems 7, 90–107 (1998)
33. Argyris, C., Schön, D.A.: Organizational learning II. Theory, method, and practice. Addison-Wesley Publishing Company, Reading (1996)
34. Kaplan, R.S., Anderson, S.R.: Time-Driven Activity-Based Costing: A Simpler and More Powerful Path to Higher Profits. Harvard Business School Press, Boston (2007)

E3value Network Quality Properties

Hans Weigand

Tilburg University, P.O. Box 90153,
5000 LE Tilburg, The Netherlands
H.Weigand@uvt.nl

Abstract. E3value is a well-known technique for modeling value networks that abstracts from processes and platform specifics. Although there exist some methodological guidelines for value modeling, no formal properties have been defined so far that could distinguish "good" from "bad" value models. This is sometimes felt as a gap, both in practice and in teaching. In this paper, some basic formal properties are introduced, based on the notion of value cycle.

Keywords: value networks, formal properties.

1 Introduction

The e3value modeling approach provides a tool for modeling value analysis, helping to determine the value flows for each of the actors [2]. Gordijn's dissertation [1] contains a complete chapter with methodological guidelines for value modeling. Most of these have to do with the meaning or interpretation of the constructs, e.g. the meaning of value object (way of thinking). Others address the question what to do when (way of working). Less attention is given to the way of modeling. What makes a good value model? This is sometimes felt as a need in the application of e3value in practice as well as in teaching (cf. [3]). One could say that the profitability analysis is the "proof of the pudding", but this analysis requires quite some more work and data, and is of little help during the value model construction. There are two ways in which this need could be remedied.

One approach is to use *value network patterns*, and assess the quality of some value network in the way it instantiates a pattern or a combination of these, similar to the approach of Weill & Vitale [7]. Value patterns have been explored in the work of Zlatev [8] which has not been continued, unfortunately. In the form of control patterns, – so focusing on a particular aspect of e3value models – patterns have also been used in [4] although their definition is broader than a value network configuration.

Another approach aims at distinguishing "good" from "bad" business models by checking some *properties* of the network. This is not a replacement of the profitability analysis, but can help during the construction and discussion of the value network, before assumptions about the market volume and costs are made. In this short paper, we follow the second approach, using the notion of value cycle. In section 2, we give

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 464–471, 2014.

a brief overview of e3value. Section 3 introduces some network properties, using the concept of value cycle, and Section 4 provides two small applications.

2 E3value – A Short Overview

The e3value modeling approach provides a tool for modeling value analysis, helping to determine the value flows for each of the actors [2]. The core elements of e3-value models (Fig. 1) are value exchanges, which show the potential transfers or exchanges between collaborating entities in a network of value objects from one actor to another. A value object is of some (economic) value for at least one of the actors. Typical examples for value objects are products, payments and services. Value objects are often bundled; this bundling is represented by the value interface with a value port for each incoming or outgoing value object. Value interfaces also group the value object and the reciprocal value object that the other actor returns. Value activities represent activities that can be performed by an actor in an economically sustainable manner. As such, they are course-grained activities. They are included in the model to allow discussions on which actor is most suitable to perform this activity.

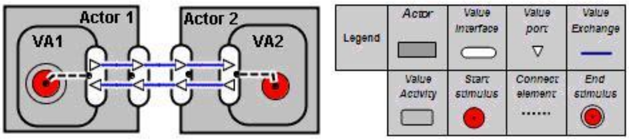


Fig. 1. Basic e3value constructs

3 Value Model Quality

In the original list of guidelines of [1], most principles are about the correct usage of a modeling construct, but two principles can be identified that have a substantial impact on the way of modeling and the quality of the value model: the *reciprocity* principle (guideline 2.7/2.13) and the *causality* principle (guideline 2.8). The latter urges the modeler to find causally related objects. For instance, if a certain good is sold by a trading company, it must have been bought. The two principles can also be paraphrased as follows: “in an exchange, you get nothing for nothing”, and “value objects do not come from nothing”. Formulated as such, they correspond closely to the REA exchange and conversion dualities [5]. Causal relationships are modeled in e3value by means of the scenario paths. However, the use of scenario paths overlaying the value network independently from the value activities has some disadvantages. Apart from that, it is unfortunate that the principles are only formulated as soft guidelines. It may be that formalization in terms of ontological axioms is too rigid, but perhaps there is something in between. In the following, I suggest an alternative approach that is based on the notion of value cycle.

3.1 Basic Definitions

The *value cycle* is a well-known concept in accounting theory, to describe the cycle of money-purchasing-goods-sales-money, or some variant of it (e.g., [6]). To be profitable, this value cycle must contain a value jump, which means that the money flowing into the cash buffer is more than the money outflow. We can map this concept (not in all its details) on the value network in the form of the following requirements:

- A value network is *sound* iff every value exchange between actors (its value object) is part of at least one value cycle and the network is connected.
- A *path* is a chain of value ports $V_1 \rightarrow V_2 .. \rightarrow V_n$ connected by value exchanges (between each two subsequent nodes). A value cycle V is a set of value ports with source $V_1 \in V$ iff each value port in V is on a path from V_1 to V_1
- We assume that value objects can only be transformed in a value activity. That is, if we have a *transformation* subsequence $V_i \rightarrow V_{i+1}$ where the value objects requested and offered, respectively, in V_i and V_{i+1} are different, then the two ports are contained in a value activity, one as in-port and one as out-port (a type III value exchange). In all the other value exchanges in the model, the value objects flowing in and out are the same.
- A value cycle is *simple* when the value exchanges between the subsequent nodes are the only value exchanges existing between any V_i and V_j, for $i< j$. When there are more of these value exchanges (the value cycle includes different paths), the value cycle is called complex.
- A value interface is *balanced* if it contains at least one value port in each direction (in and out). A value network is balanced iff all value interfaces are balanced.
- In a value interface, the incoming and outgoing value objects should be of a different type (cf. Gordijn's guideline 14).

We restrict the notion of value transaction as follows:
- A value transaction includes value exchanges between two actors (not more). There is at least one value exchange in each direction.

The basic e3value model in Fig.1 is balanced (with only one transaction), and it is also sound (one value cycle). The e3value diagram in Fig. 2a is sound but not balanced.

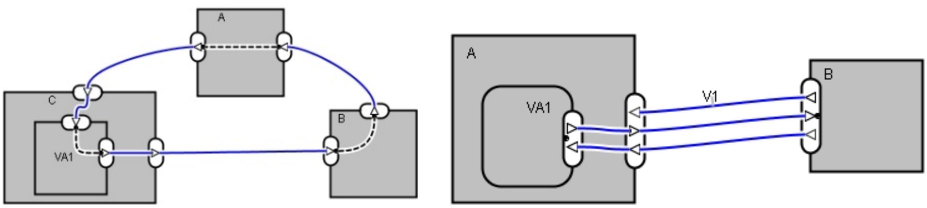


Fig. 2. (a) A non-balanced sound network and (b) a balanced non-sound network

The network in Fig. 2b is balanced but not sound. It is not sound for two reasons: (a) because of a missing value activity in actor B that could close the value cycle, and (b) value object V1 is not part of a value cycle.

3.2 Balancing and Reciprocity

Balancing is related to the reciprocity principle formulated in [1:111] as follows: "A value interface should consist of two reciprocal offerings". This guideline is adhered to quite strictly in Gordijn's dissertation, but has been relaxed sometimes in later work. One reason is that a service bundle may include many offerings. Some of these offerings, such as after-sales support, do not stand in a 1-1 economic exchange duality with a counter offering. Another reason is that the reciprocity makes sense both from a social point of view, as human exchanges strive for symmetry, and as an economic principle, but these two are not the same. We suggest that value transactions are used for those cases where there is a strict economic reciprocity. Consider for example the Google search case, where a web user gets a search service without paying for it. This is not an economic transaction. However, the "attention to advertisements" is one thing that the user returns. If we add this value exchange, the value interfaces (at both sides) are balanced, according to the definition above, but we suggest not modeling a value transaction here.

To turn the value network of Fig. 2a into a balanced network, one of the actors, e.g. C, could be changed into an intermediary. Then instead of B exchanging a value object to A directly, he returns it to C, who forwards it in some way to A.

3.3 Soundness and Causality

To turn the value network of Fig. 7b into a sound value network, the designer has to ask for the added-value of V1. Assume that the transaction is a simple "goods for money" exchange between provider A and customer B, and V1 stands for personal data that B provides as part of the transaction. The added-value of V1 could be that A aggregates the personal data and sells it to a third party. The money received in return is invested into the sales activity (VA1) and so benefits the customer via the goods that he buys. This closes a value cycle, once we have also added a value activity "consume" inside B that produces the value object V1. In Fig.3, we have used two line formats to distinguish the two value cycles in this network. Together, they cover all value exchanges, so the network is sound.

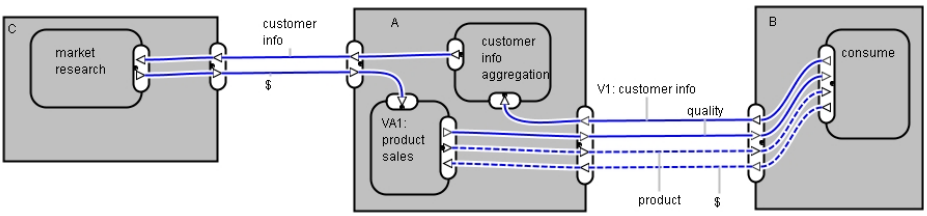


Fig. 3. A sound version of network 2(b), after some extensions

There is a close relationship between value cycle and the concept of scenario path. Both try to capture the causality principle, but in slightly different way. When the value network has only reciprocal transactions, it can be easily verified that each scenario path corresponds to a value cycle (perhaps some value activities must be added). Just follow the path from start stimulus to end stimulus and back. If it is also the case that *each* value exchange between actors is covered by a scenario path – note that this is currently not a requirement in e3value –, the value network is sound. However, the value chain requires transformations of value objects to be explicitly represented as value activities. The difference reflects a subtle difference in perspective on value creation. Value is not only created in economic transactions (value-in-exchange), but also in the combination of different resources to produce something new (co-creation of value). The first kind of value creation is valid for instance for trading companies, but when innovation is becoming a strategic concern, the second kind of value gains in importance.

Scenario path modeling is supported by Use Case Maps that include AND and OR splits. We did not define something equivalent for value cycles (yet). A scenario path with OR splits can be seen as an abstract representation of several value cycles. A scenario path with one or more AND splits corresponds to one (complex) value cycle.

4 Application

To illustrate the methodological use of the balanced and sound properties, consider the "free Internet" example from ([1]. Fig. 4(a) is the initial e3value model. The internet user gets free internet access. This model is not balanced, so we search for a reciprocal value object. In this particular case (the early days of Internet), the Internet provider has a deal with the telecom provider that he gets paid for connections he establishes, a so-called termination fee. So the value object returned by the internet user is the termination, cf. 4(b). To close the value cycle, value activities are inserted, as in 4(c).

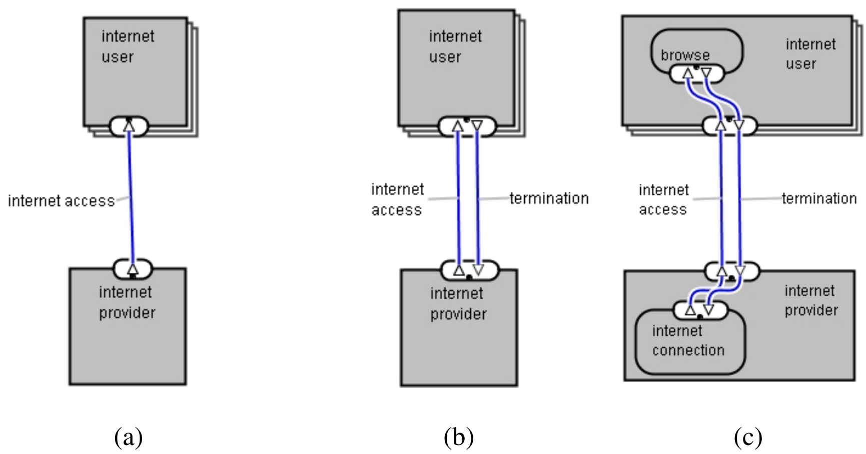


Fig. 4. Free Internet example (a) initial (b) balanced, (c) sound

Although 4(c) is a sound network, it is not clear how the termination is generated by the Internet user, and how it is turned into Internet access at the provider. This leads to an expansion of the model depicted in Fig. 5. Here the termination is traced back to a "call" activity (not free for the browser) and the termination is forwarded to the telecom provider, to boost his connection service (which is also used to serve the "call" from the user). In the resulting model, every value exchange is part of one big value cycle, paraphrased as: (browse) Internet access (internet connection) $ (connect service) termination – connection (call) connection (connect service) $ (call) $ (browse) (value activities are put between brackets and we omit identifications of the value interfaces). It should be noted that this kind of value network is just one possible business model for (early day) Internet providers.

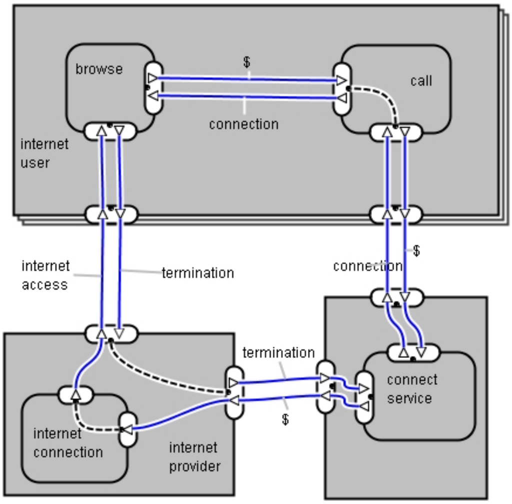


Fig. 5. Free Internet value network complete, equivalent to [1:49]

The two properties "balanced" and "sound" help to make better and more complete value networks. Still, the completeness is relative. It is possible to model a manufacturing company as a value actor with one value activity, "production", with money inflow and a product outflow. In a more complete picture, the manufacturing company uses the money inflow to buy raw material and other resources to support the production. However, every model is bounded, so we do not require the value activity "production" to be decomposed.

A second example is the eye treatment case of Henkel & Perjons [3]. The three actors in this case are the primary health care unit, the patient and the hospital (Fig. 5). First thing to note is that the network is balanced. The value activities are missing, but when they are added, it is not hard to identify two important value cycles: the cycle patient self-care (patient fee+voucher) primary health care (investigation) patient self-care; and, patient self-care (patient fee+voucher) eye surgeon (eye treatment + recipe) patient self-care. Both are complex cycles as they involve some extra exchanges: the patient voucher (that is used by the receiver for further reimbursements from the County Council) and the recipe. What is a bit more difficult is the role of the referral and referral answer. According to the authors, the referral is of value to the hospital as it will increase its income. However, this seems a bit artificial, as in most cases, referrals are necessary because of governmental regulations, and not meant as a marketing channel. It is not clear what the value is of the referral answer, apart from the fact that it creates balancing.

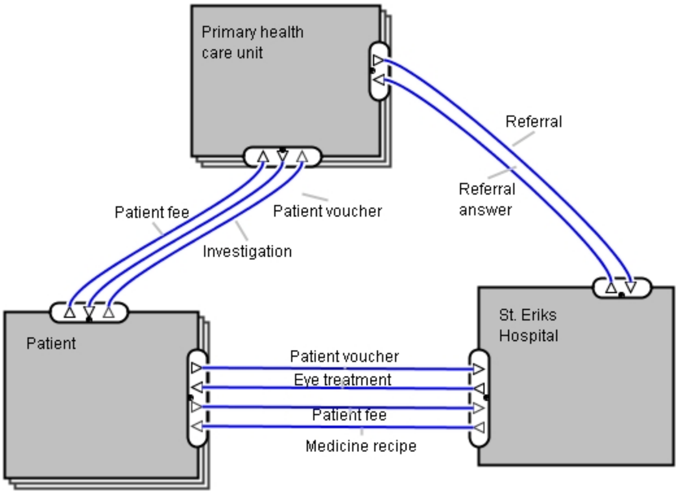


Fig. 6. The Eye Hospital case, taken from Henkel & Perjons [3]

An alternative approach is to view the *right* of referral as something of value to the primary health care units (protection of their market position), and, assumingly, this right is granted by the hospital (based on regulations). On the basis of this right, the health care unit provides a referral to the patient; this is of value as it allows him/her to get into the hospital. The referral is passed on to the hospital. It is not of value to the hospital, we assume, but that does not stop making it a value object, as it is of value to the patient. It can be a required part of the value transaction with the patient (required because of regulation). All together, this means that we can identify a third value cycle: (hospital referral control) right of referral (health unit service) referral (patient self-care) referral (hospital referral control). With this cycle added, the value network is sound. Note that the resulting network is not completely balanced, and that the third value cycle is rather different from the first two ones. This is probably due to the legal as opposed to economic nature of the value exchanges involved. More experience with inter-organizational value cycles is needed in order to get insight in the different patterns that can occur.

5 Conclusion

In this paper, some formal properties of value networks have been defined that can help to distinguish "good" from "bad" value models. The two properties sound and balanced correspond to two basic e3value principles: causality and reciprocity. However, the two properties defined need not be the only ones. This is one topic for future research.

Although the concept of value cycle is not new, its application in an interorganizational setting can be seen as innovative. A traditional value cycle is typically a combination of a physical stream and a money stream. In the market exchange, the two streams get connected and make up a cycle. Services differ from goods, but can be modeled as resources in the stream as well. An interorganizational value cycle contains several parts, of several types. It may be interesting to look at these types in more detail. For instance, physical streams should cycle, sooner or later, for a sustainable network. For that reason, Fig. 3 is incomplete, and to make it complete, as least for the focal actor(s), a “product return” exchange could be added from B to A.

References

1. Gordijn, J.: Value-Based Requirements Engineering: Exploring Innovative E-Commerce Ideas. Dissertation, Vrije Universiteit Amsterdam (2002)
2. Gordijn, J., Akkermans, J.M.: Value based requirements engineering: Exploring innovative e-commerce idea. Requirements Engineering Journal 8(2), 114–134 (2003)
3. Henkel, M., Perjons, E.: Ways to Create Better Value Models. In: Proc. 3rd Workshop on Value Modeling and Business Ontologies (VMBO 2009), Stockholm, Sweden (2009)
4. Kartseva, V., Hulstijn, J., Gordijn, J., Tan, Y.-H.: Control Patterns in a Healthcare Network. European Journal of Information Systems (19), 320–343 (2010)
5. McCarthy, W.E.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review (1982)
6. Weigand, H., Elsas, P.: Model-based auditing using REA. International Journal of Accounting Information Systems 13(3), 287–310 (2012)
7. Weill, P., Vitale, M.R.: Place to space: Migrating to eBusiness Models. Harvard Business School Press, Boston (2001)
8. Zlatev, Z., Eck, P., van, W.R., Gordijn, J.: Goal-Oriented RE for e-services. In: Proc. of the Int. Workshop in Service-Oriented Requirements Engineering (2004)

Applicability of SSM and UML for Designing a Search Application for the British Broadcasting Corporation (BBC)

Ross Fenning[1], Huseyin Dogan[2], and Keith Phalp[2]

[1] BBC, UK
Ross.Fenning@bbc.co.uk
[2] Bournemouth University, UK
{hdogan,kphalp}@bournemouth.ac.uk

Abstract. Whilst there are successful general web search engines such as Google that will find any piece of content, there is a perceived need for a specific search that makes better use of the internal knowledge the broadcasting industry (e.g. BBC) has about its own content. The British Broadcasting Corporation (BBC) is a public service broadcaster funded by the licence fee paid by United Kingdom households. This industry-based case study looks at the applicability of Soft Systems Methodology (SSM) and Unified Modelling Language (UML) to design a hypothetical, high-level view of a search application that receives web content from a variety of BBC content production systems and makes every item then searchable by a BBC website visitor using the search feature. The developers of such search applications can benefit from this specific industry-based case study that contextualised the problem space using SSM and developed UML models to solve the problem.

1 The Problem

1.1 Background

The BBC has been publishing content on the World Wide Web since the mid 1990s and since then the amount and the diversity has increased exponentially. Large websites – or indeed the web as a whole – would not have been usable nor useful without the rise in quality of web search engines.

A large challenge for any web search application is to provide a common interface and set of user interactions that can equally index, search and link to a diverse range of types of information – be it in the form of text, images, video or games. A more recent challenge has been to achieve this in a near-real time way to catch up with the rapid rate at which content is added to the web (particulary from microblogging websites such as Twitter).

Whilst there are successful general web search engines such a Google that will find any piece of content, there is a perceived need for a BBC-specific search that makes better use of the internal knowledge the BBC has about its own content.

I. Bider et al. (Eds.): BPMDS 2014 and EMMSAD 2014, LNBIP 175, pp. 472–486, 2014.

1.2 Design Goals

Whilst the BBC website currently has a functional search feature already, this paper assumes building a new search application from the ground up so as to give full freedom to apply the analysis and design techniques therein. In practice, there are engineering challenges involved in maintaining and building on top of existing systems. Such challenges and details of the existing infrastructure are out of scope for this paper.

The purpose of this paper is to explore how contextualisation through SSM and then using UML to form a high-level design might benefit building a hypothetical, new search application for the BBC. The hope is that insights from fresh analysis and design might form suitable proposals for potential improvements and development within the existing application. These approaches might also lead to further work to evaluate the current application by highlighting where the BBC Search application differs from an "ideal" model (if it can be described as such) derived from such a redesign with a contemporary view of the problem space.

Ultimately, any organisation like the BBC is unlikely to replace a large application atomically, but is likely to migrate over time to its "ideal" form with smaller, iterative improvements. It is proposed that there is some value in designing what that "ideal" form might be so as to provide proposals for those improvements.

1.3 Problem Space Contextualisation through Soft Systems

The target audience for the BBC is effectively the entire population of the UK and amongst those that do make use of BBC services, there is much diversity of needs, preferences and technical ability. It is clear it is no small task to design a search-based discovery mechanism of millions of diverse pieces of content aimed at millions of diverse people.

Using *Soft Systems Methodology* (SSM) [1], we can stand back from an ontological approach of defining what the search system *is* or *comprises* and instead take an *epistemological* view of search as a system. With this view, we could consider a system that holistically transforms members of the public's desires to find online content into the consumption of that content – whether those desires are *precise* (e.g. they want an exact article known by headline they saw earlier or a particular programme they missed on television) or those desires are *fuzzy* (e.g. news about a certain topic, any comedy programme, learning materials about the Industrial Revolution).

Checkland and Scoles[2] decribed a *Rich Picture* approach for representing a problem situation early in SSM approaches. Given the size and complexity of the search system as a whole, a useful initial step is to create such an informal representation of what is known about the problem. Figure 1 shows what the authors know of the audience, search and most BBC online content areas. Note that not all areas are covered and a strong emphasis is placed on TV catch-up (e.g. via the iPlayer product). Radio catch-up is not mentioned as it shares a lot of similarity with television in terms of use and any differences are out of scope for this design.

This rich picture was created using domain knowledge dervied from the existing search system and technical knowledge of the BBC websites. If the initial designs from this paper prove promising in practice, the authors would recommend repeating the exercise with a broader range of stakeholders and domain experts from respective subsystems. A rich picture does not have to serve as authoritative snapshot of the problem space, but can instead been seen as a collaborative exercise between stakeholders. Such contextualisation is likely worth iterating over time as the industry changes, e.g. the recent shift from separate mobile websites to *Responsive Design*[3] might change the understanding around mobile devices.

Dogan and Henshaw[4] showed how a “soft” systems approach called Interactive Management can be adopted to capture the requirements and contextualise the problem space. This involved the process of transitioning from the soft systems results to a formal model (e.g. UML). This transition was enabled by dividing the actors in the rich pictures into meta-level and direct users of the system. Although this division was subjective and depended on the interpretation and analysis of the rich pictures to provide a structure for the use case model, the rich pictures themselves were created through interactions with subject matter experts. The soft systems, and hence the Interactive Management results, provided the baseline information to derive a formal model including UML use case, sequence and domain models. The transitioning from “soft” to “hard” systems can be set within the State of the Art including the requirements analysis and modelling as used in SSM, UML and Business Process Modelling.

The overall design objective of this paper will be to create an initial proposal for a search application to drive the missing components within the holistic system depicted in Fig. 1. Some subsystems already exist, e.g. for journalists to write news articles and publish them on the BBC News website, but for the purposes of this design exercise, we will assume no existing application to drive a search-based discovery of those websites.

The design will look to integrate with existing subsystems where possible rather than attempt to replicate work already done. For example, journalists will prefer that a search application can integrate with the system into which they are publishing their articles instead of being required to publish their articles into two systems.

2 Design

2.1 Use Cases

From the rich picture in Fig. 1, we extracted the activities that are clearly within the remit of a search application. For example, the ability for editorial staff to manage the content of the search indexes sounds like a feature the search application would provide. Conversely, television actors and other contributors to a programme are likely to interact only within a television production subsystem

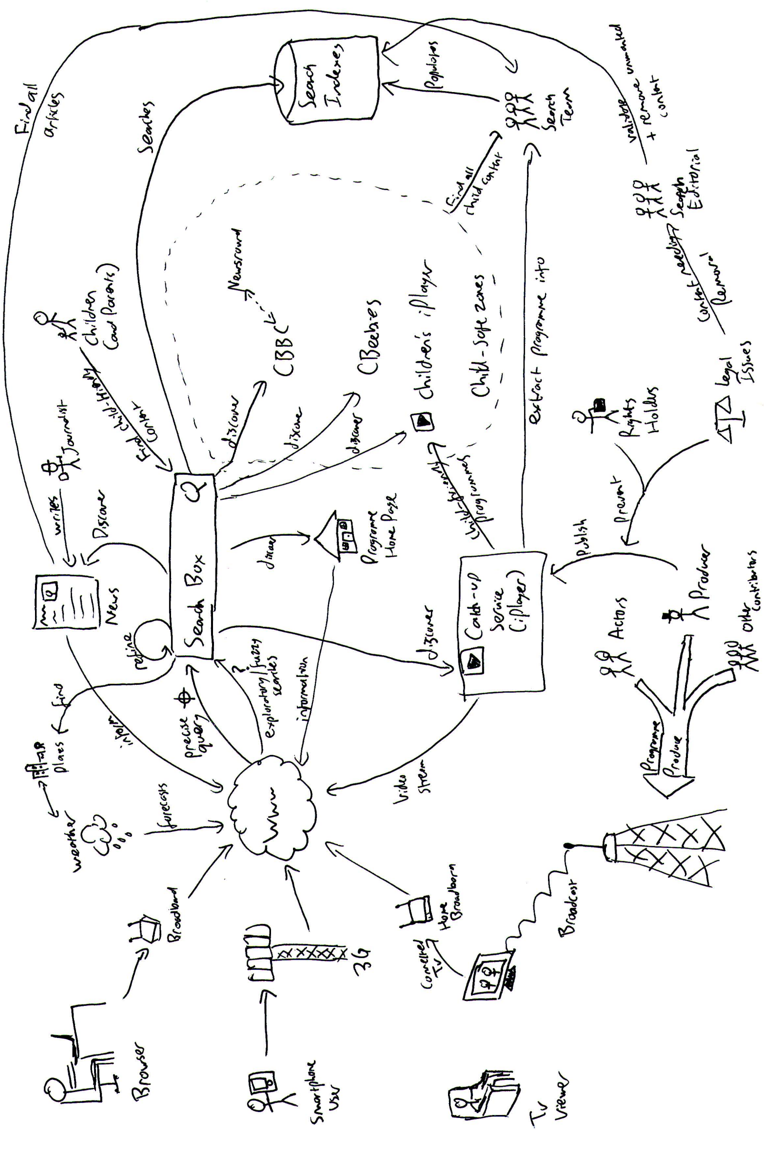


Fig. 1. Rich Picture of the BBC search system

with a producer or content editor being responsible for publishing information about the final production's broadcast and availability for streaming online.

In other words, we can say that editorial staff and content editors might be seen as direct users of the system, with actors, presenters, etc. seen as meta-level users. This is justified by noting that journalists and editorial staff will have their content exactly as typed appear in search results, but production staff behind television and radio programming contribute only indirectly to the search system. They produce the content that users will later wish to see, but they have little contribution to the discovery of that content later on.

This follows from the Interactive Management approach from Dogan and Henshaw[4] and is still largely subjective. For example, if a journalist publishes to a content management system, with which a search system then integrates without their knowledge, can they still be considered to be a direct user of the search system? In this design we argue that such syndication into the search system means their actions (e.g. to publish, remove, update articles) will have direct effect on the search and discovery of those articles (e.g. they may become searchable, cease being searchable or start being searchable under new criteria) and thus they are direct users of the search system. It is recognised however, that a more collaborative rich picture drawn up with a wider set of stakeholders might lead to a different opinion.

This is not to say that we can simply cross off certain elements from our depiction of the problem because they do not directly interact with the subsystem being designed. The *systems thinking* approach advocated by Checkland[5] encourages us to consider the irreducible properties of each system at each level of abstraction. Thus we need to consider not only a search application subsystem that solves specific problems for its immediate users, but also an application that contributes to the desirable, emergent properties of the BBC service as a whole.

In the specific example of television programming, we need to maintain systems thinking throughout the design process to ensure that we create a search application that both meets the needs of the public using the application to search for programmes and forms part of a television production and delivery system that itself meets the needs of the television-watching public.

Thus a suitable design strategy is to apply systems design to the search application in isolation – as a *hard problem* – but then to use the wider system to inform, shape and evaluate that design.

Having extraced the direct users only of the search system, further requirements-gathering and business analysis can define their respective use cases of the system. An analysis based on domain knowledge from the existing search system leads to the use case UML diagram shown in Fig. 2.

This illustrates only a subset of the expected behaviours for a full BBC search application, but touches on some of the diversity of the potential uses. For the purposes of our initial design, we can next look into defining the system behaviour for some of these use cases. It should be noted that these are illustrative of the breadth of use cases, but deeper business analysis and creation of corresponding use cases (perhaps following the style of Cockburn[6]) should be performed as part of a more in-depth study before the high-level design can be truly considered complete and valid.

2.2 System Behaviour

The breadth of content the search system will need to index and retrieve sits across multiple systems within the BBC (e.g. programme metadata lives in a distinct location to news articles). The diversity of systems, formats and metadata involved suggested approaching the more functional design as a classic *Enterprise Integration* problem[7].

Figure 3 shows how a TV producer indirectly interacts with search by providing information that ultimately ends up in the search indexes and Fig. 4 shows a sequence diagram defining a user interacting with the search system.

The key design decision in Fig. 3 is the use of asynchronous messages only. A non-blocking set of interactions such as publish-subscribe[8] is a good way to decouple systems that produce and store progamme information from the search application systems. If the systems surrounding the programmes database can be built a *channel adapter*[8] to integrate it to a messaging system, then the search indexes can receive changes to information without TV producers, journalists, content editors, etc. even being aware this is happening.

Note in Fig. 4 that while the search indexes will contain representations of content from several source systems, the intention shown is that richer information about the domain model will not be held in the search indexes. This goes along with the principle of separation of concerns[9] in that the search index component can focus on optimising its data structures around retrieval. This kind of modularity also allows source systems maintained by other teams to take the responsibility of the accuracy of the information, which fits in with the wider holistic view of the system: the volume of information involved requires that separate teams are responsible for the accuracy of their data.

This can be seen as similar to the *Lazy Load* pattern [10] in that the search indexes will only return stub objects that are capable of retrieving the fuller information to need. This, however, could lead to a lot of calls to different service applications per page of results. This can be done in parallel, but the fact still remains that the user has to wait for all this to assemble before seeing even one result.

One solution to this in modern web application design is to push the lazy loading into the web browser using AJAX[11]. Such a solution is depicted in Fig. 5. Whilst this still requires just as many calls to backend services – perhaps even more complexity as the calls are going through more layers – it can give a user experience that appears more responsive.

Preparing a search results page with minimal information that is then augmented asynchronously is a user experience technique that gives the illusion of lower latency; the additional information can update the page during the user's reaction time in the best case.

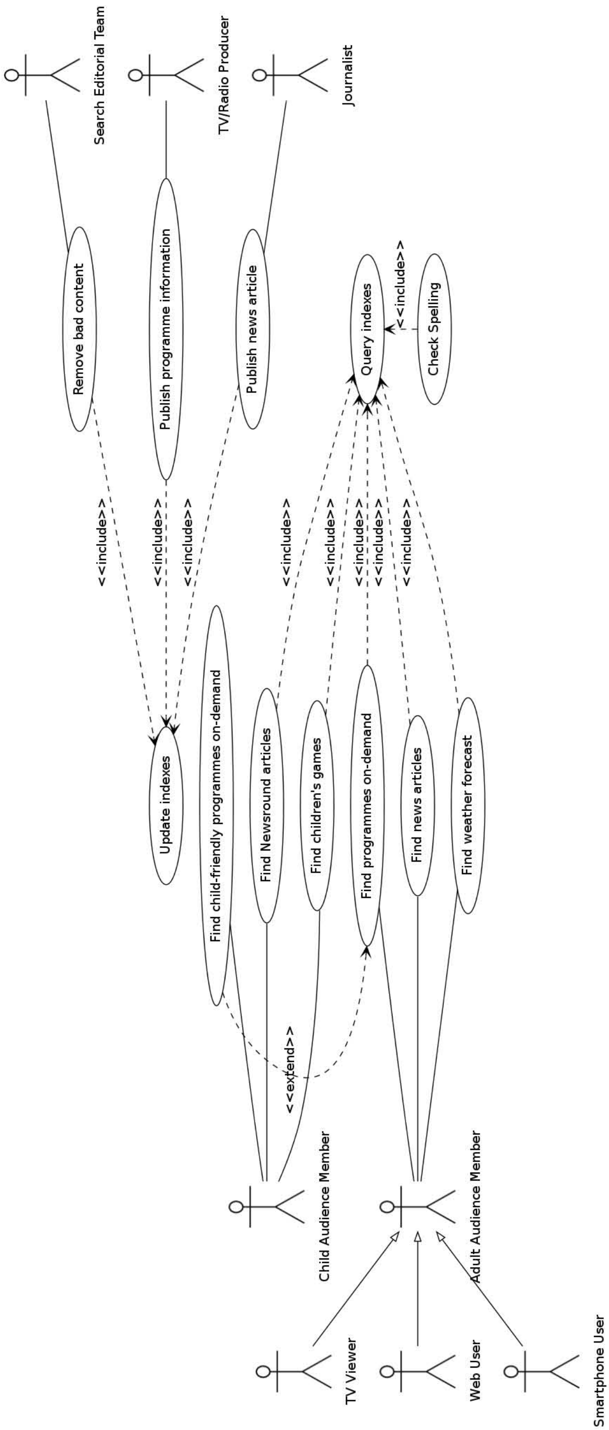


Fig. 2. Use case diagram for BBC Search application

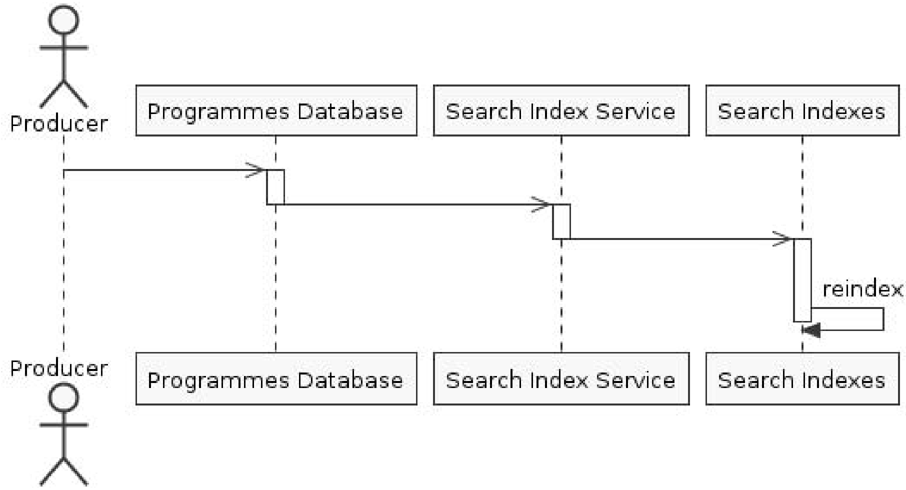


Fig. 3. Sequence diagram showing publication of programme information to the search indexes

2.3 Domain Model

It has been expressed already that a BBC-wide search application would need to index, retrieve and display a diverse set of information. Parts of the application might well want to incorporate a domain model[10] so as to understand how to perform each of these actions against each possible item the search application could return as a search result.

Domains are the distinct subject matters present in any system representing large, reusable components and are depicted using a domain model which shows an organisation of UML packages and their dependencies[12]. The use case diagram and domain model developed provides a baseline model for a future search application system.

A maximal domain model is shown in Fig. 6 that attempts to capture a good proportion of the content and concepts the BBC has been making efforts to model over several years. This model is an aggregate of individual ontologies developed for specific purposes, but given the search application has to provide the discovery for the full set of this information, it is not unreasonable that a search application would have a domain model that covers the totality.

Note that complexity of information for programmes alone[13]. A programme to a member of the public could actually refer to an exact episode or indeed the *brand*, i.e. the title of the programme in general. An example of a brand would be *Doctor Who*, [1] which itself contains multiple series, which in turn contain

[1] *Doctor Who* is a popular, long-running science-fiction television programme produced by the BBC and is frequently sought by users of the BBC iPlayer television catch-up service after an episode has aired.

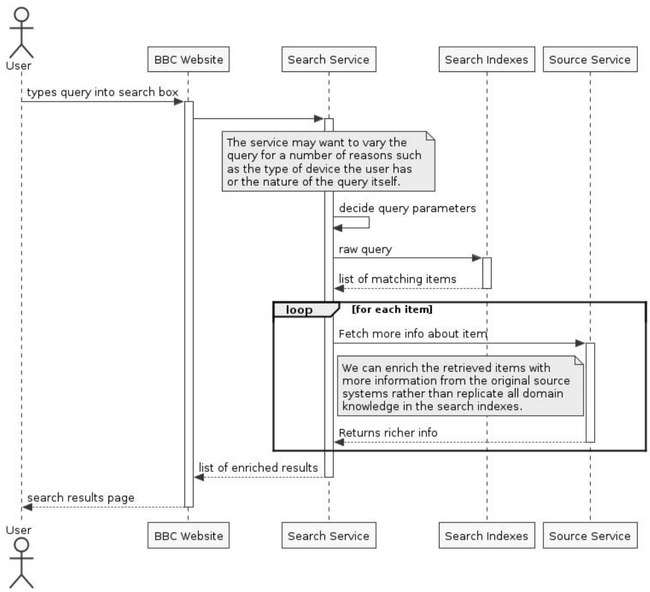


Fig. 4. Sequence diagram showing a user interacting with search

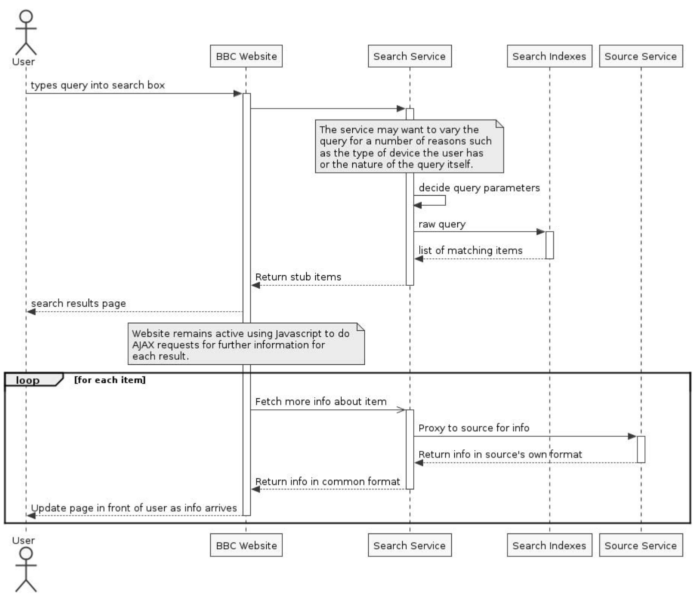


Fig. 5. Sequence diagram showing rich information lazy-loaded via AJAX

a collection of episodes. Note that the brand itself has episodes as immediate children that do not live under a series, e.g. Christmas specials. It is also the case that certain things do not have brands or series, e.g. a film is modelled as a one-off episode.

This leads to some difficult questions for a search application. If a user searches for the text "doctor who", are they expecting a link to the latest episode to watch on iPlayer, information about the next episode – such as when it is due to be broadcast – or a link to the overall home page for the entire Doctor Who brand?

The model also skims the surface of the sport ontology[14] created before the 2012 Olympic games, which aims to model the whole domain of sporting personalities, events and competitions (and more). This might be too fine-grained for the domain model used within the search application, but it is likely that people will want to search for competitions like "World Cup" or sporting disciplines such as "football". A search application that understands these concepts as entities in their own right may well be able to direct users at a curated, dedicated "home page" thereof alongside simply matching articles and other works that contain those terms.

2.4 Discussion and Analysis

The problem of a BBC-wide search application was predicted – and has certainly shown itself – to be a very large-scale problem, the full extent of which cannot be covered in this paper. Thus it is reasonable to conclude that the problem is not yet solved at every level, but the high-level solutions are certainly promising.

The use case diagram is likely not complete and would need a significant amount of user research and requirements-gathering to collect all the possible use cases of the search system. It is likely that the variations of the use cases are so numerous that different diagrams exploring different combinations of use cases should be made to replace the single one given. For instance, very little has been touched on around users seeking educational and informative material such as *Bitesize* [2] or any other learning resources.

The sequence diagram for the producer (or any other content creator) indirectly getting their content into the search system demonstrates the need for asynchronous messaging (and publish-subscribe), but the nuances of such a process are not comprehensively shown in this format. A better illustration for such a message-based integration system might be derived from the illustrated patterns created by Hohpe and Woolf[8].

The behaviour of the query web application in Fig. 4 is defined with a hard line taken on keeping the information stored in the indexes to a minimum. At a basic level, this is not unlike a *content enricher* pattern [8] whereby the search application business layer enriches the stubs in the indexes with information the source system (the index) simply does not have. At a more purist extreme, this

[2] *Bitesize* is the name given to the BBC's free web-based study materials for school children aged between 5 and 16, covering varying curricula for England, Wales and Scotland.

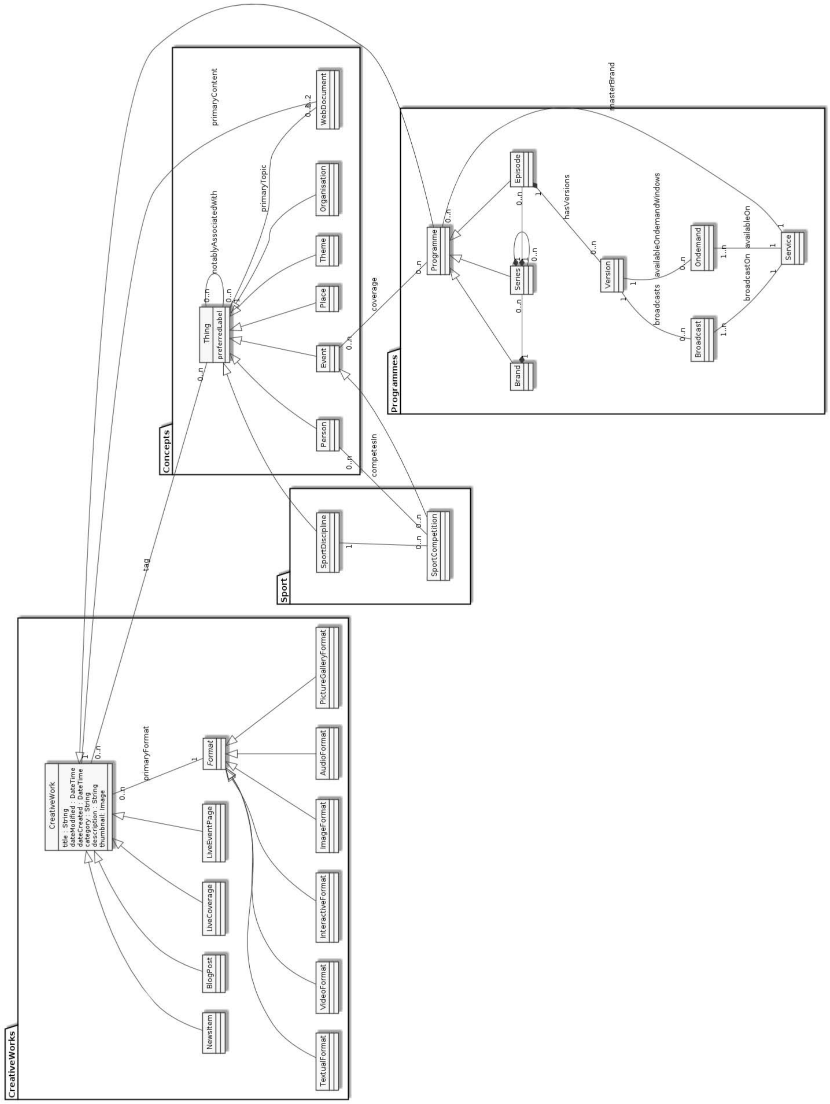


Fig. 6. Domain model for content items and other things pertinent to BBC content

could follow a *claim check* pattern where the index returns only globally-unique identifiers for the matching items and makes no attempt to present any other knowledge about them (and thus avoids any synchronisation issues if an item is updated at source).

The latter pattern fully decouples the search indexes from any deeper information – leaving that responsibility to appropriate source system – but means that the raw, stub results are near-useless to an end user. This makes the AJAX-based alternative approach in Fig. 5 less of a desirable option (would a user really be presented a list of URIs while some Javascript code replaces them one by one with actual information?). Given the AJAX-driven behaviour still appears desirable in terms of responsiveness, it would seem that some trade-off would need to happen in terms of what the indexes store as additional information.

These decisions relate to how the system would implement the data model in Fig. 6. The ontology described is an aggregation of several efforts by Raimond[13], Rayfield [14] and others – along with some additional work to join them together – to represent the wealth of information with which the BBC deals. It could be argued both that a search application that *understands* this diversity of content must reflect it in its domain model but that a search application that is *coupled* to the individual ontologies in this way is brittle with regard to changes therein. Duplication of business models across different applications would only harm maintainability.

Thus the domain model given in Fig. 6 should only serve as a communication artefact that leads to further development of a domain model more suitable to the needs of the search application without the burden of maintaining more than is necessary. The use of *subtype polymorphism*[15] is likely to be key to ensure the search domain model contains only the APIs it needs. For example, does the search application have a need to differentiate between a *NewsItem* and a *BlogPost* for the purposes of displaying the search result's title? If the CreativeWork top-level class has a title property, then the domain model so far to enable that one behaviour needs only one class!

The application could go further and interact via a single SearchResult *facade*[10] whose instances provide appropriate responses to canonical hooks such as *getDisplayTitle()* and *getDestinationUrl()* via polymorphic *composition* with different target classes from the fuller domain model. Given that the search indexes are likely to take in content from any number of future systems, there is an appeal to taking a schemaless approach[16] and allow the search application's view layer to display different kinds of results differently via *duck typing* [17].

A dynamic, schema-free domain model used within the search application – noting that, of course, schemaless truly means there is an implicit schema in the logic that creates these dynamic objects from source systems[16] – could prove to aid the search system's need to model different kinds of data in the same index. For example, our designers might express a desire to put small summaries of weather forecasts within search results that contain places. In the static sense, we could say that anything of *Place* type is displayed with such a feature. With a dynamic, duck-typed model, we could ask "does this item have weather?"

in place of “is this a place?”, thus allowing for the future design that extends forecasts to football matches (since football matches are *Event*s that occur in a *Place*, it is reasonable for a weather property to be set thereon).

Overall, we have some promising, high-level models from which to start making such more fine-grained decisions about the search application. The use cases are likely sufficient for early iterations or a *Minimal Viable Product*[18] and the behaviours capture the overall needs of the system. There is still much scope for returning to our systems thinking and Soft Systems Methodology to monitor the general model for Checkland’s “3 Es” (efficacy, efficiency and effectiveness)[2], which is only briefly touched on in the authors’ attempts to relate the data model to the user’s interaction with the system.

3 Evaluation

The high-level nature of the modelling achieved has made it difficult to meet any objectives over performance, latency or robustness of software components involved. A BBC search application needs to index news as it is published, serve millions of unique visits every week and minimise potential for losing information. These non-functional requirements are a challenge in their own right and have not been met with the UML modelling presented.

This is not to say that UML is not capable as a tool for presenting architecture around performance and resilience. However, it is perhaps more appropriate to model and present some of these aspects through component and deployment diagrams. Behaviour diagrams such as sequence and activity do not suit well to showing timing or performance, although a series of sequence diagrams could show how behaviour changes in parts of the system to tolerate failure of other components (e.g. one component could be modelled to return from an internal cache if a collaborator is returning temporary error status codes).

Even the models that are presented in this paper do not paint the full picture, but it could be argued that it is not their purpose to do so. As stated at the end of Sect. 2.4, the use case diagram provides a starting point that might be sufficient for an early iteration of a project in an *Agile* methodology[19].

An Agile approach to developing the search application could distill the use cases even further and shape the requirements to the rest thereof later in the development process based on feedback and reacting to changes. A similar approach would allow us to start with the smaller domain model also discussed in Sect. 2.4 and allow it to grow to the necessary size to need – i.e. we can defer the decision of “how big should the domain model be?” until we are at a point where we have more information to answer such a question.

Thus even if the use case and domain models are not as comprehensive or as honed as they need to be to build an entire application, they serve their purpose adequately to communicate the first iterations of development or – especially in the case of the domain model – information pertaining to the whole organisation, even if only small parts of it are they modelled directly in the application being built.

The sequence diagrams seem to provoke more debate of the merits of returning rich information in a single response versus an AJAX-driven approach (or some trade-off in between). Again, the authors would emphasise the communication aspect of UML modelling and argue that encouraging such debate is a successful application of UML, not a failure because a decision has not been made between two designs at this early stage.

Some of this hints at a drawback of UML modelling being that it encourages a lot of design decision at an early stage of a project – a stage at which we arguably have the least information[20]. However, there are plenty of efforts in spite of this that promote modelling and UML being used compatibly within an Agile process. The use of other modelling techniques in an Agile setting such as Rational Unified Process (RUP) and Agile Unified Process (AUP) have been suggested as suitable in an Agile project[21]. It may have been more suitable to incorporate a more diverse range of modelling techniques in the design and analysis presented so as to find the true strengths of each respective approach.

In conclusion, the UML modelling presented communicates some promising approaches to the BBC search problem, but it is far from sufficient in its own right. Designing to the level of detail required would result in a rigid development plan that made unverified assumptions, but an iterative approach to modelling that starts at a high-level and updates as development progresses could be used successfully in such a project.

A post-analysis study is required to evaluate the SSM and UML models. The models need to be applied to further projects for validation and verification purposes. In addition, the maturity and evolution of the artefacts need to be considered e.g. adding, deleting, or modifying, if the boundary and context changes.

References

1. Checkland, P., Poulter, J.: Learning for action: a short definitive account of soft systems methodology and its use for practitioner, teachers, and students, vol. 26. Wiley Chichester (2006)
2. Checkland, P., Scholes, J.: Soft systems methodology in action, vol. 7. Wiley, Chichester (1990)
3. Marcotte, E.: Responsive web design. A list apart 306 (2010)
4. Dogan, H., Henshaw, M.: Transition from soft systems to an enterprise knowledge management architecture. In: International Conference on Contemporary Ergonomics and Human Factors, April 13-15. Keele University, UK (2010)
5. Checkland, P.: Systems thinking, systems practice: Includes a 30-year retrospective. John Wiley & Sons (1999)
6. Cockburn, A.: Writing effective use cases. Pearson Education (2001)
7. Brosey, W.D., Neal, R.E., Marks, D.F.: Grand challenges of enterprise integration. In: Proceedings of the 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation, vol. 2, pp. 221–227. IEEE (2001)
8. Hohpe, G., Woolf, B.: Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Professional (2004)

9. Dijkstra, E.W.: On the role of scientific thought. In: Selected Writings on Computing: A Personal Perspective, pp. 60–66. Springer (1982)
10. Fowler, M.: Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co., Inc. (2002)
11. Garrett, J.J.: Ajax: A new approach to web applications (2005), http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications
12. Dickerson, C.E., Mavris, D.N.: Architecture and principles of systems engineering. Complex and Enterprise Systems Engineering Series. Auerbach Publications (2009)
13. Raimond, Y., Sinclair, P., Humfrey, N., Smethurst, M.: Bbc programmes ontology (2009), http://purl.org/ontology/po/2009-09-07.shtml
14. Rayfield, J., Wilton, P., Oliver, S.: Bbc sport ontology (2011), http://www.bbc.co.uk/ontologies/sport/2011-02-17.shtml
15. Booch, G., Maksimchuk, R.A., Engle, M.W., Conallen, J., Houston, K.A., Young, B.J.: Object-Oriented Analysis and Design with Applications. Addison-Wesley (2007)
16. Sadalage, P.J., Fowler, M.: NoSQL distilled: A brief guide to the emerging world of polyglot persistence. Addison-Wesley (2012)
17. Python Software Foundation: Duck-typing: Glossary – python v2.7.5 documentation (2013), http://docs.python.org/2/glossary.html#term-duck-typing
18. Junk, W.S.: The dynamic balance between cost, schedule, features, and quality in software development projects (2000)
19. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al.: The agile manifesto. The agile alliance 200(1) (2001)
20. Kelly, A.: Conway's law v. software architecture (2013), http://allankelly.blogspot.co.uk/2013/03/conway-law-v-software-architecture.html
21. Ambler, S.: Agile modeling: Effective practices for extreme programming and the unified process. Wiley.com (2002)

Author Index