Alexandros Nanopoulos
Werner Schmidt (Eds.)

LNBIP 170

S-BPM ONE - Scientific Research

6th International Conference, S-BPM ONE 2014
Eichstätt, Germany, April 22–23, 2014
Proceedings

Springer

Lecture Notes in Business Information Processing 170

Alexandros Nanopoulos
Werner Schmidt (Eds.)

S-BPM ONE - Scientific Research

6th International Conference, S-BPM ONE 2014
Eichstätt, Germany, April 22-23, 2014
Proceedings

Springer

Volume Editors

Alexandros Nanopoulos
Katholische Universität Eichstätt-Ingolstadt
Ingolstadt, Germany
E-mail: alexandros.nanopoulos@ku.de

Werner Schmidt
Technische Hochschule Ingolstadt
Ingolstadt, Germany
E-mail: werner.schmidt@thi.de

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-319-06064-4 e-ISBN 978-3-319-06065-1
DOI 10.1007/978-3-319-06065-1
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014934836

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

S-BPM ONE 2014 was the sixth conference in the S-BPM ONE series. Under the theme "Value.Net.Works" it brought together people from research, education, and business practice. Becoming aware of the various contextual factors of value creation in and through process networks, the aim was to explore the multiple facets of network-driven BPM. Consequently, authors were to address various topics such as:

- Governing Process Networks
- Embedding Network and Actor Theories in S-BPM
- Expressing Network Process Values
- Enriching BPMN Toward a Network Process Management Notation
- Dynamic Stakeholder Management Based on S-BPM
- Enabling Organizational Resilience Based on (S-)BPM
- Open Process Access in BPM Networks
- Advancing Social BPM Toward Subject-Oriented BPM
- Repositioning Agent-Based Systems for Value-Based BPM
- Process Values "in the Cloud"
- Value Emergence in Process Networks
- Enterprise Architecting for Value-Based Process Networks
- Enriching Cross-Organizational Business Processes for Value Creation
- Repositioning BPM in Organizational Learning and Knowledge Management Networks

In all, 43 contributions underwent strict peer reviews by an international Program Committee, 13 of which were selected as research papers for this volume. The authors of accepted contributions could share and discuss results of research, application, and experience studies and work in progress or ideas for future work with the audience. Although the submissions push forward the discussion along various dimensions of S-BPM in particular, and BPM in general, they stress the following research issues:

1. Correctness, interchange and transformation of process models S-BPM draws the attention to the crucial role of models for value-creating organizational development, not only within organizational borders but also in a setting of internetworking business partners. This is reflected by approaches to establish ontologies, formal checking, and cross-organizational execution environments for S-BPM models, as well as by a framework for embedding processes into their organizational context.

2. Handling dynamic changes through agile process management Contributions referring to this aspect endorse the growing need for concepts and solutions

to timely support organizational development. The properties of the comprehensive subject-oriented approach, having developed around its easy-to-use communication-oriented modeling language, qualify S-BPM as a promising candidate to master this challenge successfully.

Stimulating keynote talks opened each of the two conference days.

Dr. Michael Rosemann, Professor and Head of the School of Information Systems, Queensland University of Technology, Brisbane/ Australia, expressed his thoughts on the "The Future of Business Process Management."

"The Pretense of Knowing the Customer as the Vehicle to Improve Customer Experience: Repositioning Process in the Front Office of Organizations" was the title of the speech given by Dr. Jorge Sanz. He is IEEE Fellow, Director of the Business Analytics Center at National University of Singapore, Chief Innovation Officer Retail Banking, IBM Corporation, and Visiting Professor at the School of Computing at National University of Singapore.

The above-mentioned areas, together with the impulses of the keynotes, open the field for future research and collection of practical experience in applying subject-oriented concepts and other novel ideas in BPM.

Organizing a successful conference needs many people valuably contributing in various ways:

- The keynote speakers, giving inspiring lectures
- The authors of the contributions presenting their work
- The members of the international Program Committee carefully reviewing papers and giving constructive comments
- The session chairs moderating the presentations and interactions between participants

We also thank the Collegium Willibaldinum for hosting S-BPM ONE 2014 in their impressing baroque building, and the numerous helping hands guiding us through the whole program of the event. Special thanks goes to the Institute of Innovative Process Management (I2PM, www.i2pm.net), the umbrella institution of the overall S-BPM ONE conference series and related projects like Open S-BPM.

Last but not least we thank Ralf Gerstner, Viktoria Meyer, and Christine Reiß from Springer for their assistance and support in publishing these proceedings in the LNBIP series.

April 2014

Alexandros Nanopoulo
Werner Schmidt

Organization

Steering Committee

Albert Fleischmann	Metasonic AG, Germany
Werner Schmidt	Technische Hochschule Ingolstadt, Germany
Christian Stary	Universität Linz, Austria

Executive Committee

Conference Chairs

Alexandros Nanopoulos	Katholische Universität Eichstätt-Ingolstadt, Germany
Werner Schmidt	Technische Hochschule Ingolstadt, Germany

Organizing Chairs

Armin Felbermayr	Katholische Universität Eichstätt-Ingolstadt, Germany
Florian Wallner	Technische Hochschule Ingolstadt, Germany

Program Committee

Alexandros Nanopoulos (Chair)	Katholische Universität Eichstätt-Ingolstadt, Germany
Werner Schmidt (Chair)	Technische Hochschule Ingolstadt, Germany
Reza Barkhi	University of Grenoble-LIG, France
Jörg Becker	Westfälische Wilhelms-Universität Münster, Germany
Noureddine Belkhatir	University of Grenoble-LIG, France
Onur Demirors	Middle East Technical University Ankara, Turkey
Herbert Fischer	University of Applied Sciences Deggendorf, Germany
Peter Forbrig	University of Rostock, Germany
Andreas Gadatsch	Hochschule Bonn-Rhein-Sieg, Germany
Alexander Gromoff	Moscow National Research University, Higher School of Economics (HSE), Russia
Christian Huemer	Technical University of Vienna, Austria
Ebba Thora Hvannberg	University of Iceland, Iceland
Mikhail Komarov	National Research University of Higher School of Economics, Russia
John Krogstie	Norwegian University of Science and Technology, Norway

Matthias Kurz	Datev eG, Germany
Juhnyoung Lee	IBM T.J. Watson Research Center, USA
Franz Lehner	University of Passau, Germany
Jan Mendling	Wirtschaftsuniversität Wien, Austria
Tansel Özyer	TOBB Economics and Technology University Ankara, Turkey
Adrian Paschke	Freie Universität Berlin, Germany
Manfred Reichert	Ulm University, Germany
Wolfgang Renninger	University of Applied Sciences Amberg-Weiden, Germany
Susanne Robra-Bissantz	Technical University of Braunschweig, Germany
Gustavo Rossi	LIFIA Universidad Nacional de La Plata, Argentina
Detlef Seese	Karlsruhe Institute of Technology (KIT), Germany
Robert Singer	FH Joanneum University of Applied Sciences, Austria
Christian Stary	Johannes Kepler University Linz, Austria
Armin Stein	Westfälische Wilhelms-Universität Münster, Germany
Victor Taratoukhine	National Research University Higher School of Education, Russia
Alexandra Totter	ByElement GmbH, Switzerland
Eric Tsui	The Hong Kong Polytechnic University, China
Oktay Türetken	Eindhoven University of Technology, The Netherlands
Jan Verelst	Universiteit Antwerpen, Belgium
Jan vom Brocke	University of Liechtenstein, Liechtenstein
Guido Wirtz	University of Bamberg, Germany
Cornelia Zehbold	Technische Hochschule Ingolstadt, Germany

Sponsoring Institutions

ConsSys IT AG, Cham, Switzerland
IANES (Interactive Acquisition, Negotiation and Enactment of Subject-Oriented Business Process Knowledge) – EU FP 7 Marie Curie IAPP
IFG Ingolstadt AöR, Ingolstadt, Germany
INFOMEDIAservices GmbH, Vienna, Austria
IngolStadtLandPlus - Initiative Regionalmanagement Region Ingolstadt e.V.
Katholische Universität Eichstätt-Ingolstadt
Maihiro GmbH, Ismaning, Germany
METASONIC AG, Pfaffenhofen-Hettenshausen, Germany
Technische Hochschule Ingolstadt Business School, Germany
VALIAL Solution GmbH, Ilmmünster, Germany

METASONIC
business. in tune.

ConsSys IT AG
CONSULTING MIT SYSTEM

maihiro

smart4sense2act.com

VALIAL
Solution

ianes

IFG Ingolstadt AöR
Ein Unternehmen der Stadt Ingolstadt

INGOL
STADT LAND
PLUS
Region für Innovation.

Technische Hochschule Ingolstadt
Business School

KATHOLISCHE UNIVERSITÄT
EICHSTÄTT INGOLSTADT

Keynote (Abstracts)

The Future of Business Process Management

Michael Rosemann

Queensland University of Technology
Science and Engineering Faculty, Information Systems School
2 George Street, Brisbane Qld 4000, Australia
m.rosemann@qut.edu.au

Abstract. Business Process Management has substantially matured over the last two decades. The tools and methods available to design, analyze, implement, execute, monitor and mine a process have been scientifically researched and can be deployed in practice. In fact, many of these BPM capabilities are nowadays a commodity. However, in order to ensure an ongoing impact of BPM in an increasingly networked and mobile society, new challenges and opportunities need to be explored. In this context, this keynote presents four future pathways for BPM academics and professionals. Value-driven BPM (x-aware BPM) is introduced as a BPM approach driven by the desired outcomes instead of predefined methodologies. Ambidextrous BPM demands a stronger focus on explorative BPM, i.e. how to actually design future processes instead of simply exploiting existing problem-driven process analysis techniques. This leads to the demand that future BPM capabilities need to be embedded in corporate innovation systems. Finally, the shift from business processes to customer processes considers the empowered digital native and postulates an increased focus on gaining deeper insights into the actual customer experiences along cross-organizational processes.

The Pretense of Knowing the Customer as the Vehicle to Improve Customer Experience: Repositioning Process in the Front Office of Organizations

Jorge L.C. Sanz

Business Analytics Center, National University of Singapore
Global Business Services, IBM Corporation
NUS School of Computing and Business School
21 Heng Mui Keng Terrace, Singapore
jorges@us.ibm.com

Abstract. With the massive availability of personal devices and other digital capabilities, we are experiencing a major change toward what it may be called everyone-to-everyone economy. In the business-to-consumer industries, individuals are endowed with an unprecedented role in the life-cycle of services, products and brands. This means a significant transformation from past organization-centered economy in which companies traditionally defined all markets, established all the moments of value accrual and imposed channels preferences. Simultaneously, a growing volume of customers makes most services and products very difficult to be created or delivered at scalable quality. Thus, as individuals exert their empowerment, the 'digital disruption' is creating not only an opportunity but also raising a major challenge for enterprises.

In an attempt to manage this emerging economy, business-to-consumer organizations have begun intensive efforts on "knowing the customers" as a way to address their expectations better. The aim of enterprises is also to be able to shift "less profitable" customers to cheaper digital channels. And digital devices are a logical channel to attempt individualization and customization. These tall-orders include the goal of understanding "customer behavior" so that early actions can be taken to capitalize individual needs and thus, provide a superior form of "experience". On the other hand, aware of the challenges to get quality personalized services and equipped with more information instantaneously accessible through these devices from different non-mediated sources, customers have more foundations to question their loyalty.

In connection to the above trends, there are a number of important phenomena to investigate. First, we need a social theory that explicates the emerging double-hermeneutics whereby the ambition for profitability, exercised with poorly managed or absent customer co-creation, becomes a self-fulfilling prophecy. Second, the 'pretense of knowledge' (paraphrasing Nobel Prize Von Hayek in his award ceremony speech) to single out customer behavior at an individual level begs for more research on how such a deep cognitive outcome could be accomplished from data artifacts (no matter how big this data might ever be). The third issue is related to

the challenge of delivering promised service personalization in practice, or said in simple words, the question: "suppose you know the customer, so what?" begs for an answer. Finally, current practices toward customer experience also need to be revisited as results cannot be more frustrating. As an example, previously unquestionable measures of "customer satisfaction" are not longer reflecting what customers truly care about to remain loyal.

The amount of data available from customer interactions at multiple touch-points is deepening the gap between Information Management and Process Management. While Information and Process were rarely, if ever, considered in communion and consequential problems are visible, the new everyone-to-everyone economy widens this gap and will make consequences much more severe. Front Office competences in enterprises are focusing on data while Process appears condemned to be a back office concern. This might be a consequence of research communities approaching both Information Management and Business Process Management as 'scientific endeavors' (of the same type as those in Von Hayek's speech) and also a result of organizations focusing on cost reduction and automation of routine operations. Nevertheless, information and Process are two inseparable sides of customer-enterprise interactions.

Process (said in capital P, not only flows) is a fundamental instrument to improve customer experience, integrated with knowledge of the customer (said as an epistemological inquiry, not only analysis of digital footprints). Thus, the reposition of Process in the Front Office is not about integrating flexible back-office operations to deliver the promises of the Front Office, leaning the marketing function, shaving even more cost from call centers, etc. Instead, Process in these competences is an entirely new area of foundational and practical work: aligning organization and customer practices, at affordable costs for all stakeholders involved.

Table of Contents

Long Papers

Short Papers

Long Papers

The focal point of most contributions selected as long papers is the crucial role of models for value-creating organizational development.

The first contribution by Georg Weichhart and Dominik Wachholder kind of overarches this topic. The authors evaluate the potential the S-BPM approach in its entireness (BPM framework and modeling language) offers to overcome conceptual, technological, organizational barriers for interoperability of enterprise systems.

Stephan Borgert and Max Mühlhäuser introduce a formal correctness check facility for models, including error correcting support for modelers. It is based on a derivate of the Parallel Activity Specification Scheme, representing the core of the S-BPM modeling language.

The development of an ontology that helps modeling and interchanging S-BPM specifications is the result of striving for interoperable S-BPM specifications by Kai Michael Höver and Max Mühlhäuser.

Başak Çakar, Onur Demirörs reflect on the transformation of extended Event-driven Process Chains into S-BPM models.

In their paper Patrick Garon, Arnd Neumann and Frank Bensberg propose an S-BPM-based reference model for change management according to the IT Infrastructure Library (ITIL).

Alexander Lawall, Thomas Schaller and Dominik Reichelt describe a novel approach to and a prototype of organizational modeling. The latter is of particular relevance if it comes to embedding process models into an existing organization.

The article submitted by Anton Ivaschenko describes a model and technology for business processes management of Fifth Party Logistics (5PL) providers and its implementation as a multi-agent software-as-a-service solution.

On the Interoperability Contributions of S-BPM

Georg Weichhart[1,2] and Dominik Wachholder[2]

[1] Metasonic AG, Pfaffenhofen, Germany
Georg.Weichhart@metasonic.de
[2] Communications Engineering - Dept. of Business Information Systems,
Kepler University Linz, Austria
{Georg.Weichhart,Dominik.Wachholder}@jku.at

Abstract. Research in *enterprise interoperability* analyzes, describes, and improves the interaction of parts of enterprise systems. Non-interoperability is a situation where either parts of enterprises do not work together at all, or are fully integrated to an extend where the individual parts may not be separated any more. To allow enterprise systems to produce business value its parts need to be interoperable. The S-BPM approach is analyzed with respect to contributions to support interoperability in enterprise systems. Missing support is identified and requirements for tools are derived.

Keywords: Enterprise Interoperability, S-BPM.

1 Introduction

Today's competitive business environment requires sustainable collaboration that is based on the exchange of meaningful information and knowledge. For this reason, interoperability between systems is considered highly relevant as any degree of non-interoperability is likely to lead to problems. Systems which are not compatible on the one hand will not allow interaction among themselves. On the other hand, systems which are fully integrated show lower resilience due to higher probability of overall system failure when an integrated part fails. Research in the domain of "interoperability of systems" in general and "enterprise interoperability" in particular is focusing on identifying and bringing down barriers where systems or system parts are not able to interact [17,8]. The Subject-oriented Business Management Framework and the Subject-oriented Business Modelling Language hold the potential to support interoperability between organizations. Existing S-BPM Tools facilitate information exchange between organizations, humans, and technical systems.

In this paper we analyze the (potential) contributions of the S-BP Management Framework, the S-BP Modelling Language, and existing S-BPM tools to facilitate interoperability. We will also identify further areas of development that are required to be considered in order to facilitate bridging interoperability gaps using the S-BPM framework and/or tools.

The paper is organized as follows. First a framework for interoperability is discussed which is especially tailored to the context of enterprise systems. This

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 3–19, 2014.

discussion is followed by, first, aligning the S-BPM Framework with the interoperability framework, and then aligning existing tools and prototypes with the interoperability framework. This approach allows the identification of gaps that are analyzed and the required further research for fully supporting interoperability is discussed.

2 Interoperability Framework

The concept of interoperability of systems is of importance for sustainable development, and is an ongoing concern for meeting the demands of sustainability. "In today's globally networked environment, one cannot achieve environmental, social/ethical or economic sustainability of any artifact (be it physical or virtual, e.g. enterprise, project, information system (IS), policy, etc.) without achieving ubiquitous ability of the artifact and its creators and users to exchange and understand shared information and if necessary perform processes on behalf of each other in other words, interoperate. Thus, sustainability relies on interoperability, while, conversely, interoperability as an ongoing concern relies on [...] sustainability" [6, p.2].

In the following we discuss (enterprise) systems, followed by systems interoperability in general and continue later with the enterprise interoperability framework [17,8]. The dimensions of the framework will be used for discussing S-BPM support.

2.1 Systems Theory and Enterprise Systems

General System Theory (GST) [4] intends to support the identification of principles that are valid to different scientific disciplines. This is done using abstracting objects to form a *system*. GST builds upon the notion that a system is an organization of connected parts, where each part and the overall system exhibits some behavior. A system is placed in an environment and may have a function and produce some outcome according to a system's objectives [1]. The parts of a system are themselves systems.

Enterprise Systems are organizational systems which may be observed on different levels, ranging from humans undertaking an collaborative enterprise, departments of a single organization to companies being part of supply networks [24].

2.2 Interoperability Approaches

With respect to interoperability approaches, it is important to differentiate between the integration and interoperability of systems as "integration is generally considered to go beyond mere interoperability to involve some degree of functional dependence" [19, p.731]. This functional dependence, however, implies less flexibility and less resilience since it combines the involved systems in order

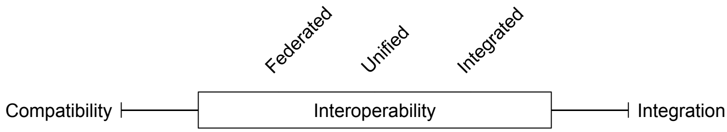


Fig. 1. Compatibility-Integration Continuum

to form a single whole [6]. Integrated systems are sensitive to failures or modifications in individual parts. Small local functional or structural changes may impact distant parts of an integrated system in an unpredictable manner. Interoperability, by contrast, fosters a more loosely coupled approach, where systems remain independent but are coordinated insofar as some collaboration is possible to take place. Functional or structural changes are less crucial as long as the interfaces defined for collaboration are not changed.

Considering this differentiation, interacting systems may be arranged along a continuum that goes from the compatibility to the integration of systems (cf. Figure 1). The concept of interoperability is located in between these two ends. Interoperability may be further distinguished in federated, unified, and integrated interoperability. Here it is important to consider that integrated interoperability and integration is not identical as interoperability merely requires a predefined way of interaction. The separate manifestations are explained in the following in more detail.

Compatibility According to the Oxford dictionary, the term compatible refers to (two) things that are able to coexist without problems or conflict[1]. Two compatible systems therefore do not interfere with each other's functioning, but might not be able to collaborate with each other so as to, for example, exchange information [19].

Interoperability Interoperability lies in the middle of not interoperable (compatible) and fully integrated [8]. Different approaches (federate, unified, and integrated) describe the characteristics of interoperability with respect to the extent of a standardized format available for interaction.

- **Federated approach** Interoperability is established "on the fly" meaning that neither interfaces nor a common formats on a meta-level exist that enable the collaboration of systems. Involved systems are required to identify and adapt to requirements during runtime [8].
- **Unified approach** A common format exists but merely describes the interaction of systems on a meta-level. Diverging concepts are mapped on a semantical level that allows the translation between multiple systems. This approach might encounter some loss of information as the systems' individual needs are not able to be represented directly.
- **Integrated approach** There is a common format which is used by all involved parties. The individual parties, however, are not integrated them-

[1] `http://oxforddictionaries.com/definition/english/compatible`

selves but exist independent of each other. Only the interface is standardized according to the aligned systems.

Integrated system Integration refers to systems that are combined with each other to form a single whole[2]. As a consequence, the modification of one part will have direct effect on other parts. Malfunction of one (sub)system will lead to a breakdown of the entire system with a higher probability than with loosely coupled systems being interoperable.

2.3 Enterprise Interoperability

Enterprise Interoperability may not be considered as unidimensional in terms of underlying issues, but rather involves a problem space composed of two dimensions that are orthogonally aligned to each other [5]. The first dimension is concerned with interoperability barriers that obstruct the collaboration of enterprise systems with respect to their conceptual, technological, as well as organizational disparities. The second dimension addresses the fact that the collaboration can take place at various different levels within an enterprise including data, service, process, and business concerns. The consideration of both dimensions as orthogonal matrix therefore allows to view the different barriers from the perspectives of the separate concerns and vice versa (cf. Figure 2). For instance, conceptual barriers are relevant as such but require to be considered from the viewpoints of the separate concerns separately. The same principle applies to the interoperability concerns, as for example the service level might involve conceptual, technological, as well as organizational interoperability barriers. A more detailed discourse on the problems space and exemplary samples to the separate points of intersection of barriers and concerns are given later in this section.

The third dimension incorporates interoperability approaches and forms, together with the other two dimensions, the solution space (cf. Figure 2). The intersection of all three dimensions therefore gives a set of solutions that encounters interoperability barriers considering a given concern.

2.4 Enterprise Interoperability Barriers

The interoperability of enterprise systems presumes their ability to communicate information that is mutually understood. In this regard, systems encounter different challenges that range from a consistent way of representing relevant information (*information problems*) to transmission principles (*machine problems*) and organizational aspects (*human problems*) as for instance access priviledges. A classification by means of conceptual, technological, and organizational barriers is specified in the following [5].

- *Conceptual Barriers.* Conceptual barriers emerge once the information to be exchanged among enterprise systems is represented in different ways. Consensus upon the syntactic representation of information as well as semantic

[2] http://oxforddictionaries.com/definition/english/integrate?q=integrability%25255C#integrate__14

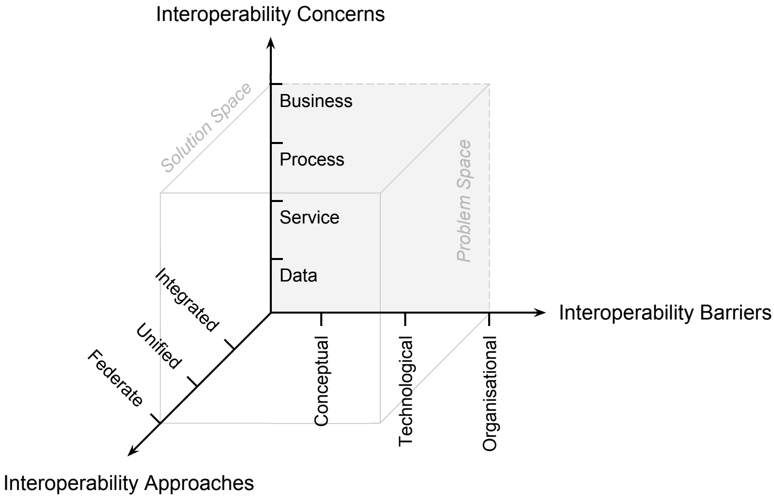


Fig. 2. Dimensions of Enterprise Interoperability [23]

concepts used in this context therefore is crucial in order to achieve conceptual interoperability. UEML [21], for example, provides an approach to align diverging enterprise models by allowing to map their different syntaxes in use. Differences on a semantic level are possible to be tackled by using ontologies.

– *Technological Barriers.* As the collaboration of enterprise systems requires their communication and the exchange of information, technological barriers are crucial to be considered so as to achieve interoperability. In this case, not the representation of information is addressed but the way of how this information is processed and finally distributed among enterprise systems. Typical barriers are, for example, different IT architectures & platforms, operating systems, encodings, transmission protocols, and standards. In spite of the attempt of standardization, technological barriers still exists as for instance different versions of a standard or protocol are incompatible with each other.
– *Organizational Barriers.* As compared to conceptual and technological barriers, organizational barriers are less technical oriented but address disparities in organizational structures and management techniques of collaborating enterprises. It is for example crucial to consider differences in implemented responsibility and authority concepts that govern who is responsible for what (e.g., data, service, etc.) and who is authorized to perform which tasks (create, modify, and maintain data, processes, etc.).

2.5 Enterprise Interoperability Concerns

In order to achieve interoperability among enterprise systems it is not sufficient to consider the abovementioned barriers only. Orthogonal to these barriers, interoperability concerns are crucial since they describe where interoperability actually takes places. Chen [5] therefore distinguishes between interoperability concerns on a data, service, process, and business level.

- *Data.* The interoperability of data, both analog and digital, plays an important part in the context of collaborating enterprise systems. It not only facilitates the mere exchange of data (technological barriers) but also the comprehensibility in terms of its representation (conceptual barriers) as well as its handling on an organizational level (organizational barriers). Thus, the existence of diverse data structures, query languages, right up to incompatible security and permission policies might negatively affect the interoperability on a data level. The ability of systems to collaborate on a data level provides the basis for the remaining concerns.
- *Service.* The interoperability of service addresses the proficiency of enterprise systems to collaborate with each other on a functional level. In this context, the term *service* is understood in a more broader sense with respect to any function that is provided by the collaborating partners within an enterprise. It is therefore not limited to just the technical part of, for instance, a service-oriented system architecture. Challenges so as to achieve interoperability arise once services are described differently (conceptual barriers), exhibit different granularities (technological barriers), or underlie diverging policies as related to the management of services (organizational barriers).
- *Process.* A process defines the arrangement of separate services (functions) that in collaboration serve a common business need. Typically, an organization has several of such processes in place which are described using (different) process description languages. Process interoperability intends to eliminate barriers that emerge when composing processes in order to perform verification, simulation, and execution tasks collaboratively. Diverging syntactical and semantical constructs (conceptual barriers), process behaviors (technological barriers), and business process behaviors (organizational barriers) are typical examples that obstruct the collaboration of enterprises on this level.
- *Business.* The interoperability of business describes organizations that have a mutual understanding of how business is performed. It refers to an aligned mode of operation among organizations where diverging constructs related to business issues are negotiated and mapped among each other. Barriers to interoperability of business might range from diverse visions and strategies (conceptual barriers) and different support by means of ICT infrastructure (technological barriers) to issues as related to incompatible organization structures and decision-making processes (organizational barriers).

2.6 Problem Space of Enterprise Interoperability

Ducq et al. give a more detailed discourse on the problem space of interoperability barriers and concerns in their paper by attributing non-exclusive examples to each point of intersection [7]. Table 1 summarizes these findings. Interoperability barriers are given in the columns and interoperability concerns are in rows respectively.

Table 1. Problem Space of Enterprise Interoperability [7]

Barriers / Concerns	Conceptual	Technological	Organizational
Business	Visions, strategies and culture; Business semantics; Business syntax	IT requirement fulfilment; Degree of computerization	Methods of work; Organization structure; Legislation
Process	Process semantics; Process syntax; Process content	Process behavior	Business process behavior;
Service	Service semantics; Service syntax; Service content	Service granularity	Service management
Data	Data semantics; Data syntax; Data content	Exchange format	Classified information; Information ownership

In addition to the two orthogonally aligned dimensions of interoperability barriers and concerns, the timing of the envisioned solution is of relevance when seeking to support the creation of such an solution. Solution timings refer to the circumstances when interoperability problems are tackled. A-priori solutions are approaches that allow to anticipate problems and to overcome barriers before systems are build. A-posteriori solutions are approaches that allow to identify and correct problems after they occur in the running system [16].

3 S-BPM

The acronym “S-BPM” is used for [11]:

- Subject-oriented Business Process *Management Framework*,
- Subject-oriented Business Process *Modeling Language*, and
- Subject-oriented Business Process *Modeling Activity*.

For the discussion on the support of interoperability, these manifestations will be distinguished. The Management Framework defines a set of activity bundles (or phases) for organizational development and business process management. These activity bundles are typically “carried out along a feedback control cycle composed out of the phases: analysis, modeling, validation, optimization,

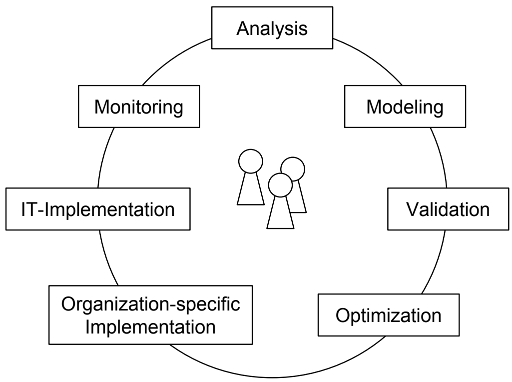


Fig. 3. S-BP Management Framework Activity Bundles

organization-specific implementation, IT-implementation, monitoring" [11, p.30] (cf. Figure 3). S-BP Modeling thereby is considered as an integral part of these phases. However, modeling is only one activity within the framework which (naturally) has a larger scope.

3.1 S-BPM Framework Contributions to Interoperability

In the following discussion we will focus on contributions of the Subject-oriented Management Framework [11] to interoperability. This is done, by using the interoperability framework described above. Support by the Management Framework and S-BPM Tools in general will be identified and mapped to the problem space of interoperability. Subject-oriented Business Process Modeling will be used to refer to the *activity of modeling* and not the modeling environment neither the modeling language itself.

Conceptual Barriers. The following list shows S-BPM Framework support for interoperability concerns regarding conceptual barriers. As mentioned above, conceptual interoperability refers very much to the semantics and syntax of the different concerns (cf. Table 2).

– *Conceptual/Business.* Per-se no support for (e.g. cross-organizational) business interoperability on conceptual level is given directly.
  However, S-BPM Framework distinguishes multiple types of stakeholders which are addressed explicitly and collaboratively work on conceptual interoperability on business level may be realized indirectly through the work of these stakeholders.
– *Conceptual/Process.* The analysis and modeling activities require stakeholders and actors (the process participants) to provide coherent models of their tasks. By using a common language (S-BPM) conceptual interoperability is

facilitated. Validation activities check the process and might uncover conceptual process interoperability problems.
- *Conceptual/Service.* During the validation phase, stakeholders are in charge of checking processes, to see if these realize the desired service. This implies that in S-BPM the service is realized through the underlying processes. Service execution may additionally be monitored through Key Performance Indicators. Establishing key performance indicators in the company, provide additional conceptualizations in terms of what is important, and how performance is measured [11].
- *Conceptual/Data.* No explicit support for overcoming data syntax and semantics issues is provided by the framework. However, data issues are addressed to a limited extend in the IT-Implementation phase of S-BPM.

Technological Barriers. The following list shows S-BPM Framework support for interoperability concerns regarding technological barriers. Interoperability with respect to technology is addressed mainly in S-BPM Framework's IT-Implementation phase.

- *Technological/Business.* Using the framework will result in models that describe manual and automated tasks. Interoperability issues by technology in business will be addressed when applying the framework.
- *Technological/Process.* Explicit process models help to communicate an exchange information about processes. The IT-Implementation phase will result in technological support for processes.
- *Technological/Service.* No explicit support for determining or providing technical services (eg. web services) is given by S-BPM. In S-BPM software services may be called from within subject's states but the framework leaves this to be supported by tools.
- *Technological/Data.* No explicit support for modeling Business Objects is provided by the S-BPM framework. It leaves this to be supported by tools.

Organizational Barriers. The following list shows S-BPM Framework support for interoperability concerns regarding organizational barriers:

- *Organizational/Business.* Organizational structures are not part of the S-BP Models per-se. To some limited extend organizational structures are mapped using "Subject carrier groups" [11] to implement functional roles in S-BPM. Support for organizational change (management) and cross- organizational business issues is missing.
- *Organizational/Process.* S-BPM models do make processes transparent by definition.
- *Organizational/Service.* Service management has to be described in separate S-BPM models. The monitoring phase facilitates management of the quality with which a service is provided [11].
- *Organizational/Data.* Organizational management of data is not explicitly supported by S-BPM.

3.2 Tool Contributions to Interoperability

In addition to the above support by the S-BPM framework, a number of tools exist, which provide support, also for interoperability However, for this discussion we do not constrain the discussion to the commercially available tools (e.g., Metasonic Suite[3]) or features, but also include research prototypes. These prototypes may be stand-alone or used in conjunction with the suite.

Conceptual Barriers. The following list shows S-BPM Tool support for interoperability concerns regarding the conceptual barrier:

- *Conceptual/Business.* Bastarz and Halek are using the smart tool with the smart4sense2act [2] to facilitate conceptual clarification of business systems.
- *Conceptual/Process.* Metasonic Build and Proof as S-BPM design system and process validation system respectively, facilitate the clarification of process related concepts.
- *Conceptual/Service.* As services (from the S-BPM point of view) are composed of processes, the clarification of processes (see above) leads to clarification of services. Metasonic Touch which is based on Comprehand [18] facilitates the articulation work of all process participants in order to clarify the work to be done.
- *Conceptual/Data.* No support for clarification of a Business Object's semantics and content is available today.

Table 2. *S-BPM Framework* Support for Interoperability in the Problem Space

Concerns \ Barriers	Conceptual	Technological	Organizational
Business		All Framework Activities	
Process	Analysis, Modeling, Validation Framework Activities	IT- and Organization Implementation Framework Activities	All Framework Activities
Service	Validation Framework Activity		Monitoring Framework Activity
Data			

Technological Barriers. The following list shows S-BPM Tool support for interoperability concerns regarding the technological barrier. Since S-BPM focuses on the modeling and implementation of workflows, any S-BPM execution (workflow) engine supports technical interoperability through an integration interoperability approach.

[3] http://www.metasonic.de/metasonic-suite

- *Technological/Business.* In the IT-Implementation phase of the S-BPM Framework, the computerization of business processes is realized. The configuration of the workflows will clarify the interface between different tools and human operators on a technology level.
- *Technological/Process.* Metasonic Build and Proof facilitate interoperability of processes with respect to technology through an "integrated interoperability" approach (see above) [11].
- *Technological/Service.* Technical Services like web services may be integrated in S-BPM workflows using refinements in Metasonic Flow.
- *Technological/Data.* Any S-BPM workflow engine may be used as middle layer for establishing interoperability of external tools. By including external data sources in the business process some interoperability between software systems through federation using the process may be realized. This means, that as soon as conceptual/data interoperability is realized, the S-BPM Tools allow to map data to business objects and facilitate data flows from and to other tools.

Organizational Barriers. The following list shows S-BPM Tool support for interoperability concerns regarding organizational barriers:

- *Organizational/Business.* No direct tool support is currently available.
- *Organizational/Process.* The S-BPM design tools naturally support clarification of interaction within processes. Metasonic Touch and Comprehand [18] facilitate the communication and articulation of work related information between process participants beyond pure modeling and design of processes. A research prototype developed in the project jCPEX! [15] enables dynamic routing of business objects to different actors of the same role. The research project results have demonstrated the feasibility of this approach across companies. jCPEX! facilitates standardized interfaces for cross-organizational processes.

  Another research prototype demonstrates the possibility of flexible BPM by making use of ad-hoc process deviations [20]. This allows to maintain interoperability besides changed requirements and allow implementation of process agility [12].
- *Organizational/Service.* KPI management featured in Metasonic Suite, supports monitoring and management of services [11] through supporting the development of performance indicators that clarify goals of a service.
- *Organizational/Data.* No direct support for rights and permissions of access to business objects is given.

3.3 Missing Support

So far, we have analyzed and brought together two existing frameworks. Subject-oriented Business Process Management Framework and existing S-BPM tools and research prototypes do provide support for interoperability in the context of the Enterprise Interoperability Framework. However, as S-BPM has not been designed to address interoperability issues explicitly, a few gaps exist.

Table 3. *S-BPM Tool* Support for Interoperability in the Problem Space

Concerns \ Barriers	Conceptual	Technological	Organizational
Business	smart4sense2act	S-BPM Design Tools	
Process	S-BPM Design and Execution Tools	S-BPM Design and Execution Tools	Metasonic Touch, Comprehand, S-BPM Design Tool, jCPEX!, Ad-hoc Deviations
Service	Metasonic Touch, Comprehand	S-BPM Execution Tool	KPI Management Tool
Data		S-BPM Execution Tool as Enterprise Bus	

Table 4 shows existing support for enterprise interoperability by the S-BPM framework [11], existing tools, and research prototypes respectively [3,15,20,11]. In this table also the solution approach category (integrated, unified, federated) is given. However, the borders between the categories are blurry.

Business Concern: smart4sense2act allows to negotiate a common abstract picture of the business context of processes. — The application of the S-BPM framework provides a common ground for aligning manual and technical tasks.

Process Concern: Using the S-BPM framework and the Tools allow the integration of process related concepts. — For technological barriers support for integration is also provided. — A number of tools exist that allow to overcome organizational barriers by integrating different processes. Tools like Comprehand facilitate the negotiaton of processes among process participants. Metasonic Touch makes use of a common language and hence facilitates unification.

Service Concern: The framework provides through the S-BPM notation a unified language that allows to specify service descriptions. Metasonic Touch and Comprehand allow to negotiate the processes that realise a service. — The uses of the execution environment integrates the different services (services as finer granular process parts). It also allows to provide integrated processes that realise a service. — S-BPM provides a common language to overcome organizational interoperability on service level. Through the use of KPIs the service's outcome and output might be specified in depth.

Data Concern: The S-BPM tools allow to integrate other software systems allowing to exchange data between systems along processes.

It has to be noticed that there is some minor conceptual disagreement between the interoperability framework and S-BPM. In the later, a service is defined by processes which result in delivering value to a customer, the former is more vague in its definition of service. However, the former conceptualization leans toward defining a process as alignment and structuring multiple (more fine-grained) services. For this discussion, however, this disagreement does not lead to interoperability problems between the two frameworks.

Table 4. S-BPM Support for Interoperability in the Problem Space

Concerns \ Barriers	Conceptual	Technological	Organizational
Business	smart4sense2act[3]	Framework (All Activities)[2] S-BPM Design Tools[2]	
Process	Framework (Analysis, Modeling,Validation)[1] S-BPM Design and Execution Tools[1]	Framework (IT-, Organization Implementation)[1] S-BPM Design and Execution Tools[1]	Framework (All Activities)[1] Metasonic Touch[2], Comprehand[3], S-BPM Design Tool[1], jCPEX![1], Ad-hoc Deviations[1]
Service	Framework (Validation)[2] Metasonic Touch[2], Comprehand[3]	S-BPM Execution Tool[1]	Framework (Monitoring)[2] KPI Management Tool[2]
Data		S-BPM Execution Tool as Enterprise Bus[1]	

1 . . . integrated; 2 . . . unified; 3 . . . federated;

For any interoperability support that attempts to fill the gaps, two modes of operation have to be considered. *A-priori* support facilitates the clarification of interoperability issues during design-time of systems. *A-posteriori* support enables negotiation and agreement on appropriate steps of involved parties after an interoperability issue has emerged.

Better support by S-BPM framework and tools is required for overcoming the following barriers:

- *Organizational/Data.* "The structures for assigning rights to data (different rights for different partners); Differences in which an information is to be regarded as classified with respect to the collaboration partner" [8, p.850]. With respect to organizational networking and interoperability on the level of data, trust management, security, and legal issues require further research [22]. This is especially true when considering trust transitivity and determining how to handle permissions once information is passed from one actor to the next which, in turn, further passes the data to another actor. A transparent approach is needed which supports individual actors and applications to not (incidentally) pass classified information to actors who are not authorized to access that data.
- *Conceptual/Data.* "Coverage, i.e. content, of the respective data representation; Heterogeneous data format and structure; Data meaning disagreements" [8, p.850]. Interoperability support for conceptual data interoperability is also a topic of ongoing research. There are ontology and semantic web approaches

(e.g., based on OWL-S) [22] for a-priori mapping of different data types and fields.

– *Organizational/Business.* "The legislative requirements that influence different actors; How enterprises are organized on a high level; High level differences regarding how work is performed in the organizations" [8, p.850]. While business process approaches offer some possibility to overcome organizational issues on process level, a needed high-level overview of organizational differences requires more specific support.

In all cases above, when considering dynamic environments and a-posteriori support the challenge gets even greater. In the case of a need for action because of interoperability issues in running business situations, content and communication needs to be intertwined to support a negotiation of resolution strategies. The dynamics of business needs to be respected by empowering users to act flexible in situations. An organizational learning environment, which facilitates data and knowledge exchange across companies would foster organizational interoperability through supporting communication, negotiation, and providing an organizational memory for interoperability [25].

For example, the SUddEN environment [25] facilitates interoperability on supply-network level, by making use of key performance indicators (KPI) and supporting the collaborative developments of a network-wide performance measurement system (PMS). However, with respect to the enterprise interoperability framework, this environment misses support on process and data concern level.

An organizational learning environment to support interoperability meeting all concerns, needs, in addition to an organizational point of view, an actor specific view (not only in order to follow the logic of S-BPM). In the actor point of view, it must be assumed that knowledge is distributed within the company or even across companies. This naturally leads to situations where it is not possible to establish a global (over-)view [11]. In order to keep workflow participants (to a large extend) autonomous (but interoperable), advanced support for organizational interoperability is needed on business level. Users may change their processes if required (by them). Giving this freedom is a potential source of non-interoperability and hence comes with great responsibility. The approach to be researched has to enable users to study the impact of changing their behaviors autonomously. A potential technology to do so, but which is not put into use with S-BPM, is agent-based simulation [26,9]. Here autonomous software components (called agents) represent users. For facilitating learning in this context, the agents would be assigned multiple subject-roles [10]. This allows independent interaction of users with a virtual environment which is build on actual process models. Users are enabled to explore the impact of changing a behaviour by allowing individuals to change a subject-behaviour-diagram, and then run an agent-based simulation of the organization in order to identify the impact of the (proposed) changes to other users. Multi-agent simulation has already shown to be useful for planning in distributed environments and providing a high-level overview of supply networks [14].

For making the above described environment useful for a-posteriori organizational interoperability, support for technological interoperability has to be build-in. The technical interoperability needs to address two different, independent aspects. Making existing S-BPM tools interoperable on technical level with the environment allows to extend the learning environment. Comprehand [18] for example, enables knowledge transfer between users on a conceptual level. This tool supports interoperability required in a learning environment for interoperability. The second aspect is to make the learning environment's agents interoperable with existing systems for supporting the agent's decision making during simulation. The first technical interoperability extends the learning environment with other tools to facilitate learning organizational interoperability, the second enables to access information required for decision making during learning.

4 Conclusions

S-BPM, although a process management framework, promises to support interoperability, especially in contrast to integration. Other process frameworks like ARIS [13] aim at providing a global "world-view" based on a global control flow, where IT, actors, and organizational tasks are parts of a fully integrated system. S-BPM uses a different approach as it puts the individual actor in the center of the organizational models [10]. S-BPM assumes independent agents interacting and communicating with each other.

S-BPM does deliver a communication and subject oriented view where process participants are loosely connected through the exchange of business objects and messages. This provides some freedom for the participants, who may optimize their individual task structures. This freedom is a source of non-interoperability and hence requires some additional collaborative effort to support overcoming interoperability barriers.

Overall we have identified three larger areas where the S-BPM Framework should be extended to deliver a full approach to support enterprise interoperability.

- Trust and security needs to be handled for interoperability, especially when considering multiple involved systems which share data.
- Interoperability concerning the enterprise's business requires support, while maintaining S-BPM's autonomous agents world-view.
- Interoperability concerning the data level for different IT systems is currently under developed.

Based on existing works [23,24,14,25], we have additionally described an approach to foster interoperability supporting a-posteriori exploration and learning with respect to organizational interoperability. That approach will make use of agent technology in order to stay close to the S-BPM logic.

Acknowledgement. The research leading to these results has received funding from the European Commission within the Marie Curie Industry and Academia

Partnerships and Pathways (IAPP) programme under grant agreement n^o 286083. For more information on the IANES project see `http://www.ianes.eu`.

References

1. Ackoff, R.L.: Towards a system of systems concepts. Management Science 17(11), 661–671 (1971), http://mansci.journal.informs.org/content/17/11/661.abstract
2. Bastarz, F., Halek, P.: Seeing the wood for the trees again! smart - a holistic way of corporate governance offering a solution ready to use. In: Stephanidis, C. (ed.) Universal Access in HCI, Part IV, HCII 2011. LNCS, vol. 6768, pp. 187–194. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-21657-2_20
3. Bastarz, F., Halek, P.: smart4sense2act: A smart concept for systemic performance management. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 109–114. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-23471-2_8
4. von Bertalanffy, L.: General System Theory - Foundations, Development, Applications, 17th edn. George Braziller, New York (1969)
5. Chen, D.: Framework for enterprise interoperability. In: Congrès International de Génie Industriel, CIGI 2009 (2009)
6. Dassisti, M., Jardim-Goncalves, R., Molina, A., Noran, O., Panetto, H., Zdravković, M.M.: Sustainability and interoperability: Two facets of the same gold medal. In: Demey, Y.T., Panetto, H. (eds.) OTM 2013 Workshops 2013. LNCS, vol. 8186, pp. 250–261. Springer, Heidelberg (2013)
7. Ducq, Y., Chen, D., Doumeingts, G.: A contribution of system theory to sustainable enterprise interoperability science base. Computers in Industry 63(8), 844–857 (2012), http://www.sciencedirect.com/science/article/pii/S0166361512001248
8. Ducq, Y., Chen, D., Doumeingts, G.: A contribution of system theory to sustainable enterprise interoperability science base. Computers in Industry 63(8), 844–857 (2012); Special Issue on Sustainable Interoperability: The Future of Internet Based Industrial Enterprises
9. Fioretti, G.: Agent-based simulation models in organization science. Organizational Research Methods 16(2), 227–242 (2013)
10. Fleischmann, A., Kannengiesser, U., Schmidt, W., Stary, C.: Subject-oriented modeling and execution of multi-agent business processes. In: The 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (2013)
11. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management. Springer (2012)
12. Heininger, R.: Requirements for business process management systems supporting business process agility. In: Oppl, S., Fleischmann, A. (eds.) S-BPM ONE 2012. CCIS, vol. 284, pp. 168–180. Springer, Heidelberg (2012)
13. IDS Scheer AG: Handbuch zur ARIS-Methode - Version 5. Saarbrücken (May 2000)
14. Karageorgos, A., Mehandjiev, N., Weichhart, G., Hämmerle, A.: Agent-based optimisation of logistics and production planning. Engineering Applications of Artificial Intelligence 16(4), 335–348 (2003)
15. Meyer, N., Feiner, T., Radmayr, M., Blei, D., Fleischmann, A.: Dynamic catenation and execution of cross organisational business processes - the jcpex! approach. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 84–105. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-23135-3_5

16. Naudet, Y., Guedria, W., Chen, D.: Systems Science for Enterprise Interoperability. In: International Conference on Interoperability for Enterprise Software and Applications China, IESA 2009, pp. 107–113 (2009)
17. Naudet, Y., Latour, T., Guedria, W., Chen, D.: Towards a systemic formalisation of interoperability. Computers in Industry 61, 176–185 (2010)
18. Oppl, S., Stary, C.: Facilitating shared understanding of work situations using a tangible tabletop interface. Behaviour & Information Technology, 1–49 (2014)
19. Panetto, H.: Towards a classification framework for interoperability of enterprise applications. International Journal of Computer Integrated Manufacturing 20(8), 727–740 (2007)
20. Rothschädl, T.: Ad-hoc adaption of subject-oriented business processes at runtime to support organizational learning. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 22–32. Springer, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-29133-3_2
21. Vernadat, F.B.: UEML: Towards a unified enterprise modelling language. International Journal of Production Research 40(17), 4309–4321 (2002)
22. Vernadat, F.B.: Technical, semantic and organizational issues of enterprise interoperability and networking. Annual Reviews in Control 34(1), 139–144 (2010), http://www.sciencedirect.com/science/article/pii/S1367578810000155
23. Wachholder, D.: Enabling Context-sensitive Behavior Adaptation of Information Resources in Dynamic Environments: Grounding Interoperability on Bigraphical Structures. Master's thesis, Johannes Kepler University Linz, Dept. Communications Engineering - Business Informatics (2013)
24. Weichhart, G.: Supporting interoperability for chaotic and complex adaptive enterprise systems. In: Demey, Y.T., Panetto, H. (eds.) OTM 2013 Workshops 2013. LNCS, vol. 8186, pp. 86–92. Springer, Heidelberg (2013)
25. Weichhart, G., Feiner, T., Stary, C.: Implementing organisational interoperability the sudden approach. Computers in Industry 61(2), 152–160 (2010), http://www.sciencedirect.com/science/article/pii/S0166361509002048
26. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review 10(2), 115–152 (1995)

Formal Based Correctness Check for ePASS-IoS 1.1 Process Models with Integrated User Support for Error Correcting

Stephan Borgert and Max Mühlhäuser

Technische Universität Darmstadt
Dept. of Computer Science
64289 Darmstadt, Germany
{stephan,max}@tk.informatik.tu-darmstadt.de

Abstract. To ensure the correctness of business process models, automatic and manual methods are applied. Since the manual checks are time consuming and expensive, the automatic methods should be as effective as possible. An established verification check is the check for the interaction soundness, i.e. the process model can be executed without deadlocks. Normally, these approaches compile the graph based models to expressions of a formal language which is passed on to a model checking tool for verification. The drawback with this methods is that the results are hard to use for analyzing the causes of errors. In this paper, we present an integrated approach that is able to find important error patterns, and supports the user in correcting errors while still having a high performance.

Keywords: Formal Verification, Interaction Soundness , Subject Oriented Modeling, PASS, π-Calculus.

1 Introduction

A model designer of the Metasonic[19] suite can move from a business process to a complete application in four steps: 1) Model 2) Evaluate 3) Integrate 4) Execute. The correctness of the models is checked in the steps 1) and 2) and in accordance to [[10], p.312, Sect. 16.3] two aspects of correctness are distinguished

- A system must have certain properties, e.g. livelock free, deadlock free which are independent of the application. This is implicit correctness. A deadlock free system is also called an *interaction sound* system in the context of service compositions. A formal definition is given in [25].
- A specified system must do what a designer has intended. This is explicit correctness.

Explicit correctness checks what does the process and implicit correctness checks how the process does it. To be exact: Does the process run without errors. Therefore, explicit correctness has a higher priority than implicit correctness

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 20–40, 2014.

but explicit correctness is only given when implicit correctness is fulfilled. If a process is running into a deadlock, this should not fit the designers intents. Manual interrupts are no solution for this problem. Of course, the deadlock would be solved but if the model designer knows about a deadlock, the solving of the deadlock should be part of the process model. The same applies for timeouts. When they are only used to solve a deadlock but not to terminate the process model in a proper way, explicit correctness is not given. Explicit correctness is checked in step 2) by manual simulation, business users execute the model on an abstract level and simulate the behavior of the subject providers. Checking for explicit correctness has to be done manually since current systems are not able to interpret the semantics of the models. The effort for this should be as low as possible for three reasons. 1) Manual simulation is slow, 2) employees are expensive and 3) they still can overlook errors which could become expensive if they occur during runtime. To minimize the effort computer supported tools are used. The Metasonic Suite includes a check for structural soundness of process models. It checks for properties like: Is there exactly one start node per subject, is there at least one end node per subject, is there a path from every node to an end node and so on. This is very helpful to avoid many errors but there are still more error types the tool can not find. Since subject and service oriented business process models are also models of distributed systems, the verification for interaction soundness is a common correctness check.

The “Parallel Activity Specification Scheme” (PASS[9,12]) is the modeling concept implemented in the Metasonic Suite. An extended version of PASS was introduced in [11]. We denote this version as extended PASS or ePASS and a variant of it, focusing on support the internet of service paradigms, as ePASS-IoS[5].

Distributed systems suffer from 3 inherent issues. Firstly, non deterministic behavior can occur: every time a distributed system is executed, the behavior of the system can differ. The behavior is the interaction behavior i.e. the communication protocol among the involved sub systems. The reasons for this are the following: Not all subsystems of a distributed system work with the same speed at all run times and further more the time needed for conveying the messages can differ. Secondly, accessing the global state of the system is impossible. Monolithic systems are easier to analyze because e.g. debug mechanisms like break points can be used to detect errors. The same mechanisms are not suitable for distributed systems because even if it were possible to place break points in every sub system, they would not meet at the same time per system run because of the nondeterministic behavior. Therefore it would not be possible to stop the system in every run at the exactly same global state. Finally, the system time will be different on every subsystem: the system times will not be set to the exact same time. This is an important point for time critical systems, for business processes it is of less significance. Hence, it does not have to be considered for our purposes. All in all, error detection in distributed systems can be very hard and therefore the methods like verifying interaction soundness are supposed to be useful for ePASS process models as well.

The authors in [7] distinguish interconnection modeling and interaction modeling. Modeling the internal behavior of the participants and relating them with message flows is defined as interconnection modeling. Modeling the interaction behavior between the different participants is called interaction modeling. Therefore ePASS falls in the category interconnection models. BPMN supports interconnection and interaction modeling techniques. Both techniques support the modeling of choreographies but the expressiveness is different[6]. An example for interaction modeling can be found in [7].

Anti-patters were derived and used in [3,4,7]. In [3] anti-patterns are defined as "Anti-patterns describe undesirable constructs that may introduce errors or inefficiencies." The authors in [4] have stated: "Each anti pattern declaratively describes a violation scenario." In [7] a set of 8 choreography modeling anti-patterns were identified. These patterns were collected by observing people modeling BPMN and "can be observed in a large number of interconnection models." The anti-patterns can be expressed by interconnection models.

The paper is structured as follows: Firstly, we briefly introduce the ePASS-IoS language elements. Next, we motivate our decision to use π-calculus as formal foundation and introduce how we exploit it for the purpose of interaction soundness verification. After giving implementation details of the demonstrator, we evaluate our approach by applying it on the 8 different choreography anti-patterns. Then, the performance of the approach is discussed and finally, the summary and conclusion are given.

2 Verification of ePASS IoS Models

2.1 ePASS-IoS

Graphical Syntax. The graphical elements of the extended Parallel Activities Specification Scheme for the Internet of Services (ePASS IoS) 1.1 have been introduced in [5]. The graphical language set has been explained by graphical examples. We introduce a simplified notation in figure (1), in order to save space depicting the examples.

Subjects are an abstraction of actors. The execution of the subjects is done by concrete actors that are humans, machines or software services. These concrete actors are called subject providers. Rules have to be defined for every process to link the subject providers to the subjects during runtime. This late binding mechanism also has the advantage of decoupling the process models from the enterprise. More information can be found in [11,5]. Subjects are connected by unidirectional channels, which enables communication among subject providers by exchanging messages.

As shown in figure (1), we draw the internal behavior directly into the subjects. This is in contrast to other ePASS systems where two layers are used.

Send, **Receive** and **Action** are the basic activities. Send and Receive states are denoted with `(S)` and `(R)`, while the name of the action is written directly into the state symbol. The transition labels are *exit conditions. Action* states have the results of the action as *exit conditions*. When Action states only have

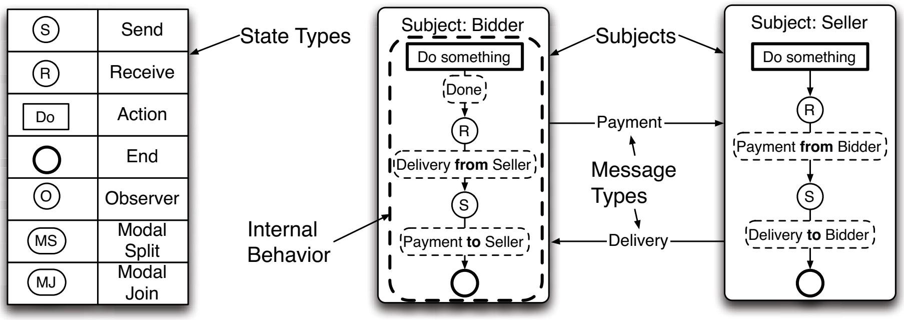


Fig. 1. Simplified notation of the ePASS-IoS 1.1 language elements

one exit condition, the exit condition label can be omitted. The exit conditions of **Send** states are (<Message-Type>, <Receiver>), written as *<Message-Type>* ***to*** *<Receiver>*. The exit conditions of **Receive** states are (<Message-Type>, <Sender>), written as *<Message-Type>* ***from*** *<Sender>*. The internal behavior of subjects have exactly one **Start** state and can have an arbitrary number of **End** states. Start states are denoted by a bold border. Every state can be a start state. The **Observer** manages interrupts and exceptions. In case of such an event, the control flow is lead to an alternative behavior. Start states can have an arbitrary number of out-transitions and the End state can have an arbitrary number of in-transitions. All other states can have an arbitrary number of in- and out transitions. The semantics of this pattern is an exclusive choice split and join. In order to fulfill a task, it is often the case that certain activities have to perform and others could optionally be performed in addition . This is why `(MS)` and `(MJ)`, which are called **Modal split** and **Modal join**, are introduced. They form a combination of an AND- split and join pair and an OR- split and join pair. These symbols are specified e.g. in the BPMN 2.0 [22] specification. Since the concerning semantics is an interleaving semantics, the parallel execution does not have to be performed in parallel. Thus, the human actors are still able to perform these kinds of behavior. In addition to the normal control flow transitions, **timeout transitions** can be used in PASS / ePASS. We would depict timeout transitions with a dashed line. Since the anti-patterns do not take timeouts into consideration, we are not using them in this paper. How ePASS timeouts can be handled concerning the formal verification can be found in previous work [5].

2.2 π Calculus as Formal Foundation

A formal semantics for the language elements is needed for the purpose of formal verification. A common way to do this is by encoding the elements with a suitable formalism. Many approaches of the BPM community use petri nets and its variants for it. Others use process algebras, abstract state machines, or other state and transition based diagrams.

The used formalism should fit the verification requirements as well as possible. For the purpose of verifying ePASS IoS models for interaction soundness, the following requirements have been stated by us:

- As mentioned above, the purpose of the verification method is to perform a check for interaction soundness. It has to detect deadlocks.
- Further more it has to be internal and not external. An external one delegates the verification task to model checker tools which verify the models and return the results. The advantage of this approach is that it is easier to get a result. The disadvantage is, the results are not easy to interpret. The model checkers input consist of roughly two parameters. The first one is the model which has to be checked and the second one is the specification the model has to meet. The specifications are usually formulated in a formula from mathematical logic and therefore allows general purpose model checking. The results are difficult to interpret and it is not easy to analyze the reason of error sources automatically. The results have to be analyzed and new specifications have to be formulated; they have to be given to the model checker again and again until the error sources are found. A user friendly support for business users is hard to implement and therefore an internal check is required. In consequence, verification algorithms must be implemented into the modeling system. This way a user friendly analysis can be achieved.
- The correctness check must not take to long. We required that the check has to be faster then 15 seconds on an average computer.

Different diagram types can serve to model distributed systems like automata, state charts , FSM etc. One, very simple type, is the Labeled Transitions System (LTS).

Definition 1 (Labeled Transition System [14]). *Let* Act *be a fixed set of actions. A labeled transition system* $LTS = (\textsc{Proc}, \rightarrow)$ *over* Act *consists of*

- *A set* Proc *of states and*
- *A set* $\rightarrow \subseteq \textsc{Proc} \times \textsc{Act} \times \textsc{Proc}$ *of transitions between states.*

To model distributed systems, three different types of actions were introduced: send, receive and the internal action τ. These three actions are the base language elements of ePASS but the internal action τ is specified more precisely in the internal behavior. They can be converted to τ actions in order to anonymize the internal behavior. Thereby, an external representation is obtained which can still be used for verification purposes like deadlock, livelock checks or checks for equivalent interaction behavior of different subjects. Modeling distributed

Table 1. Syntax of the π-calculus. $\{x, y, z\}$ are *names* and τ the hidden action. The term $\overline{\mathrm{x}}\langle\tilde{y}\rangle$ denotes sendig $y_1, \ldots, y_n$ messages via channel x while $\mathrm{x}(\tilde{y})$ denotes receiving $y_1, \ldots, y_n$ messages via channel x.

Processes P, Q ::=		$\mathbf{0}$	Inactive Process
	|	$\pi.P$	Prefix
	|	$P + Q$	Sum
	|	$P \mid Q$	Parallel
	|	$(z)P$	Restriction
	|	K	Identifier
Prefixes π ::=		$\overline{\mathrm{x}}\langle\tilde{y}\rangle$	Send
	|	$\mathrm{x}(\tilde{y})$	Receive
	|	τ	Hidden Action

systems only with LTS is a very challenging task because the number of states of a distributed system grow very fast with the number of subsystems and the number of states of each subsystem. In this case a good choice to model the system is a process algebra. Process algebras consist of a set of actions and a set of operations defined on these actions. One of the simplest process algebras is the Calculus of Communicating Systems (CCS)[21]. Its actions are also send, receive, and τ. A more sophisticated and the most popular process algebra is the π-calculus[20,26]. It has as additional feature to CCS like a channel construct and a concept of mobility of channels, which allows dynamic restructuring of process models. The syntax of the channels is defined in table (1) and the semantics in table (2). The formalism should be as simple as possible to avoid needless complexity. Since CCS is simpler it should be the first choice. On the other side, two good reasons led to the choice for the π-calculus as formal foundation: 1) The π-calculus is more expressive than CCS. Hence, complex processes can often be modeled with less effort in π. 2) The use of π-calculus is necessary to support multi subjects. Although multi subjects are not supported in the current work, using π-calculus simplify extension offers. A term of process algebra actions and operations is called a process. The operations and the syntax are given in table (1) and the formal semantics is given in table (2). There are also three different types of actions: 1) send, receive and hidden actions. The send and receive actions are used for exchanging messages between processes. The hidden action is always donated by a τ and serves as abstraction of concrete actions. In this section a brief outline of the π-calculus is given since it is the foundation for this formal verification approach. The syntax of the algebra is given in table (1).

Inactive Process. The inactive process **0** does not do anything. It is a termination of a process.

Prefixes. The different prefixes define the different actions. A send action $\overline{\mathrm{x}}\langle\tilde{y}\rangle$ sends the value y via the channel x. Angel brackets stand for free values, namely y. Receive actions like $\mathrm{x}(\tilde{y})$ receive a value via the channel x and save the result in the variable y. A send action of one process can be synchronized with a receive action of another one by using the same channel. The result is a τ action which is also called hidden transition. It is an abstraction of what the transition really does.

Sum, Parallel, Restriction. The sum is an exclusive choice between the processes P and Q. $P \mid Q$ denotes the parallel execution of P and Q and $(z)P$ restricts the scope of the name z to P.

Identifier. The identifier enables the definition of names for processes.

Table 2. The reduction semantics of the π-calculus

$$\text{R-INTER}\ \frac{}{\big(\overline{x}y.P_1 + M_1\big)|\big(x(z).P_2 + M_2\big) \rightarrow P_1|P_2\{{}^{z}\!/_{y}\}}$$

$$\text{R-TAU}\ \frac{}{\tau.P + M \rightarrow P} \qquad \text{R-PAR}\ \frac{P_1 \rightarrow P_1'}{P_1|P_2 \rightarrow P_1'|P_2}$$

$$\text{R-RES}\ \frac{P \rightarrow P'}{(z)P \rightarrow (z)P'} \qquad \text{R-STRUCT}\ \frac{P_1 \equiv P_2 \rightarrow P_2' \equiv P_1'}{P_1 \rightarrow P_1'}$$

$$\text{R-IDENT}\ \frac{P \rightarrow P'}{K \rightarrow P'} K = P$$

The semantics is defined by structural operational semantics rules (SOS) [23] of the general form $Rule = \frac{premises}{conclusion} condition$. The prefix "R-" denotes reduction. To give an example, the rule R-PAR can be written as:

$$\textbf{if}(P_1|P_2 \textbf{ AND } P_1 \rightarrow P_1') \textbf{ then } \quad P_1|P_2 \rightarrow P_1'|P_2$$

Let $P = \big(\overline{x}y.P_1 + M_1\big)$ and $Q = \big(x(z).P_2 + M_2\big)$. $\overline{x}y$ is a send prefix and $x(z)$ a receive prefix. Therefore P can send y via the channel x to Q where the variable z is assigned to y. In consequence all z terms of P_2 are renamed to y which is denoted by $P_2\{{}^{z}\!/_{y}\}$. The prefixes are consumed. The sub-processes M_1 and M_2 of P and Q are rendered void. In summary $P|Q$ evolves to $P_1|P_2\{{}^{z}\!/_{y}\}$.

A variant of the conclusion of R-INTER would be: R-INTER2-CONC $\big(\overline{x}y.P_2 + M_2\big)|\big(x(z).P_1 + M_1\big) \rightarrow P_2\{{}^{z}\!/_{y}\}|P_1$. We do not have to define a second rule for this variant because of the R-STRUCT rule: from $P_1 \equiv$ and $P_2 \rightarrow P_2'$ and $P_2' \equiv P_1'$ infer $P_1 \rightarrow P_1'$, where $\equiv$ is a structural-congruence relation. It is possible to infer R-INTER2-CONC from R-INTER-CONC using R-STRUCT because $P|P' \equiv P'|P$ for any processes P and P'. The remaining rules can be interpreted in the same manner. For details we refer to [26]. Although this semantics is simpler than the semantics used in [5], all language elements can be encoded.

2.3 Verification of ePASS-IoS Process Models

In this section we firstly give an outline of the algorithm and explain each step in more detail afterwards. The algorithm is roughly processing the following steps:

– Translate the graphical process models to a π-calculus representation. We explained this step for all language elements of ePASS-IoS in [5]. In fig. (2b) a simple example is illustrated.
– Infer a LTS by exploiting the SOS rules.
– Identify states which have no exit transitions. These states are deadlocks.
– Ignore the final legal deadlock, determine the shortest paths to all possible illegal deadlocks and select the shortest path of all of them.
– Determine all involved subjects and their actions involving on the path route.
– Give feedback to the UI and label the involved subjects and actions with a bold border in case the deadlock is illegal.

Inferring Labeled Transition Systems. The analysis of deadlocks is done on the transition system of the respective π-calculus terms. To obtain such a transition system, an ePASS process is firstly translated into an internal representation of an π-calculus expression. Afterwards, a transition system is inferred from this expression. Figure (2) shows the method exemplarily. In figure (2a) a simplified auction business process is shown. The bidder subject (S1) performs an internal action at first, then sends the payment and expects the delivery afterwards. The seller subject (S2) performs an internal action too, waits for a payment and then sends off the delivery. This ePASS process is translated into the π-calculus expression depicted in figure (2b). Thereby all "Do something" actions are translated into τ operators. The translation of Send actions leads to an overlined channel following the grammar: *<sender subject name>-<message type>-<receiver subject name>*. The channels for receive actions follow the same grammar except not using overlined channels identifiers. The π expression is translated to the transition system depicted in figure (2c) in accordance to the following definition.

Definition 2 (Labeled Transition System of a Process). *Let P be a π-calculus process. The transition system of P consists of:*

– *the set S of states which contains P itself and all processes which are reachable from P via transitions,* and
– *the transition relation $\longrightarrow$ between processes in S, which is specified by derivation rules given in table 2.*

The start state of the process P is referred to the root node of the **L**abeled **T**ransition **S**ystem. Every edge to successor states will go out of this node. Both of the π processes $S1$ and $S2$ start with a τ operation. Therefore, the R-Par rule of table 2 combined with the R-Tau rule leads to the first node of the "left" branch of the LTS. Either $S1$ will perform firstly the action followed by performing τ of $S2$ or the other way around will take place. The "right" branch of

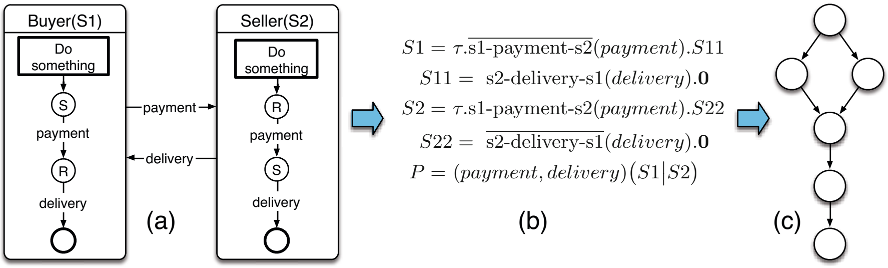


Fig. 2. The sequences of *S1* and *S2* (a) are translated to π expressions (b). By exploiting the SOS rules, defined in table (2), a LTS is inferred (c) in accordance to definition 2. The capacity of the subject input-pools are set to 0 in order to obtain a simple LTS.

the LTS can be inferred of by applying the R-Struct rule. Since the process P is restricted by the exchange of the messages *payment* and *delivery* , the message *payment* can be exchanged only between $S1$ and $S2$ in the next two steps. This system is used for the deadlock analysis. A deadlock is a state without outgoing transitions. The example transition system has only one deadlock state which is also the final state. It is called legal deadlock since we are modeling business process instances which have to be terminated at the end of their runtime. Every subject is equipped with a storage for incoming messages. This storage is called input pool and its data structure is a set of FIFO buffers. One FIFO buffer is needed for every receive exit condition which is a pair of (<Message-Type>, <Sender Subject>). The inputpool enables asynchronous communication and the multi set data structure provides a flexible access to the stored messages. It has a limited capacity which can be set in the editor. Without loss of generality, all FIFO buffers of a subject have the same capacity.

Different Cases of Deadlocks. When a deadlock was found, four different cases have to be distinguished.

All internal behaviors and all inputpool actors are terminated. In this case the process instance can always finish well.

Some internal behaviors are not terminated but all inputpool actors are terminated. The non terminated internal behaviors are in a receive state but no message will be send to them.

All internal behaviors are terminated and some inputpool actors not. In this case some messages left in the non terminated input pools. This can happen when either messages are send in a loop or when the internal behavior of the sender subject has one send transition more then the internal behavior of the receiver subject.

Some internal behaviors are terminated, others are still in the start state. When a process model includes a subject that only will be used optionally, this subject will stay in the start state.

The first case is a valid case. The second one is not valid, the cases 3 and 4 can be both and the verification system can not make this decision without input of the model designer. Therefore the system informs the user who can specify whether this is a valid case or not.

2.4 Implementation

A graphical editor was developed for the verification of the subject oriented process models. Fig. 3 and 4 show screenshots of the editor. The editor follows the same designing principles like the original Metasonic[19] editor but is using the ePASS-IoS elements and is extended by the check for interaction soundness. The implementation of the editor is based on the Graphical Editing Framework (GEF) [2] and on frameworks of the Metasonic AG. GEF is an open framework of the Eclipse environment and provides the opportunity to implement a graphical user interface for an existing data model. The Metasonic frameworks enabled the use of data binding and the pre-compiler AspectJ[1] RCP plug-ins were implemented which can be integrated into the Eclipse environment to use the ePASS editor.

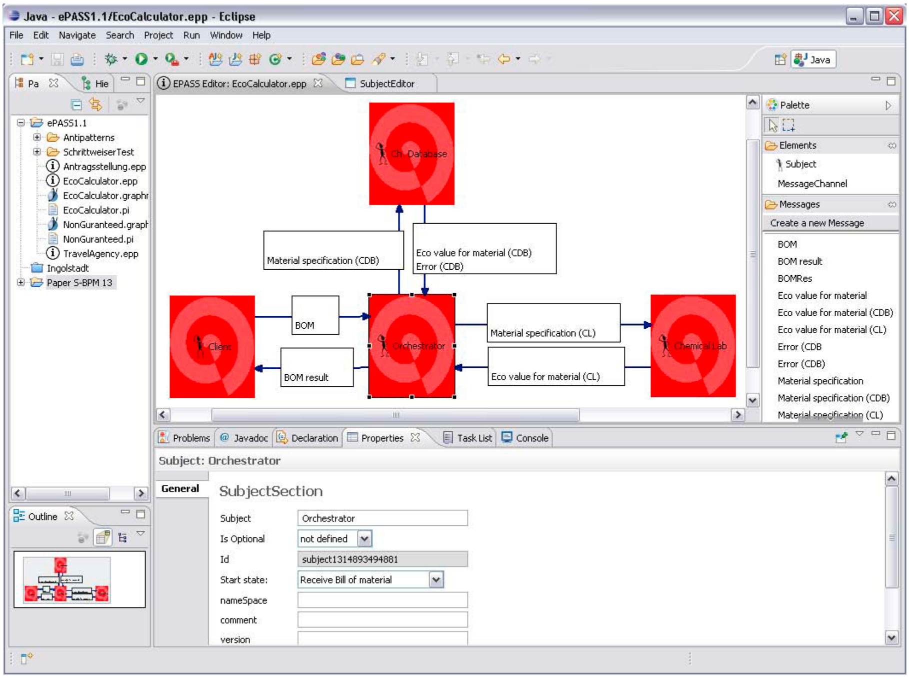


Fig. 3. Screenshots of the eclipse based editor, developed for the evaluation. The subject interaction view is shown with one of the test process models.

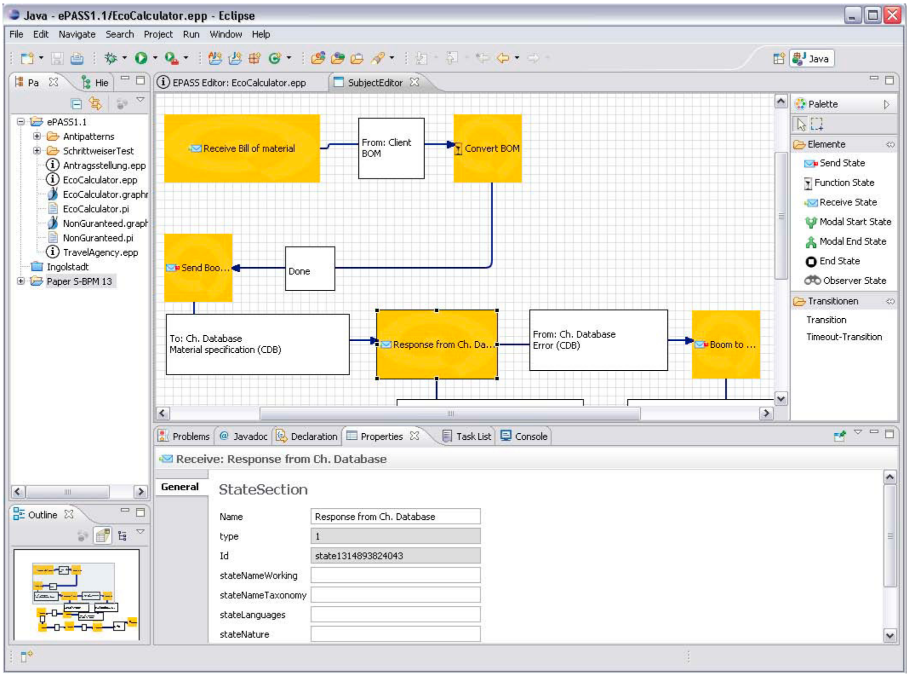


Fig. 4. Screenshots of the eclipse based editor, developed for the evaluation. An internal behavior of a subject is illustrated.

3 Evaluation

In [7] eight choreography patterns were identified. We have chosen these patterns for our evaluation because they are the most suitable ones for our purpose. The anti-pattern introduced in [4] tackle problems concerning compliance checking. They are not suitable for issues which can be arise by incompatible interaction behavior. Some of the patterns introduced in [3] are covered by anti-patterns described in [7]. We translated the choosen patterns including the examples in subject oriented models and verified them afterwards.

The input pool is the tool which enables asynchronous communication and has a limited capacity. Therefore a S-BPM process consisting of two subjects can run into an illegal deadlock in exactly three cases. Firstly, two participants are waiting for each other. Secondly, one participant has already terminated and the other one is still waiting for a message. Thirdly, the input pool limit of one participant is exceeded and the other one will send a message. The first two cases are particularly well represented by the contradicting sequence flow and the not-guaranteed termination anti-pattern. The last case is non of the 8 introduced anti-patterns but we evaluated that our approach detect this pattern as well.

3.1 AP1: Incomplete Sequence Flow

The first anti-pattern is called *Incomplete sequence flow* and is the most simple one. A path in the internal behavior of a subject is interrupted. This anti-pattern is already detectable by the check of structural soundness which is included in the Metasonic suite.

3.2 AP2: Contradicting Sequence Flow

A *contradicting sequence flow* means that the order of a send / receive action sequence is contradicted in two participants who are interacting with each other. An example for this anti-pattern is shown in figure 5. The bider will do the payment after the delivery arrived and the seller will deliver after the payment arrived. This leads to a classical deadlock situation. The final state of the model LTS is an illegal deadlock as shown in the right figure of (5). In the example each subject provider can only perform the internal action *Do something* and is locked in the receive state afterwards. Hence the LTS has exactly 4 states. Because of the non deterministic behavior either the Bidder or the Seller can perform the first action. When the Bidder performs the first action the path (0 -> 1 -> 3) is generated. Otherwise the path (0 -> 2 -> 3) of the LTS in figure (5) is generated.

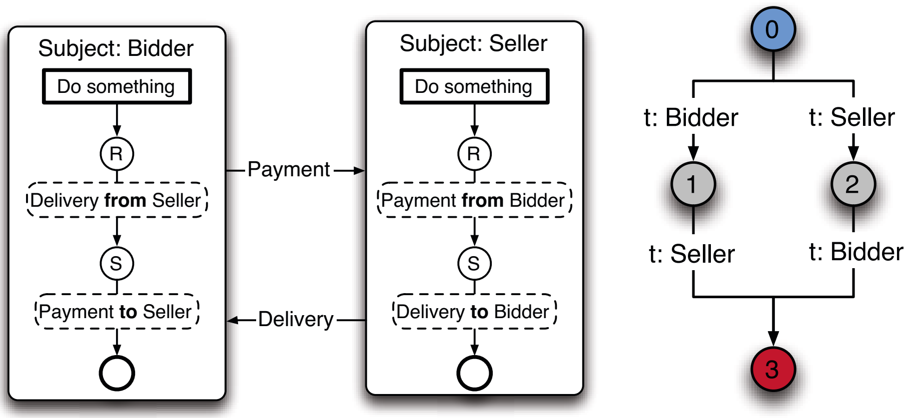


Fig. 5. Contradicting sequence flow. Each participant is waiting for each other what leads to an illegal deadlock (state 3). t:Bidder denotes that the Bidder performs an internal action τ.

3.3 AP3: Not-guaranteed Termination

An example for an *not-guaranteed termination* is depicted in figure 6. A bidder will process always an *Feedback* request / replay sequence but the *Auctioning Service* demands only sometimes one. When the *Auctioning Service* demands no feedback the *Bidder* is running into a deadlock. The concerning LTS includes besides the legal deadlock (state 8) the illegal deadlock (state 4)

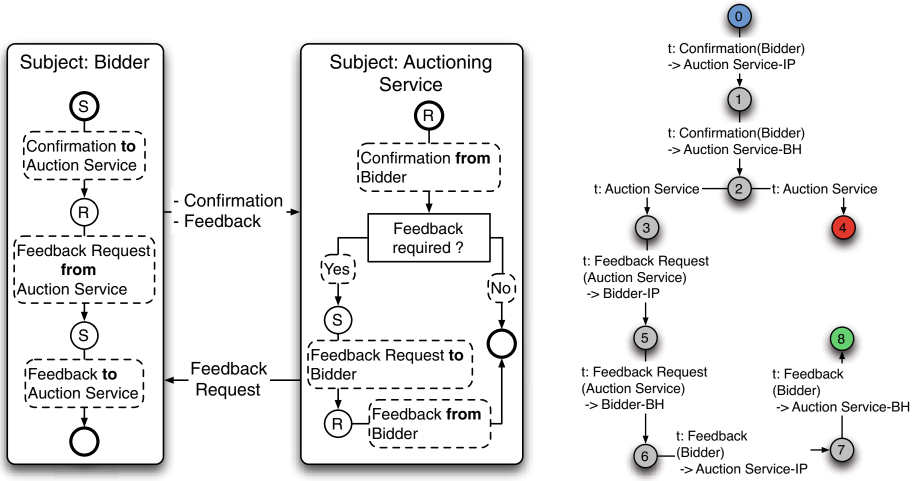


Fig. 6. A not guaranteed termination process snippet and its concerning LTS. The Bidder always expect a Feedback request but the Auctioning Service not.

3.4 AP4: Incompatible Branching Behavior

The next anti-pattern is called *incompatible branching behavior*. Internal behaviors of different subjects can contain same states with same transitions. If the decision made at these states are different, the control flow will take different branches and therefore the interaction behavior can become incompatible. This issue is depicted in figure 7 The two subject exchange the messages *Bank transfer* and *Credit card*. Both internal behaviors contain the action state *Payment* which has two successor transitions each. The exit conditions of the transitions are *Bank transfer* and *Credit card*. When the two subjects have different information about the payment options, the different exit conditions lead to different successor states. Since the successor states are *Send* and *Receive* actions, which will exchange the concerning messages, the interaction behavior is not compatible to each other.

When we apply the verification on these patterns, the verification will state that the model is not valid and that the labeled transition system contains two faulty end-states. They are colored red in figure 7 which depicts the inferred LTS of the process model. The LTS was created with the yed editor [29] which offers functions for automatic layouting. The green end state in figure 7 is the legal deadlock. In this state all system actors are terminated, namely the actors for the internal behavior and for the inputpool of each subject. The conditions have two different transitions and therefore four different exit condition combinations are possible. When the bidder and the seller make the same choice of of payment options, the successor paths will lead to the same end state, the legal deadlock state with number (15) in the figure. When the bidder choose *Credit card* and the seller choose *Bank transfer* the paths will lead to one red deadlock state.

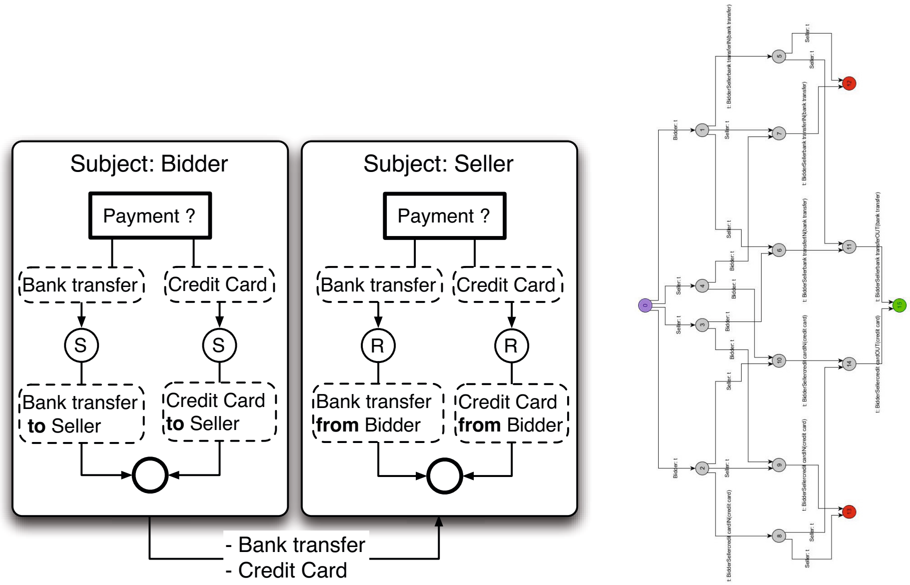


Fig. 7. Incompatible branching behavior of the subjects *Bidder* and *Seller*. The execution of the process model starts at the purple state in the LTS and can lead to one legal deadlock (green) and two illegal deadlocks (red).

The other way around the system will end in the other deadlock state. Another variant of this anti-pattern contains loops but the result is the same.

3.5 AP5: Impossible Data-Based Decisions

Sometimes process model designers put decisions in the model which are supposed to work on data not be available. These *impossible data-based decisions* are not detectable by the structural soundness check and only in certain cases by the formal verification for interaction soundness. The formal verification works on an abstract level and takes every internal action as a τ action. Therefore impossible data-based decisions can only be detected if they have an impact on later interaction actions. In these cases the anti-pattern is treated in the same way like the incompatible branching behavior pattern. The other cases effects only the participant where the error occurs and can not block the hole process. In figure (8) an example is outlined where it is assumed that the *Bidder* works on not available data.

3.6 AP6 Optional Participation

Not all of the including subjects have to participate in every process instance. 4 eye principles or fallback solutions are examples for this and the concerning pattern is called *optional participation*. When a subject is not being used, this

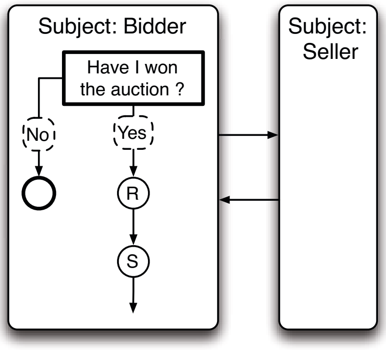


Fig. 8. Outline of the impossible data-based decisions anti-pattern. The *Bidder* can not know whether the auction is won. These decisions lead most often to incorrect interaction behavior.

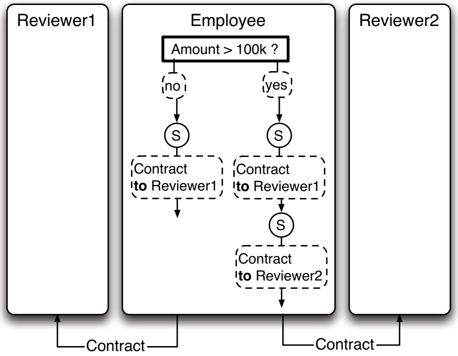


Fig. 9. Outline of the optional participation anti-pattern. When the contract is not send to *Reviewer2*, the concerning internal behavior stays at the start state. This leads to a deadlock.

subject will stay in its start state or in one of its first receive states. This is a deadlock situation: The verification system detects the deadlock and throws an error message on the UI first. The option to ignore this deadlock is included in the message dialog. The process designer can inspect the issue and decide whether it is a wanted or not wanted situation. If the process designer labeled it as wanted the verification system is not throwing the error message again.

The 4 eye principle process in figure 9 is a good example for optional participation. An insurance has to cover a damage and when the amount of loss exceeds a certain threshold, a second reviewer is needed. In other cases the subject *Reviewer2* is still waiting for an application and the overall process can not terminate.

3.7 AP7: Uni-lateral Sequentialization

Uni-lateral sequentialization is another anti-pattern. Because of the modal split / join operator pair, sequences can be processed in parallel. The subject *Seller* in figure 10 is processing a part of the interaction actions in parallel. The behavior is more permissive than it would be in the case that all interaction actions are ordered in only one sequence. This leads not to an error in ePASS modeled processes, since the subjects store received messages in their input pools. Although, the processing of certain tasks can be delayed. If incompatible interaction behavior occurs as aftereffect, our method will detect this.

3.8 AP8: Mixed Choices

One of the most complex anti-pattern is the *mixed choices* pattern. It consist of a branching of a send and a receive action. The send action can as long take

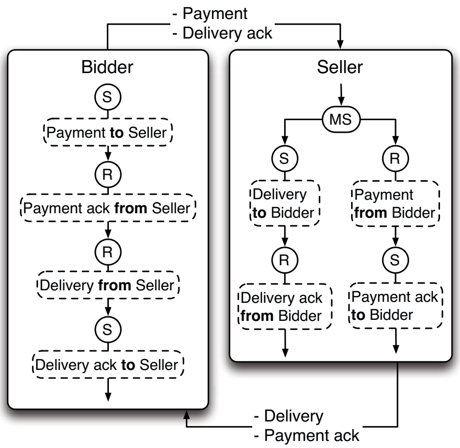


Fig. 10. The *uni-lateral sequentialization* anti-pattern does not lead directly to errors. It just states that one internal behavior is more permissive than the other one. If incompatible interaction behavior occurs as aftereffect, our method will detect this.

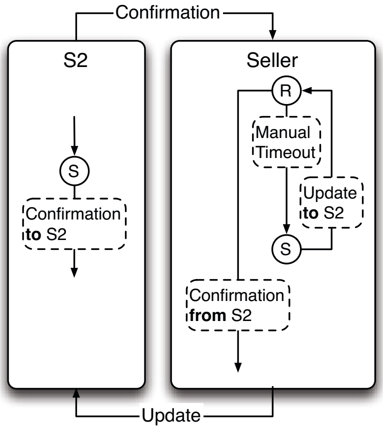


Fig. 11. The *mixed choices* anti-pattern is hard to model correct. When it leads to a left message in the input pool, the verification method would find it.

place as the receive action did not receive a message. The figure (11) gives an example for that. The seller can update its setting until the confirmation message is received. The solution for this kind of problem is not obvious and according to [7] even experienced modeler make errors describing such situations. While the seller is in the branch of sending update information, the *Confirmation* message could be received. Due to this overlapping, the *Update* message could be left in the input pool of subject *S2* which would be detected by our verification method as explained in section (2.4).

3.9 Performance of the Verification Method

Next to the detection of the anti-patterns the solution is supposed to be that fast, that it does not slow down the work of the model designer to much. To check whether our performance requirements are met, we applied the algorithm to ten different process models. Three of them were delivered with the S-BPM Suite from Metasonic [19] and serve as example process models for customers. One is a process modeled as use case in the Theseus / TEXO project [24]. Since these process models could be verified in less than a second, we modeled 6 further fictional process models. The fictional process models were developed such a way that LTSs with a large amount of states could be inferred easily . The tests were run on a notebook with an Intel Core 2 Duo microprocessor and with 4 GB RAM. The operating system was Windows 7. The results are listed in table 3. Nine of ten models could be verified in less then 15 seconds and eight of ten in

Table 3. Verification time of different process models. F1 to F6 are fictional process models. They were developed by using a high number of modal split and modal join actions.The action sequences between these actions can be executed in parallel and therefor a large number ofLTS states could be inferred. Only the verification time of F6 exceed the required limit of 15 seconds.

Process	Subjects	LTS states	Verification time [s]
Eco Calculator	4	52	< 1
Application Process	3	22	< 1
Banking process	5	105	< 1
Ordering Process	4	28	< 1
F1	2	20	< 1
F2	3	108	< 1
F3	4	271	< 1
F4	5	559	< 1
F5	6	4072	8
F6	7	12223	41

less than one second. All of the practice related process models ($F1$-$F4$) could be verified in less than a second. Although the verification of $F6$ took 41 seconds, which is almost 3 times higher than the required limit, in summary the results can be scored as good. Nevertheless the algorithms should be improved since we could not verify real world processes.

3.10 User Support for Error Correction

We required not to use external verification tools but integrate verification algorithms into the tool. The reason for that is to get the ability providing better support for the user to correct errors. We developed two mechanisms to achieve this goal. The first mechanism is to visualize possible error sources graphically. Therefore, the shortest path from the start state to the deadlock state is determined. This path delivers information about the involved subjects and the involved actions of the subjects. Hence it is possible to label these elements. We have labeled the involved subjects and the involved internal actions with a big blue frame. The user can easily see which elements are involved and therefore can better trace the error sources. In some cases many subjects can be involved on the path from the start state to a deadlock state. The ability to detect errors suffer from this fact and so a second mechanism has been introduced to find error sources. It is called stepwise check and gives the ability to check only parts of the process. The user can select subjects which are supposed to verify. Instead of the remaining subjects an open environment is used. This environment is an ideal interaction partner. It receives all messages the subjects will send and delivers necessary messages when the subjects requires them. If the verification method detect errors, the error sources can be localized within the subjects verified. This way the process designer is able to find errors systematically.

4 Related Work

In [25] a solution for checking interaction soundness for service orchestrations is introduced. The authors introduce an graphical example but have to write the concerning π-calculus terms manually, since the BPMN related, graphical notation does not allow an automatic transformation. To evaluate the method the Mobility Workbench (MWB) is used[27,28]. The tool takes a π-calculus expression as input and outputs a binary result in this case. Either the check is positive or negative. Therefore they can not use this result to analyze the error sources. Further more, the tool is not optimized for this purpose in that sense, that certain loop constructs are not detectable. In [13] a BPMN extension is introduced together with a graphical editor. This editor enables the automatic transformation to π-calculus terms. The weaknesses of the verification method itself are still unsolved.

In [16,17] Lohmann et al. transform BPMN and BPEL models to *Open Workflow Nets*. The *Open Workflow Nets* are verified with a model checker tool and exploiting a graph edit distance and deadlocks are supposed to solve automatically. The algorithm is not able to discover in all cases which participant behavior causes the error. No graphical support is provided for the model designer to solve the cause of the error in that cases it can not be done automatically.

In the work of Deng et al.[8] services are described by the Web Service Definition Language (WSDL) and were transformed to π-calculus expressions. They also propose to use the MWB to verify the expressions though without giving details how the results could be used to detect the reasons for errors.

A further approach is introduced in [15]. Business process are described with BPEL and the Business Property Specification Language (BPSL). The BPEL process were transformed to π-calculus expressions and were inferred into finite state machines. Linear temporal logic (LTL) forms were generated from the BPSL expression. Both kinds of expressions were given to a compliance checking framework called OPAL. Again, the outputs are either the check failed or the check succeeded.

In [4,18] approaches for compliance checking are featured. These approaches give visual feedback to the model designer but can not be used for compositions of participants. The capacity and the structure of the technique for asynchronous communication must be taken into account. Both approaches fail concerning this point.

5 Summary and Conclusion

Finding deadlocks in distributed systems and resolving them are two very challenging tasks. These challenges are well known and many different formalisms, algorithms and tools have been developed over the years to tackle the associated problems. Nevertheless it is not clear which solution is suitable for finding deadlocks in S-BPM processes and resolving them. In this paper it has been shown how ePASS-IoS 1.1 process models can be verified for interaction soundness by using the π-calculus as formal foundation. An graphical editor was developed and the verification algorithms were integrated. Further more, two mechanisms to support the user finding the reasons for errors are integrated. We modeled the eight choreography anti-patterns in ePASS-IoS and investigated which of them can be detected. The *incomplete sequence flow* anti-pattern can already be detected by the structural soundness check which is integrated in the Metasonic Suite. Five of them can always be detected namely *contradicting sequence flow*, *not-guaranteed termination*, *incompatible branching behavior*, *impossible data-based decisions* and *optional participation*. The remaining two *Uni-lateral sequentialization* and *mixed choices* can be detected when they lead to incompatible interaction behavior. Further more, we evaluated that the performance is fast enough to use the method in practice, although the algorithms are simple.

For future work, we currently plan the following:

– Extend ePASS IoS and formal semantics to support multi subjects.
– Optimize the algorithms to obtain even faster results to be able to verify also very complex processes.
– Combine the verification method with Abstract State Machines.

Acknowledgments. We thank Kai Mehringskötter for his great work developing the graphical editor and implementing the verification method. We thank Albert Fleischmann for fruitful discussions and valuable comments. The work presented in this paper was performed in the context of the Software-Cluster project SINNODIUM (www.software-cluster.org). It was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. "01IC10S01" | "01|C10S05". The authors assume responsibility for the content.

References

1. AspectJ, `http://www.eclipse.org/aspectj/` (last accessed on January 15, 2014)
2. Graphical Editing Framework, `http://www.eclipse.org/gef` (last accessed on January 15, 2014)
3. van der Aalst, W.M.P., Mooij, A.J., Stahl, C., Wolf, K.: Service interaction: Patterns, formalization, and analysis. In: Bernardo, M., Padovani, L., Zavattaro, G. (eds.) SFM 2009. LNCS, vol. 5569, pp. 42–88. Springer, Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-01918-0_2`
4. Awad, A., Weske, M.: Visualization of Compliance Violation in Business Process Models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 182–193. Springer, Heidelberg (2010)
5. Borgert, S., Steinmetz, J., Mühlhäuser, M.: ePASS-IoS 1.1: Enabling Inter-enterprise Business Process Modeling by S-BPM and the Internet of Services Concept. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 190–211. Springer, Heidelberg (2011)
6. Decker, G., Kopp, O., Barros, A.: An Introduction to Service Choreographies. Information Technology 50(2), 122–127 (2008)
7. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Information Systems 36(2), 292–312 (2011), `http://linkinghub.elsevier.com/retrieve/pii/S0306437910000591`

8. Deng, S., Wu, Z., Zhou, M., Li, Y., Wu, J.: Modeling Service Compatibility with Pi-calculus for Choreography. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 26–39. Springer, Heidelberg (2006)
9. Fleischmann, A.: PASS - A Technique for Specifying Communication Protocols. In: Proceedings of the IFIP WG6.1 Seventh International Conference on Protocol Specification, Testing and Verification VII, pp. 61–76. North-Holland Publishing Co, Amsterdam, http://portal.acm.org/citation.cfm?id=645831.670083
10. Fleischmann, A.: Distributed Systems: Software Design and Implementation. Springer, Berlin (1994)
11. Fleischmann, A.: What Is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-15915-2_7
12. Fleischmann, A., Lippe, S., Meyer, N., Stary, C.: Coherent Task Modeling and Execution Based on Subject-Oriented Representations. In: England, D., Palanque, P., Vanderdonckt, J., Wild, P.J. (eds.) TAMODIA 2009. LNCS, vol. 5963, pp. 78–91. Springer, Heidelberg (2010)
13. Freund, J., Rcker, B., Henninger, T.: Praxishandbuch BPMN. Hanser (2010), http://books.google.com/books?id=bw9YPgAACAAJ
14. Keller, R.M.: Formal Verification of Parallel Programs. Communications of the ACM 19(7), 384 (1976), http://portal.acm.org/citation.cfm?id=360248.360251
15. Liu, Y., Müller, S., Xu, K.: A static Compliance-Checking Framework for Business Process Models. IBM Systems Journal 46(2), 335–361 (2007), http://dx.doi.org/10.1147/sj.462.0335
16. Lohmann, N.: Correcting Deadlocking Service Choreographies Using a Simulation-Based Graph Edit Distance. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 132–147. Springer, Heidelberg (2008)
17. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing interacting BPEL processes. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 17–32. Springer, Heidelberg (2006)
18. Ly, L.T., Knuplesch, D., Rinderle-Ma, S., Göser, K., Pfeifer, H., Reichert, M., Dadam, P.: Seaflows toolset compliance verification made easy for process-aware information systems. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 76–91. Springer, Heidelberg (2011)
19. Metasonic: Metasonic Suite (2014), http://www.metasonic.de/ (last accessed on January 15, 2014)
20. Milner, R.: Communicating and mobile systems: The π-calculus. Cambridge University Press (1999)
21. Milner, R.: A Calculus of Communication Systems. LNCS, vol. 92. Springer, Heidelberg (1980)
22. OMG: Business Process Modeling Notation. 2.0 edn. (2012), http://www.omg.org/spec/BPMN/2.0/ (last accessed on January 15, 2014)
23. Plotkin, G.D.: A structural approach to operational semantics. Tech. Rep. DAIMI FN-19, University of Aarhus (1981)
24. Project of the German Federal Ministry of Economy and Technology: TEXO Infrastructure for Web-based services (2012), http://theseus.pt-dlr.de/en/914.php (last accessed on January 15, 2014)
25. Puhlmann, F., Weske, M.: Interaction Soundness for Service Orchestrations. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 302–313. Springer, Heidelberg (2006)

26. Sangiorgi, D., Walker, D.: The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press (2003)
27. Uppsala Universites, Department of Information Technology: The Mobility Workbench (2006), http://www.it.uu.se/research/group/mobility/mwb (last accessed on January 15, 2014)
28. Victor, B., Moller, F.: The Mobility Workbench - a tool for the π-calculus. In: Dill, D.L. (ed.) CAV 1994. LNCS, vol. 818, pp. 428–440. Springer, Heidelberg (1994)
29. yWorks: yED Graph Editor (2012), http://www.yworks.com/ (last accessed on January 15, 2014)

S-BPM-Ont: An Ontology for Describing and Interchanging S-BPM Processes

Kai Michael Höver and Max Mühlhäuser

Technische Universität Darmstadt
Dept. of Computer Science
64289 Darmstadt, Germany
{kai,max}@tk.informatik.tu-darmstadt.de

Abstract. The description of business processes is important for an unambiguous definition and its reusability. However, due to different data models of business modeling tools, an exchange of process definitions is difficult. This demands for a formal model of business processes. An ontology is an explicit specification of a conceptualization and can therefore be used as an (inter)lingua to describe S-BPM processes. This paper presents a S-BPM ontology modeled with the Web Ontology Language.

Key words: S-BPM, Ontology, OWL, Modeling.

1 Introduction and Motivation

S-BPM is an emerging method for modeling business processes [1,2]. Its growing popularity and distribution comes along with a rise of tools for modeling S-BPM processes. Such tools, however, use different data models for representing S-BPM processes. This results in a low interoperability between S-BPM applications, be it modeling applications or applications for executing S-BPM processes. All of them need a representation of the processes, and these are as said stored in a various kind of data models and formats. To take an example, the two S-BPM modeling applications Metasonic Suite[1] and S-BPM Groupware [3] describe their process models differently. The first stores models in XML representations, the latter uses Scala data models [2], JSON (JavaScript Object Notation), and database models. Thus, an exchange of business models is not instantly possible.

One approach to facilitate the interoperability between applications is to write translators to convert from one S-BPM process data model to another. This solution demands for $O(n^2)$ translators for n different data formats. A meta-format helps to reduce the number of translators. One approach is to define S-BPM processes with the Extensible Markup Language[3] (XML), particularly with a definition of an XML schema[4] for S-BPM. However, this approach only

[1] http://www.metasonic.de/en/metasonic-suite
[2] http://www.scala-lang.org
[3] http://www.w3.org/TR/xml/
[4] http://www.w3.org/XML/Schema

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 41–52, 2014.

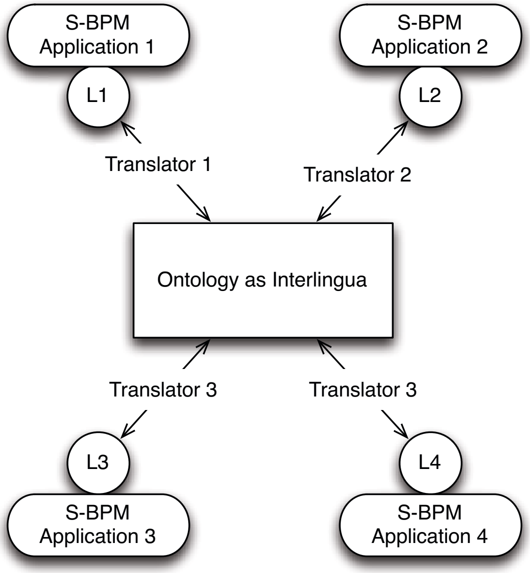


Fig. 1. Translation between different languages used by different applications

defines the syntax but lacks of semantics. In contrast, ontologies define a domain theory as a conceptual level on top of XML data.

Ontologies enable the formal definition of an inter-lingua meta-format. Using an ontology reduces the number of translators to only $O(n)$. "To assist interoperability, ontologies can be used to support translation between different languages and representations" [4]. Unlike data models, an ontology is independent of specific data model, it is "a representation of a world external to any particular system" [5], so that it "can be reused by different kinds of applications/ tasks" [6]. Figure 1 depicts the general approach and illustrates the use of an ontology as an interlingua between different languages (data representations) and corresponding translators to translate not directly between languages, but between a language and an ontology (interlingua).

The paper is structured as follows. First, we provide a short introduction to ontologies and the Web Ontology Language (OWL). Then we present an ontology for the representation of S-BPM processes. Here, we first explain the applied methodology and then present the ontology's concepts in more detail. We close with a summary and an outlook on future work.

2 Introduction to Ontologies

Several definitions for ontologies exist [7]. One of the most cited definitions is provided by Gruber in 1993. He defines an ontology as an "explicit specification of a conceptualization" [8]. In 1997, Borst extended this definition by the point that the conceptualization should express a shared view [9]. Studer et al. merge these definitions and define an ontology as a "formal, explicit specification of a shared conceptualization" [10]. The important elements of this definition are:

– *conceptualization*: an ontology describes the objects, concepts in a certain domain of interest and the relationships between them
– *formal* and *explicit*: the ontology should be specified in a language that comes with formal semantics
– *shared*: because if not, the benefits of having an ontology are limited.

One of the most popular ontology description languages is the Web Ontology Language (OWL), a W3C recommended standard since 2004[5]. OWL is based on description logics and enables the definition of classes, properties, and relations among conceptual objects [11].

After this short introduction we present a S-BPM ontology modeled with OWL.

3 A S-BPM Ontology

In this section, we present an ontology for describing S-BPM processes called S-BPM Ont. First, we present the applied methodology for engineering the ontology. Then we present the design of the ontology.

3.1 Methodology

We use the Ontology Development 101 methodology for the engineering of the S-BPM ontology [12]. It is an iterative process for creating ontologies. The methodology suggests to follow 7 steps:

1. Determine the domain and scope of the ontology
2. Consider reusing existing ontologies
3. Enumeration important terms
4. Define classes and the class hierarchy
5. Define the properties of the classes
6. Define constraints
7. Create instances

Subsequently, we orientate ourself by these steps for defining the S-BPM ontology.

3.2 Scope of the Ontology

The scope of the S-BPM ontology is the domain of subject-oriented business processes. The purpose is to be able to describe S-BPM processes in an application-independent, formal way.

In order to refine the scope we present some so called competency questions in the following as suggested in the methodology of Grüninger and Fox [13]. Competency questions have the purpose to ask questions the ontology should be able to answer.

[5] `http://www.w3.org/TR/owl2-overview/`

3.3 Reusing Existing Ontologies

To our best knowledge, there does not exist any ontology for describing S-BPM processes that could be reused. On et al. [14] define a behavior ontology for process algebra. As the ontology is not defined in any ontology language it cannot be directly reused, however partly its concepts. Related, but with another scope is the publication of Fleischmann et al. [15]. Here, the authors define a meta-model to support multi-agent business processes.

3.4 Important Terms

Albert Fleischmann defines the elements of the S-BPM modeling language in [1]. According to this publication we can identify several important terms. The basic modeling features include for the communication structure *subjects*, *messages*, and *business objects*. Further, subjects can perform actions which include *sending* and *receiving* messages as well as executing *internal actions*. The *behavior* is defined by these activities that are performed in a certain order. Depending on the outcome of activities, the execution of the process chooses one of the available *exit conditions* or *alternatives*. Subjects exchange messages via their *input pools*. Each input pool can define *size constraints* for the whole input pool, specific message types, or senders of messages.

3.5 Define Classes and the Class Hierarchy

In this section we define the classes of the ontology after identifying important terms. For this purpose and the definition of both properties and constraints we follow the definitions of S-BPM defined by Fleischmann [1,2] and Börger [16].

In the following, we define classes and the class hierarchy of the S-BPM ontology. It is depicted in figure 2.

Subject A subject is an acting element within a process. Subjects execute and synchronize their activities by exchanging messages. Each subject has an assigned behavior and an input pool.

Behavior The process behavior that is assigned to a subject and defined by its states and state transitions.

State A state is part of a behavior. A subject can be in only one state at a time. There are end states and initial states. A behavior diagram can have only one initial state but one or more end states. A state has further a specific type of action (send, receive, function).

- **InitialState** The initial state of a subject behavior
- **EndState** An end state of a behavior. A subject behavior may have one or more end states

Edge An edge defines the transition between two states. A transition is true if it satisfies a certain exit condition.

- **Timeout** Timeouts are a subset of edges as well as
- **UserAbruption** that defines the interruption of a process by a user.

ExitCondition The class of exit conditions

Action In a state a certain type of action is defined. We distinguish between internal function and communication acts (send or receive). Formally, an *Action* is defined as $Action \equiv CommunicationAct \sqcup Function$.

- **CommunicationAct** A *Communication Act* is either an act of sending or receiving messages. It is defined as $CommunicationAct \equiv Send \sqcup Receive$. Further, the classes of *Send* and *Receive* are disjoint: $Send \sqcap Receive \equiv \emptyset$
  - **Send** A send communication act
  - **Receive** A receive communication act
- **Function** An internal function that is performed by a subject. The class of internal functions is disjoint with the class of communication acts and thus its complement: $CommunicationAct \sqcap Function \equiv \emptyset$

InputPool An input pool for exchanging messages between subjects

Restriction An input pool may be configured by size restrictions, i.e.,

- **TypeRestriction** A size restriction regarding a certain type of messages
- **SenderRestriction** A size restriction regarding a certain sender subject

Message A message is exchanged between subjects via their input pools

MessageType Each message is of some type like a business trip request.

3.6 Properties of the Classes and Their Constraints

After outlining the classes of the S-BPM ontology, we define the relations between them, technically also called *properties*. In OWL, properties are also called *roles*. Relations are always binary. OWL distinguishes between datatype properties and object properties. *Datatype properties* define relations between classes and concrete data types like string, integer, date, etc. *Object properties* define relations between classes.

Relations between Messages, Message Types, and Subjects

A message can be of a certain type. We therefore define a relation *hasType* between a *message* and a *message type* with

$$\exists hasType.\top \sqsubseteq Message \tag{1}$$

$$\top \sqsubseteq \forall hasType.MessageType \tag{2}$$

defining that the domain of *hasType* is a message and the range is the type of the message. Further, a message is of exactly one message type.

A message has exactly one sender and one receiver (this is also true for multi-subjects at the process structure level). This constraint is expressed by the relations *sender* and *receiver*, and the specification of the *Message* class (see equations (5)-(7)). The definition of *sender* is

$$\exists sender.\top \sqsubseteq Message \tag{3}$$

$$\top \sqsubseteq \forall sender.Subject \tag{4}$$

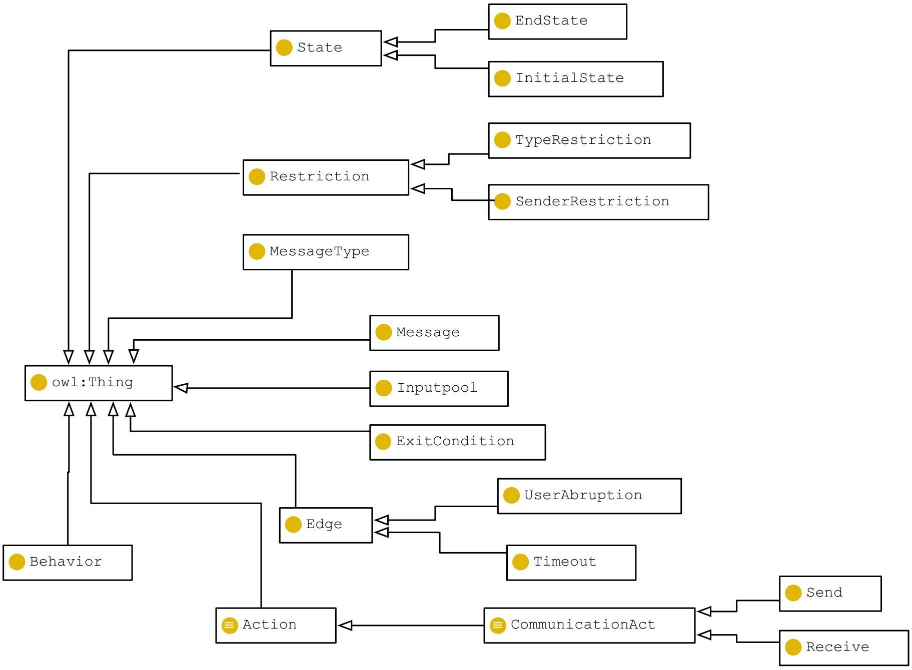


Fig. 2. The class hierarchy of the S-BPM ontology

The definition of *receiver* is analogous.

This leads us to the definition of the class *Message* with:

$$Message \sqsubseteq (\leq 1\ hasType) \sqcap (\geq 1\ hasType) \tag{5}$$
$$Message \sqsubseteq (\leq 1\ sender) \sqcap (\geq 1\ sender) \tag{6}$$
$$Message \sqsubseteq (\leq 1\ receiver) \sqcap (\geq 1\ receiver) \tag{7}$$

Relations between States, Edges, Behavior, and Subject

States and their transitions define the behavior of a subject. As introduced in section 3.5, subclasses of states are initial states and end states. Transitions are directed edges that have both a source and a target. For this we define two relations *source* and *target* for edges.

$$\exists source.\top \sqsubseteq Edge \tag{8}$$
$$\top \sqsubseteq \forall source.State \tag{9}$$
$$\exists target.\top \sqsubseteq Edge \tag{10}$$
$$\top \sqsubseteq \forall target.State \sqcap \neg InitialState \tag{11}$$

Both source and target relations are unambiguous, that means that they refer to exactly one state each. This constraint can be expressed by defining these relations as functional. This means that $source(s_1, e_1)$ and $source(s_1, e_2)$ implies $e_1 = e_2$. The same is true for the *target* relation.

States have one ore more *outgoing edges*. A special type of an edge is a time out edge. Each state may have at most one timeout edge but many user abruptions. An end state may have outgoing edges but an initial state has only outgoing edges. Doing so, the class of states is specified as

$$State \sqsubseteq \geq 1\ outgoingEdge.Edge \sqcap \leq 1\ outgoingEdge.Timeout \tag{12}$$

$$State \sqsubseteq (\leq 1\ hasServiceType) \sqcap (\geq 1\ hasServiceType) \tag{13}$$

Further, a state performs a certain type of action (send, receive, function), except of an end state. The relation hasServiceType between a state and an action class is defined as:

$$\exists hasServiceType.\top \sqsubseteq (State \sqcap \neg EndState) \tag{14}$$

$$\top \sqsubseteq \forall hasServiceType.Action \tag{15}$$

The behavior of a subject is described as a set of states and their direct edges between them:

$$Behavior \sqsubseteq (\geq 0\ Edge) \tag{16}$$

$$Behavior \sqsubseteq (\leq 1\ hasState.EndState) \sqcup (\geq 2\ hasState.State) \tag{17}$$

A subject is assigned to at most one behavior and exactly one input pool. The relation *hasInputpool* is inverse functional, because the owner of an input pool is unambiguous. The definition of these two relations are

$$\exists hasBehavior.\top \sqsubseteq Subject \tag{18}$$

$$\top \sqsubseteq \forall hasBehavior.Behavior \tag{19}$$

$$\exists hasInputpool.\top \sqsubseteq Subject \tag{20}$$

$$\top \sqsubseteq \forall hasInputpool.InputPool \tag{21}$$

$$\tag{22}$$

Furthermore, the class of subjects is formally defined as

$$Subject \sqsubseteq (\leq 1\ hasInputpool) \sqcap (\geq 1\ hasInputpool) \tag{23}$$

$$Subject \sqsubseteq (\leq 1\ hasBehavior) \sqcap (\geq 1\ hasBehavior) \tag{24}$$

Relations between Messages, Input Pools and Their Constraints
Input pools can be configured by several size constraints:

- the overall capacity of an input pool
- size restriction regarding a special type of message
- size restriction regarding a special sender

We specify the overall capacity of an input pool with a datatype property *hasCapacity* which enables to set the maximum number of messages of an input pool with a positive number. For *message type restrictions* and *sender restrictions* we define a relation *hasRestriction* with its two sub-properties *hasTypeRestriction* and *hasSenderRestriction*. The class for message *type restrictions* is defined as

$$TypeRestriction \sqcap (\leq 1\ hasCapacity.posInt) \sqcap (\geq 1\ hasCapacity.posInt) \tag{25}$$

$$TypeRestriction \sqcap (\leq 1\ regarding.MessageType) \sqcap (\geq 1\ regarding.MessageType) \tag{26}$$

$$\tag{27}$$

Analogously, the class of *sender restrictions* is specified as

$$SenderRestriction \sqcap (\leq 1\ hasCapacity.posInt) \sqcap (\geq 1\ hasCapacity.posInt) \tag{28}$$

$$SenderRestriction \sqcap (\leq 1\ regarding.Subject) \sqcap (\geq 1\ regarding.Subject) \tag{29}$$

$$\tag{30}$$

An input pool then has an arbitrary number of restrictions regarding message types or senders (or both), but exactly one overall capacity restriction. This is necessary to be able to distinguish between synchronous and asynchronous communication.

$$InputPool \sqcap (\leq 1\ hasCapacity.posInt) \sqcap (\geq 1\ hasCapacity.posInt) \tag{31}$$

$$InputPool \sqcap (\geq 0\ hasSenderRestriction.SenderRestriction) \tag{32}$$

$$InputPool \sqcap (\geq 0\ hasTypeRestriction.TypeRestriction) \tag{33}$$

$$\tag{34}$$

Figure 3 illustrates the constraints model of input pools.

Create Instances. In our case, instances are not part of the S-BPM ontology until a concrete process is defined. Therefore, the proposed ontology only contains concept definitions also called TBox (terminology knowledge) and no assertional knowledge (ABox). However, in the following section we present an example business process and its instances that use the defined ontology.

3.7 Ontology Example

In order to illustrate the use of the ontology, we present the employee subject behavior known from the business trip application example (see [2, p. 20]). Figure 4 depicts the employee's behavior.

In figure 5 an ontological representation of the employee behavior diagram is presented. It depicts the employee with its input pool and its behavior. The behavior consists of the states (fill out request, send request, receive answer, do BT, and End) as well as the edges between them. Each edge defines a source and a target edge.

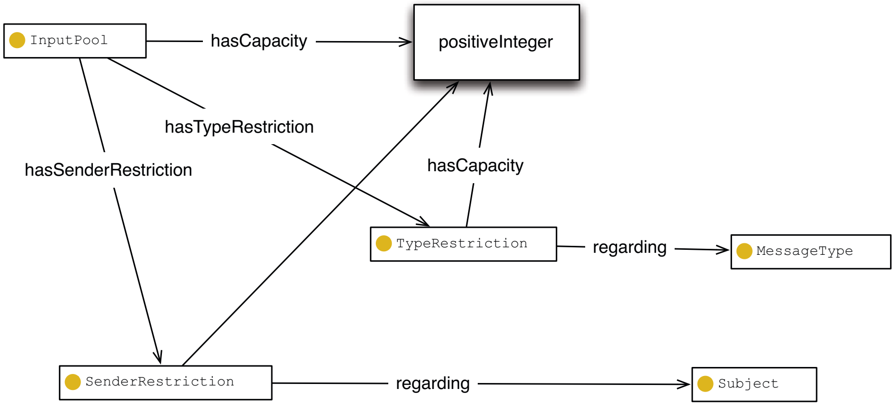


Fig. 3. The model of input pool size constraints

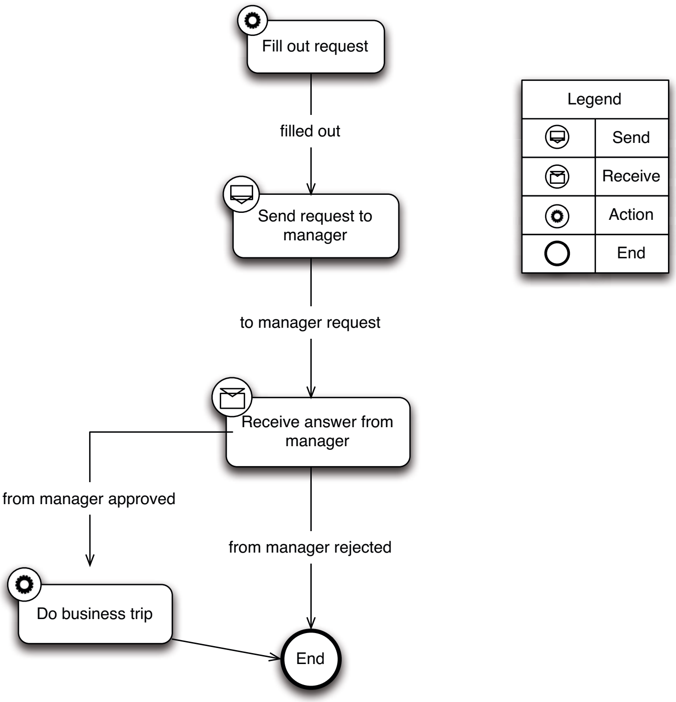


Fig. 4. The behavior diagram of the subject employee in the business application process

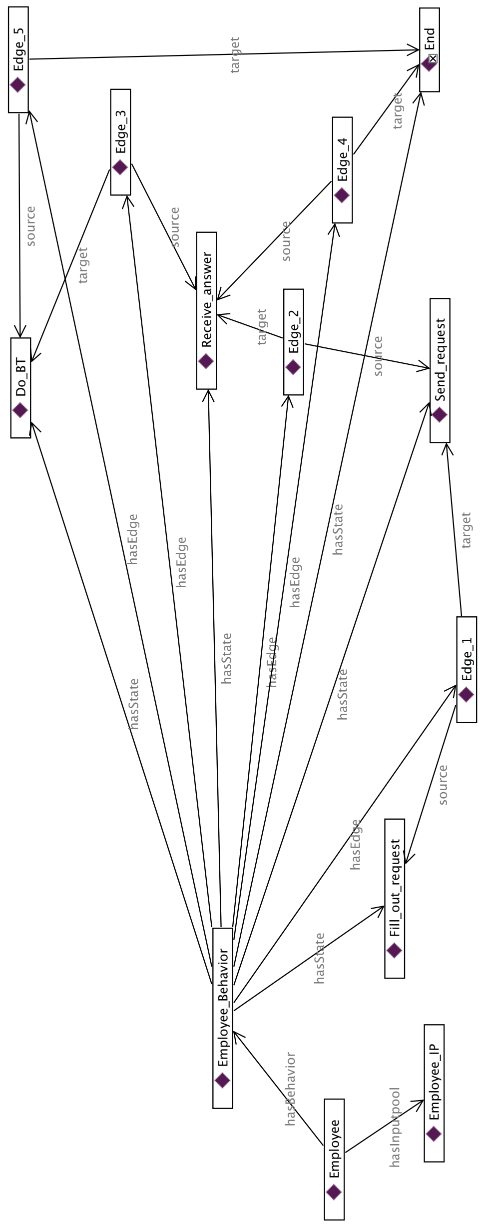


Fig. 5. An ontological representation of the employee's behavior

4 Summary and Future Work

The exchange of models between applications is crucial for interoperability and their reuse. This is especially true for business process models, as they often involve several distributed acting objects distributed over different IT infrastructures. Therefore, we have proposed an ontology for S-BPM processes. It provides a formal specification using the Web Ontology Language.

For future work we plan to continue and to add more S-BPM elements like macros. Specifications for multi-subjects and multi-send and -receive actions are missing in the current ontology version. We intend to integrate these concepts in the ontology and to refine the communication acts as special cases of multi-send and multi-receive, respectively. Furthermore, we plan to define logical rules and further constraints to improve the S-BPM ontology's semantics.

References

1. Fleischmann, A.: What Is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
2. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Brger, E.: Subject-Oriented Business Process Management. Springer, Heidelberg (2012)
3. Borgert, S., Mühlhäuser, M.: A S-BPM Suite for the Execution of Cross Company Subject Oriented Business Processes. In: Nanopoulos, A., Schmidt, W. (eds.) S-BPM ONE 2014. LNBIP, vol. 170, pp. 161–170. Springer, Switzerland (2014)
4. Uschold, M., Grninger, M.: Ontologies: Principles, methods and applications. The Knowledge Engineering Review 11(02), 93–136 (1996)
5. Colomb, R.M.: Ontology and the Semantic Web. Frontiers in Artificial Intelligence and Applications, vol. 156. IOS Press (2007)
6. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record 31(4), 12–17 (2002)
7. Guarino, N., Oberle, D., Staab, S.: What Is an Ontology? In: Staab, S., Rudi Studer, D. (eds.) Handbook on Ontologies, International Handbooks on Information Systems, pp. 1–17. Springer, Heidelberg (2009)
8. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5(2), 199–220 (1993)
9. Borst, W.N.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD thesis, Universiteit Twente, Enschede (September 1997)
10. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25(1-2), 161–197 (1998)
11. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From {SHIQ} and {RDF} to OWL: The making of a Web Ontology Language. Web Semantics: Science, Services and Agents on the World Wide Web 1(1), 7–26 (2003)
12. Noy, N.F., Mcguinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Technical report (2001)
13. Grninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
14. On, J., Choe, Y., Lee, M.: An Abstraction Method of Behaviors for Process Algebra. In: 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 133–138 (2013)

15. Fleischmann, A., Kannengiesser, U., Schmidt, W., Stary, C.: Subject-Oriented Modeling and Execution of Multi-agent Business Processes. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 2, pp. 138–145 (2013)
16. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Brger, E.: A Precise Description of the S-BPM Modeling Method. In: Subject-Oriented Business Process Management, pp. 227–240. Springer, Heidelberg (2012)

Transformation from eEPC to S-BPM: A Case Study

Başak Çakar[1] and Onur Demirörs[2]

[1] ASELSAN Inc., SST-MD-YMM
P.K. 1, 06172 Yenimahalle /Ankara, Turkey
bcakar@aselsan.com.tr
[2] Informatics Institute, Middle East Technical University,
06531 Ankara, Turkey
demirors@metu.edu.tr

Abstract. Business process models are vital assets of any organization. The organizations prefer to use one of the modeling methods and notations according to its features like tool support, size of user base, ease of use. On the other hand, prevalent methods (formalisms) to model these processes change over time. Furthermore, some modeling tools and/or methods might be pulled out of the market completely. In order to handle these kinds of changes easily, migration methods has to be defined. In this work, model transformation is proposed as a method to migrate from eEPC to S-BPM. Direct mapping rules are defined to transform models and the application of these rules is demonstrated by on a real world case study.

Keywords: Process Modeling, eEPC, S-BPM, Model Transformation.

1 Introduction

Business process models describe logical order of activities and dependencies [1]. There are many process modeling languages and organizations use them to reach a comprehensive understanding of organizational processes and analyze them. In the frame of this study, we particularly focus on eEPC and S-BPM languages.

EPC is a business process modeling technique developed by Scheer et al. at the Institute for Information Systems in Germany, in 1990s [2][3]. EPC represents business process as an ordered graph which shows chronological sequence and logical interdependencies between elements. In order to model more complex business processes, EPC notation is extended with additional elements from the organization and data view, which is called eEPC (extended Event-driven Process Chain). It is relatively simple notation to model business processes and highly accepted by the practitioners from diverse areas for business process re-engineering, management and documentation.

S-BPM is a paradigm that is developed by Albert Fleischmann [4] to describe and execute business processes from the perspective of subjects. S-BPM gets inspired from natural languages and the structure of S-BPM is similar to sentence structure of natural languages. According to S-BPM, subjects are active elements in a business

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 53–73, 2014.

process. Therefore, they should be the starting point of the activities (like natural language sentences) [5]. Subjects execute business processes by exchanging messages with each others. Interactions between subjects are shown in the Subject Interaction Diagram (SID). SIDs visualize subjects and data flow (exchange messages) among them. Internal activities of subjects are shown in Subject Behavior Diagram (SBD). S-BPM uses top down approach in determining communication between subjects and uses bottom-up approach in determining internal behavior of subjects.

In the industry EPC is widely accepted during the last decade by means of ARIS toolset. A number of organizations represented their processes using eEPC. However; there is a gap between business and information technology systems in the eEPC since it focuses on notations and this makes difficult to design business processes within IT systems [6]. In order to close that gap, S-BPM has emerged as a new modeling paradigm. S-BPM enables to create dynamic business applications and to integrate them into the existing systems seamlessly. This also provides organizations, which use S-BPM as modeling language, competitive advantage. In addition to this, S-BPM provides a better representation for human interaction patterns and its notation is simple and easy to understand. As an alternative to EPC, S-BPM gaining ground with IT support. Migration of legacy eEPC models requires considerable effort. Furthermore, it's a labor intensive work which increases the usage of personal resources and costs dramatically.

Our goal is to facilitate the transformation process by defining rules with direct mapping. Model transformation is adopted as a main method to provide automation. A model transformation takes a source model and transforms it into a target model by using predefined transformation definition (rules) [7]. The both of the models conform to their respective metamodels. A transformation is defined with respect to the metamodels. The transformation definition is executed on concrete models by a transformation engine. In our case, eEPC is the source model and S-BPM is the target model. Transformation definition is explained in details throughout this document. In order to automate eEPC to S-BPM transformation we developed an extension to existing transformation engine namely UPROM (Unified Process Modeling Tool) developed by Bilgi Grubu and SMRG Research Group. UPROM is based on bflow* toolbox which is an open source modeling tool contributed by at the University of Hamburg and the University of Applied Sciences Emden/Leer [8].

Before the conversion, syntax of the eEPC model is needed to be checked. eEPC syntactical validation rules are stated in Section 3 according to formal definitions given in [9] [10]. The rules to check the validity of generated S-BPM model is also defined in Section 4. These rules include constraints on inter-elements relations and element sequence. In this study we check the validation of input and output models manually. However; after the development of the tool is completed, it is done semi-automatically.

In this paper, we define a set of rules (guidelines) to convert eEPC models to S-BPM models and we validate these by applying our transformation method to a case study. The remainder of this paper is organized as follows: Related work is mentioned in Section 2. Notation of eEPC, its validation rules and metamodel from scratch are presented in Section 3. Notation of S-BPM with validation rules and

S-BPM metamodel from scratch can be found in Section 4. Section 5 presents transformation methodology and rules. Case study can be found in Section 6. Discussions and conclusion are presented in Section 7. Finally references can be found in Section 8.

2 Related Work

In the literature, there are many research studies on EPC and eEPC that are useful for model transformation. W.M.P. van der Aalst gives a formal definition which explains the requirements of an EPC element and defines the core elements as well [9]. Kees van Hee et al. define extended-EPC (eEPC) by providing syntax and semantics [10]. Those studies provide a foundation for our transformation and validation rules.

Transformation techniques to generate models based on existing eEPC models are defined for different formalisms like BPMN [10] and UML [11]. Nüttgens et al. [11] transform EPC models into object oriented models. Relations between EPC and UML diagrams (use case, activity diagram, class diagram and application structure diagram) and the details about which information in an EPC element is used to transform related diagram are defined. In this study, mapping between EPC and UML elements is not stated; only structural transformation approach is explained.

In [12], Tscheschner describes a direct mapping technique to convert eEPC to BPMN and defines transformation rules to map eEPC elements to BPMN elements. On the other hand, eEPC and BPMN differ in their semantics and formalization. Therefore, a complete mapping (structural and semantic) is almost not achievable by solely using direct mapping for each and every component. In order to get complete one, elements of core EPC definition and a subset of eEPC elements are used for mapping.

In [13], Aguilar-Savén compares different modeling languages in terms of message exchange, communication partner's role, process flow and timing, visualization of none sequential process steps, understandability and clear structure of models in order to find the most suitable language for a specific project. According to this study S-BPM is very successful in visualizing exchange messages between subjects. Behavior of the communication partners is also well defined in S-BPM and it has a comprehensive notation. On the other hand, eEPC is stronger than S-BPM in terms of showing the details of process flow and ease of understanding.

In [14] Sneed provides a mapping method for bidirectional transformation between BPMN and S-BPM. Transformation consists of two main parts. In the first part; rules for atomic structures (basic modeling constructs) are defined. The first part of the transformation provides mapping for Subject Behavior Diagrams. In the second part, mapping rules for complex structures are defined. Complex structures are used to visualize the communication view between subjects. In contrast to our study, this study focuses on SID generation.

3 eEPC Notation

The subset of eEPC elements covered by our transformation rules are listed in Figure 1. In order to elaborate elements and their relationships, composed meta-model is depicted in Figure 2 (it's also used for model transformation). It is based on the formal definition of EPC defined by W.M.P. van der Aalst [9] and eEPC Kees van Hee et al. [10].

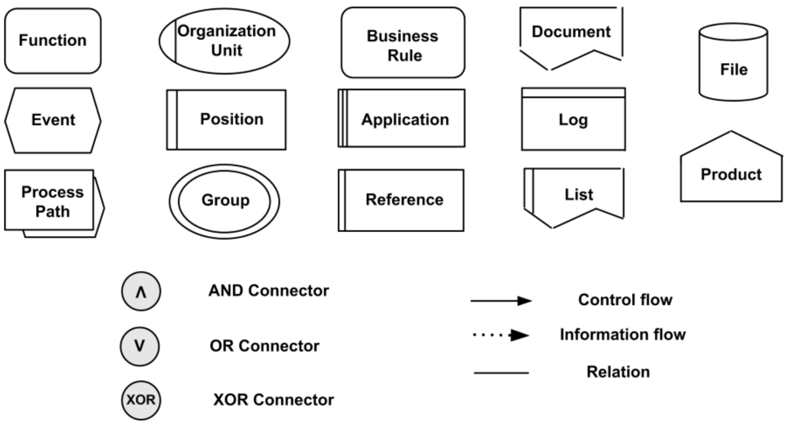


Fig. 1. Covered set of eEPC elements

According to our meta-model, a process consists of at least five process elements (start event, function, end event and control flows between them). Process elements can be workflow elements (function, event, process path, control flow, split connector and join connector) or extended elements (data object, resource object, actor, information flow and relation). eEPC workflow elements are consecutive to each other to form a process flow. Core elements (function, event and process path) are connected to each other by control flows. Data objects (document, list, log, product and file) are connected to functions or process paths via an information flow and they are connected to an actor via a relation. Relation also connects functions and process paths to actors and resource objects (application, reference and business rule).

In order to validate an eEPC diagram the following rules are used [9] [10]:

- There must be at least one start event,
- There must be at least one end event,
- All elements must be connected,
- All functions or process paths must have exactly one incoming and one outgoing control flow,
- Events cannot be consecutive to each other,
- Split connectors must have one incoming control flow and more than one outgoing control flow,
- Join connectors must have more than one incoming control flow and one outgoing control flow.

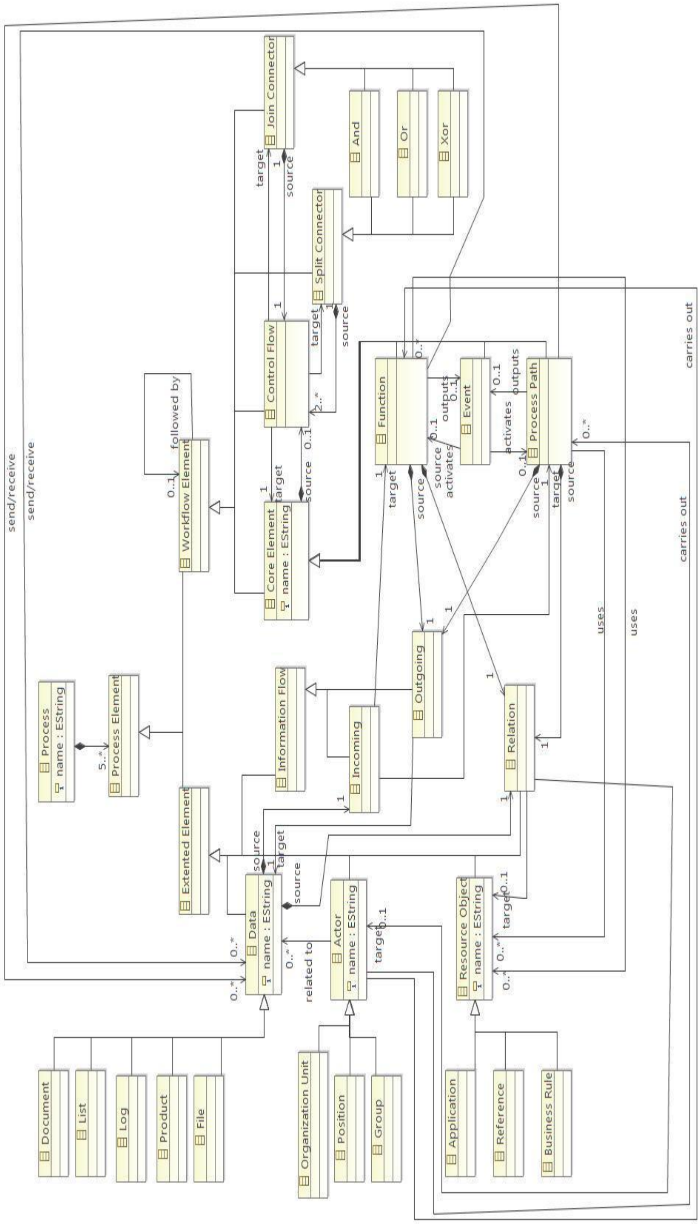


Fig. 2. eEPC Metamodel from scratch

4 S-BPM Notation

Since "eEPC to S-BPM" transformation is an injective non-surjective function, all S-BPM elements are not covered in our mapping. The covered elements of S-BPM elements are listed in Figure 3.

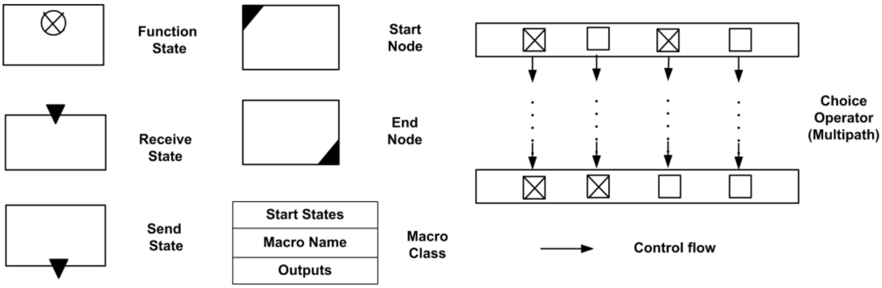


Fig. 3. Covered set of S-BPM elements

SBD elements and their relations are shown in Figure 4 as a meta-model. SBD is composed of at least 3 process elements; start subject state, end subject state and an arc that connected these two states. SBDs consist of subject states (Receive, Send and Function) and outgoing control flows (arcs). A control flow has a label whose value is determined according to its source subject state. "Receive" state depicts the receiving message action and it is followed by a control flow. That flow shows information of sender and received data (Receive Arc). "Send" state is used to show sending message action. It is followed by a control flow (Send Arc) that shows receiver info and sent data. "Function" state with an outgoing arc (State Arc) is used for tasks and post-condition of the function. In addition to those SBD elements, macro classes and choice operators ("Multipath" structure) are also used. Macro classes are used to show sub-processes that repeat in different SBDs in order to avoid repeated patterns. The notation of macro class consists of three parts, in the first part valid start states which activate the sub-process are shown. Name of the macro (sub-process) is shown in the second part. In the final part, the outputs of the sub-process are shown. Choice operator consists of a number of parallel paths which are activated simultaneously. It starts and ends with a bar which includes beginning and end switches for each path.
Validation rules for SBDs:

- There must be at least one start subject state,
- There must be at least one end subject state,
- All elements must be connected,
- All receive nodes must be followed by a receive arc which is annotated by a receive clause,
- All send nodes must be followed by a send arc which is annotated by a send clause,
- All macro classes must have at least one incoming control flow (receive, send or state arc) and at least one outgoing state arc,

- All multipath structures must have exactly one incoming control flow and exactly one outgoing state arc,
- Start bar and end bar of a multipath structure must have the same number of switches.
- All alternate paths in a multipath structure must be start and end with a switch.

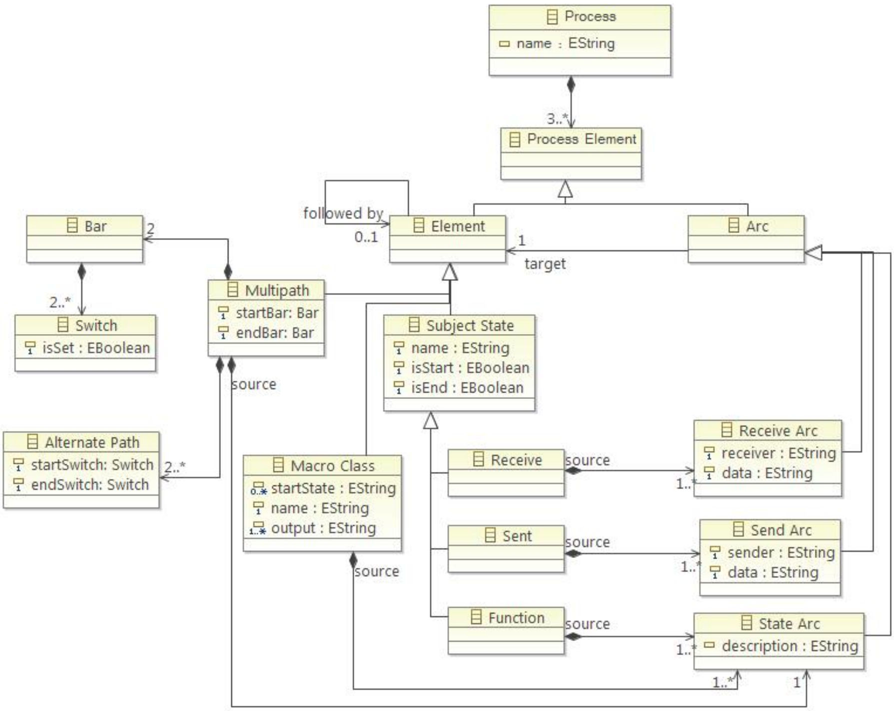


Fig. 4. S-BPM Metamodel from scratch

5 Transformation

eEPC and S-BPM differs the most in the aspect of adopted modeling techniques. eEPC uses flow-oriented approach. Due to that, it is generally considered as a kind of flowchart. It visualizes the sequence of tasks which are performed by different actors. On the other hand, S-BPM uses subject-oriented modeling technique which means that it focuses on subjects (actors) and their relationships. In eEPC, a business process performed by more than one actor can be visualized by one eEPC diagram. However, in S-BPM that business process is visualized by SID in higher level and internal activities of each subject are shown in separate SBDs in lower level. For that reason, our conversion strategy consists of two phases. In the first phase, eEPC model needs to be decomposed into individual eEPC models for each actor. Then, as a second phase, SBDs are generated for each individual eEPC model. Input eEPC diagrams which will be transformed are assumed to be valid according to syntactic and semantic rules defined in [9] [10].

Structural mapping is used as transformation technique; nevertheless there are significant differences between eEPC and S-BPM notations. For transformation, patterns which composed of different elements and transformation rules for those patterns are defined in the following.

5.1 Phase One: Generating Individual eEPCs

As we stated in the previous section, eEPC to eEPC transformation is initially performed in case of more than one actor are responsible for the business process. The tasks of different actors are transformed into individual eEPC diagrams. First, the actors who are only connected to functions are determined and then eEPC diagrams for each of them are generated.

In the individual eEPC generation process, function or function set which are connected to different actors are ignored. In order to generate continuous flow for the actor, the output event of the last function performed by concerning actor and the output event of the last function performed by a different actor are concatenated with an "and" connector(Figure 5).

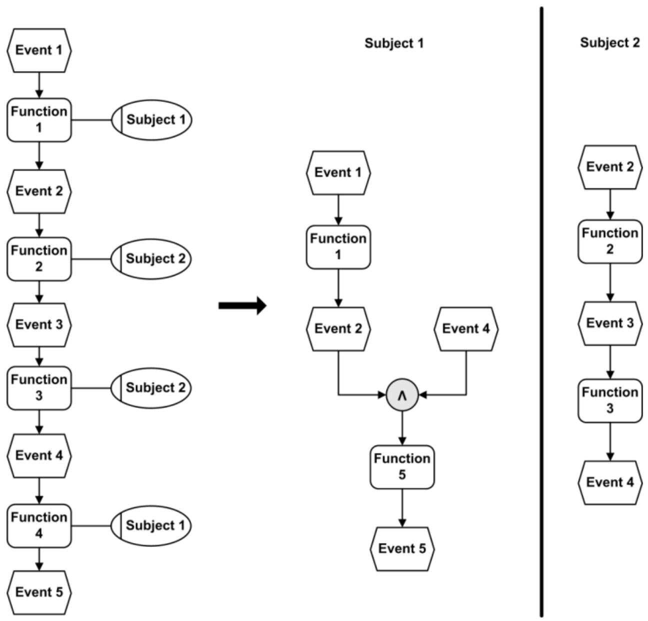


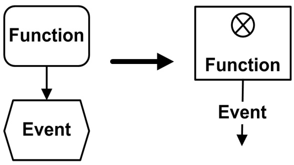
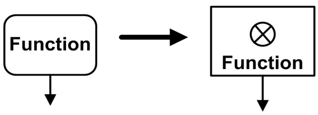
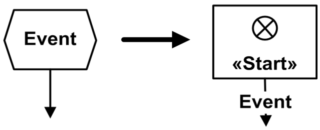
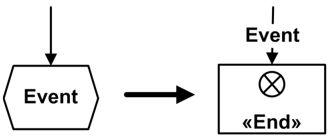
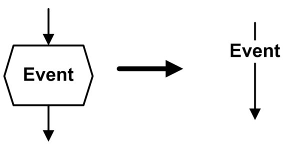
Fig. 5. eEPC to eEPC Transformation

5.2 Phase Two: Generating S-BPM Models

SBD generation starts from the root node and follows through the nodes in sequential order. The events without incoming control flow, the events without outgoing control flow and function-event pairs are taken into account firstly. In the conversion of function-event pairs, relations of the function with other elements (data and resource object) are inspected. Matched patterns (defined in following subsections) converted into respecting target patterns.

Functions and Events. Functions and events are the most crucial elements of eEPC. Functions represent tasks or activities which are executed by organization units, groups or positions. Events show the state of the process. They are triggered by functions and they also trigger functions as well. Function-event pairs show the flow of the business process. Events are problematic in transformation because there is not a corresponding element in S-BPM to map. In order to avoid information lost, following rules are defined in Table 1.

Table 1. Transformation rules for functions and events

Functions followed by an event	
	Functions are directly mapped to performing action (function state) element of S-BPM. Events are shown as text on the outgoing control flow.
Functions without following event	
	Functions without following event are mapped to performing action element and a control flow without any label.
Events without incoming control flow	
	Events without incoming control flow are interpreted as start event. It's mapped to a dummy start function with **«Start»** annotation
Events without outgoing control flow	
	Events without outgoing control flow are interpreted as final event. It's mapped to a dummy end function with **«End»** annotation
Events with incoming and outgoing control flows	
	Events with incoming and outgoing control flows are mapped to a control flow with a label that includes event description.

Data and Subjects. In the eEPC, there are many different types of data and subjects, however the corresponding representations of those objects in S-BPM is not available. Transforming these elements by ignoring their type causes information lost. In order to overcome that, annotations for data and subject types are introduced. Table 2 gives the subject type annotations and Table 3 gives annotations for information, material and resource objects.

Table 2. Subjects and their annotations

Subject Type	Annotation
Organization Unit	«unit»
Group	«group»
Position	«position»

Table 3. Information, Material and Resource Objects and their annotations

Information, Material and Resource Objects	Annotation
Document	«doc »
List	«list »
Product	«product»
File	«file»
Log	«log »
Application	«app»
Reference	«ref »
Business rule	«rule»

Document, list, product, file and log can be seen as a data object. However; application, reference and business rule are resource objects. Applications are systems and supports functions for execution. References (laws, regulations, standards, guidelines, etc…) are used to provide information to execute related function. Business rules restrict the operations of functions. For resource objects, notation with "Used" keyword and respective annotation («app», «ref» or «rule») is used. The description of resource object is also given as a part of this notation. Resource object notation is connected to functions with a dotted line (Refer to Figure 6).

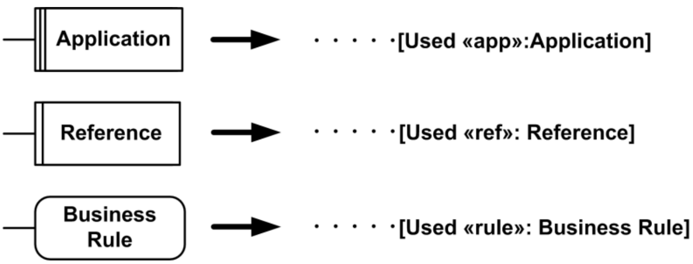
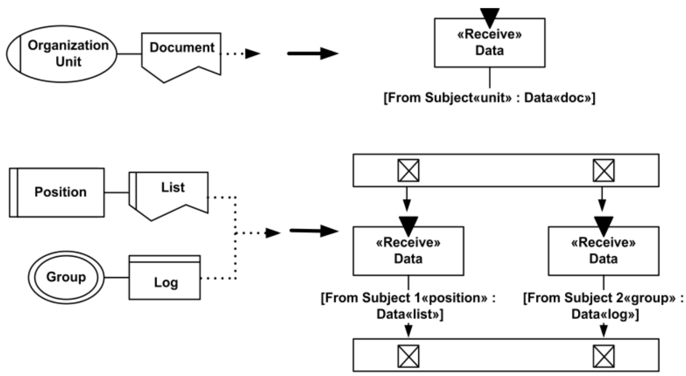
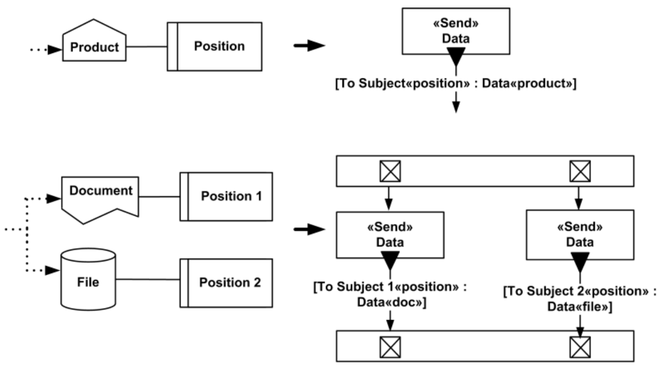
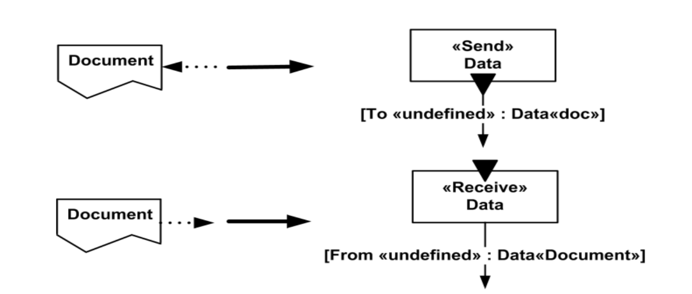


Fig. 6. Resource object transformation rules

Receive/Send Data. In the eEPC, there are data objects which are used by functions (input) or produced by functions (output). Direction of the information flow between data object and function shows whether it is a receiving or sending data. Data objects do not affect the process flow. In SBD, data objects are stated on control flow as a label. The value of that label is set according to direction of the flow (sent or received data). "Receiving Message" and "Sending Message" elements are used during the conversion of data-subject pairs. Since the data objects are not ordered in the process flow, it's assumed that all receiving messages take place before the concerning function and all sending messages come after the function in the flow. The order of received and sent messages can be stated by the user before the transformation. In the following patterns for sending and receiving messages and their transformation rules with semantics are explained in Table 4.

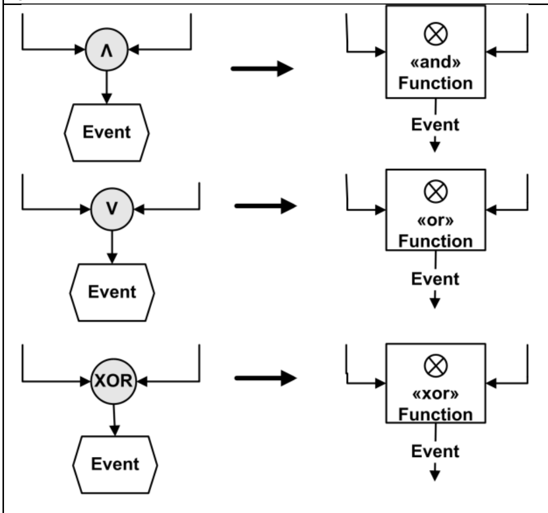
Table 4. Transformation rules for data objects and subjects

Data with outgoing information flow	
	Name of the element is given as **"«Receive» data name"** automatically. If there is more than one data object received by a function, choice operator to combine receiving messages is used.
Data with incoming information flow	
	Name of the element is given as **"«Send» data name"** automatically. If there is more than one data object sent by a function, choice operator to combine sending messages is used.

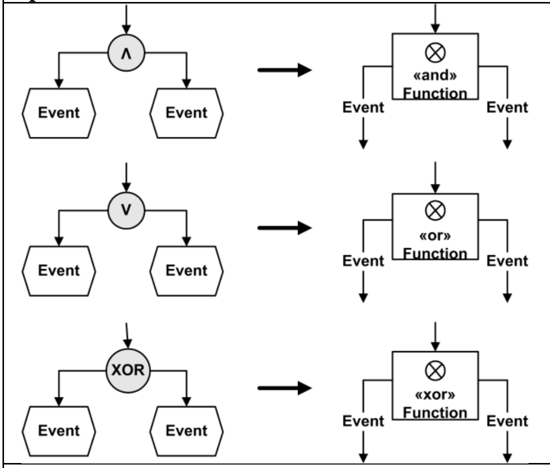
Data without any subject relation	
	**«undefined»** annotation is used for missing subjects.

Logical Connectors. Logical connectors show the logical relationships between functions and events in the control flow. They are used to split one control flow into two or more control flows or to concatenate two or more control flows. In the eEPC there are three types of logical connectors; "and", "or" and "exclusive-or". Since there is not any element in S-BPM with the same behavior to map logical connectors, conversion of them is the most problematic part of the transformation process. Three new functions with "«and»", "«or»" and "«xor»" annotations are defined in order to use as logical connectors. In S-BPM functions with more than one incoming control flows and more than one outgoing control flows are possible. Therefore, "Performing Action" element with a respective annotation is used to depict logical connectors. In Table 5, logical connectors grouped by behavior are mapped to S-BPM elements with brief descriptions.

Table 5. Transformationrules for logical connectors

Join Functions	
	Join Functions connectors joins the controls flows that ends with functions and activates the result event. Join Functions is mapped to performing action elements with «and», «or» or «xor» annotations according to logical connectors. These elements have incoming control flows without any label and an outgoing control flow with a label that includes the description of the output event of join function.

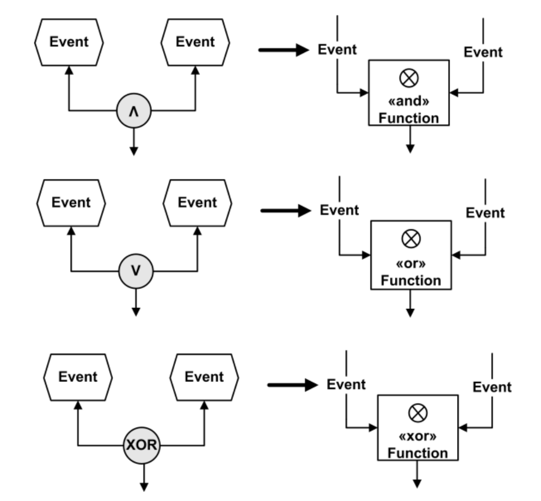
Split Function



Split Function connectors activate two or more post-conditions (events) of a function.

Split Function is mapped to performing action element with «and», «or» or «xor» annotations. These elements have one incoming control flow and outgoing control flows with a label that includes event description.

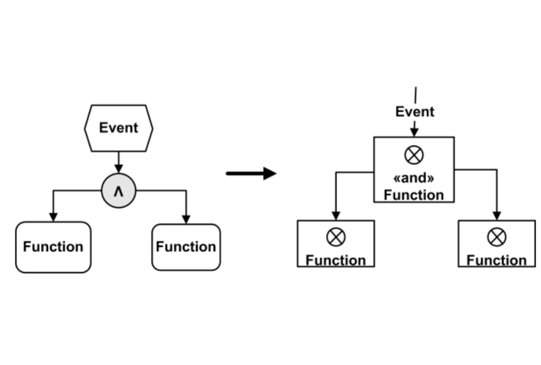
Join Events



Join Events connectors concatenate preconditions (events) and activate the following function.

Join Events is mapped to performing action element with «and», «or» or «xor» annotations. These elements have incoming control flows with a label that includes the description of precondition (event) and an outgoing control flow without any label.

Split Event



Split Event connector activates two or more functions when the precondition (event) is satisfied.

For Split Event, performing action element with «and» annotation is used. It has an incoming control flow with a label that includes the description of precondition (event) and outgoing control flows without any label.

In order to transform of “and” block (“and” connectors using in pairs), choice operator is used. Choice operator leaves the execution sequence of the activities to the discretion of the respective person, can be preferred for transformation. However, usage of choice operator is directly related to structure of eEPC model. In eEPC logical connectors can be used singly or in pairs. If there is only one connector is used for parallel processes in eEPC, there is no way to determine the end of those processes. Therefore, predefined mapping rule is applied. If “and” connectors are used in pairs (logical elements match and parallel paths are opened and closed with the same element), they are transformed into choice operator (Figure 7). In most situations, this goes well with the intended semantics of “and” in EPC process models when the following functions are to be executed by the same person. However, in order to identify the behavioral difference in “and” connectors, modelers have to define whether logical connector starts/ends an alternative path block.

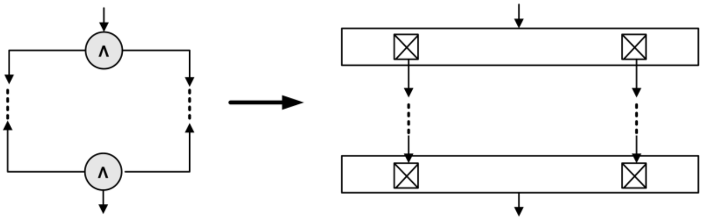


Fig. 7. Mapping rule for “And” connector block

Process Path. Process path navigates the control flow to sub-processes. Transformation rules, defined for function element, are also applicable to process path element. In contrast to function element, process path element is mapped to macro class. In the notation of macro class element, invalid start states, name of the macro and possible output transitions are shown respectively. In transformation, firstly the expansion of the process is sought in the project folder. If the model of the sub-process is in the project folder, start states of sub-process are used as valid start states of macro class and end states of sub-process are used as possible output transitions (Figure 8). If the model of the sub-process is not in the project folder, user can define start states and output transitions manually, or «undefined» annotation is used for missing information.

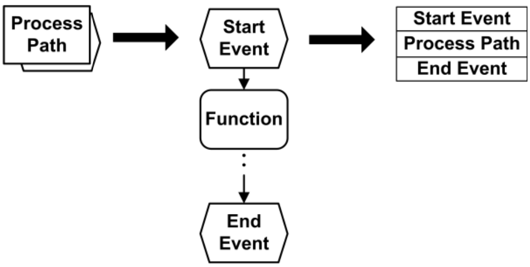


Fig. 8. Transformation rule of Process Path element

6 Case Study

eEPC models generated by Bilgi Grubu in UPROM within a project, which is initiated by Ministry of Development of the Republic of Turkey, are transformed to S-BPM. The aim of the project is to define business processes and to support those processes with IT Systems. Within the scope of the project, main and supporting business processes which belong to development agencies are modeled [15]. Two essential processes of Archive Management System are selected for case study. Figure 9 shows activity and data (sent/received) flow of archiving process. In Figure 10, eEPC model of the Acquiring archive material process is given.

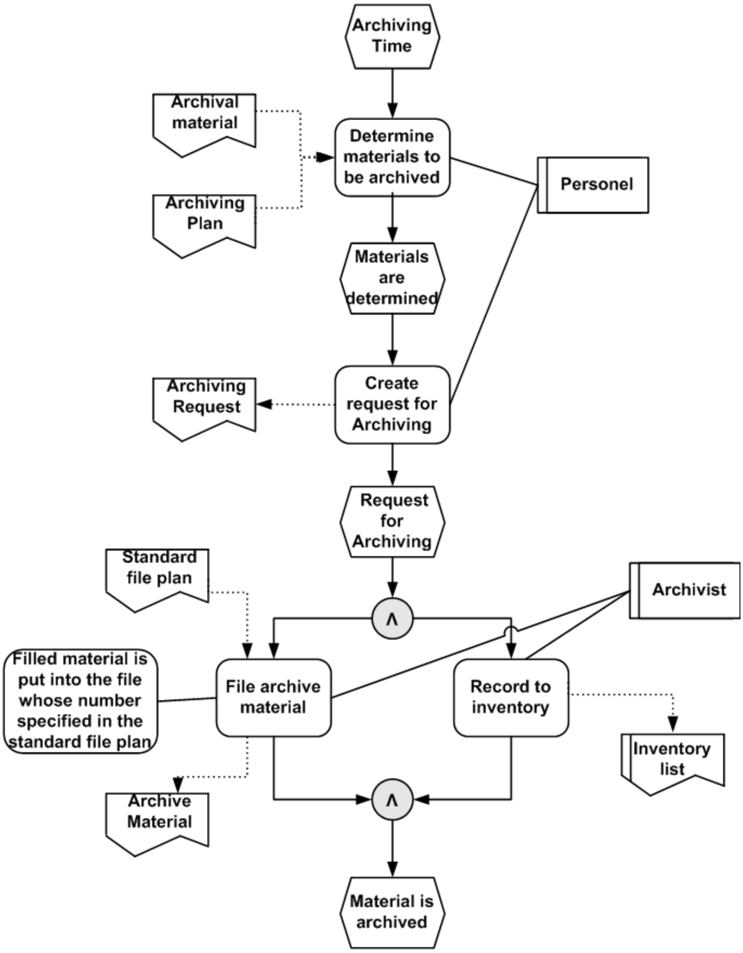


Fig. 9. Archiving Process

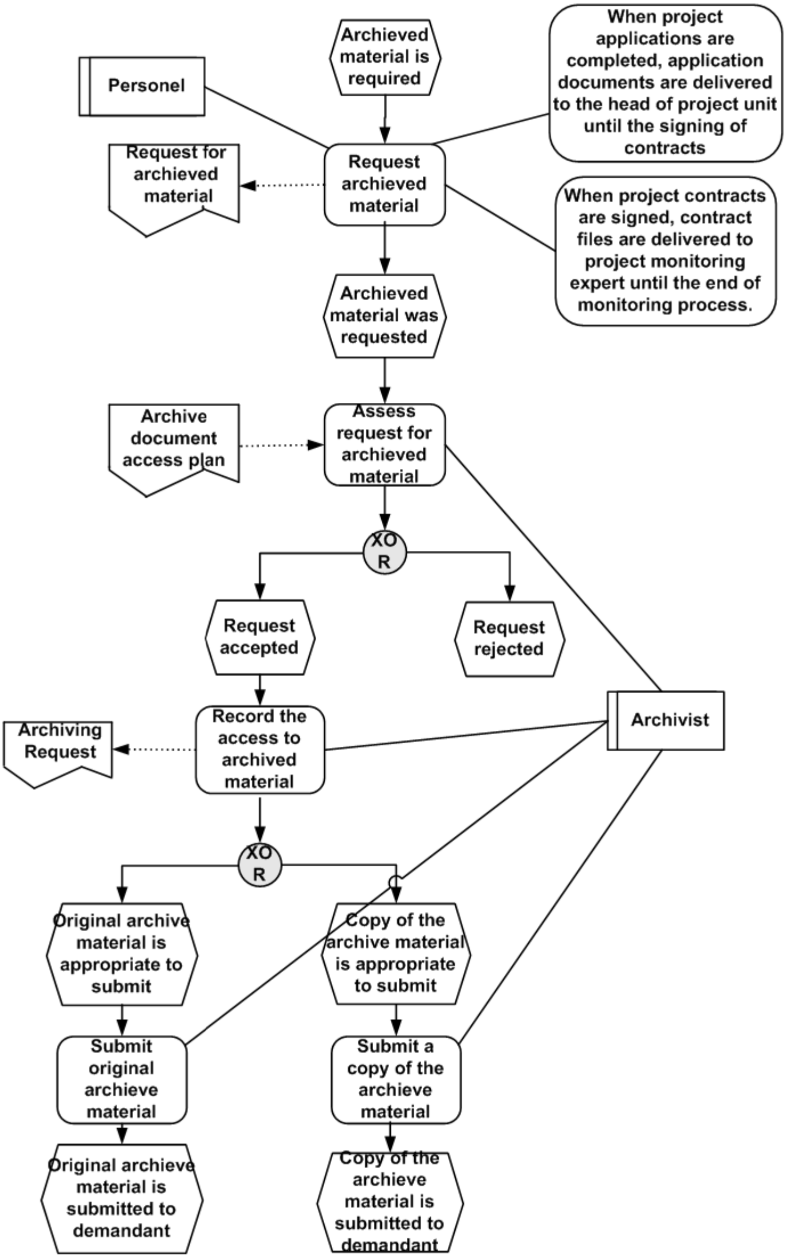


Fig. 10. Acquiring Archive Material Process

In both of them, two actors Personnel and Archivist takes a role to accomplish the process. Thus in the first part of the transformation, process are divided into two sub-processes by omitting actors which are directly connected to functions from the model. Figure 11.a-12.a shows the sub-processes of Archiving process and Figure 13.a-14.a shows the sub-processes of Acquiring archive material process.

In the second part of the transformation SBD for each sub-process are generated by using the rules defined in section 5.2. Thus, only the name of the applied rule is given in the explanations of transformations. While transforming Personnel_Archiving Process, the rules “Events without incoming control flow”, “Data with outgoing information flow”, “Functions followed by an event”, “Data with outgoing information flow” and “Events without outgoing control flow” are used respectively.(Figure 11.a-11.b).

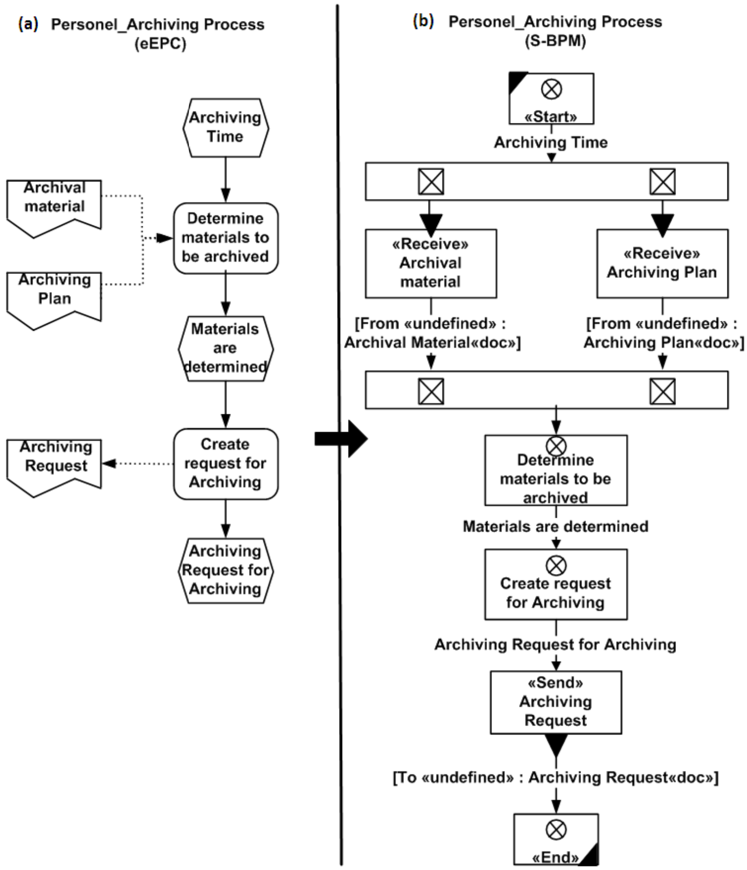


Fig. 11. Transformation of Archiving Process for Personnel

In the transformation of Archivist_Archiving Process of Archivist subject, “Events without incoming control flow”, “Split Event”, “Data with outgoing information flow”, “Functions without following event”, “Resource object transformation”, ”Data with incoming information flow”, “Join Functions” and “Events without outgoing control flow” rules are used (Figure 12.a-12.b).

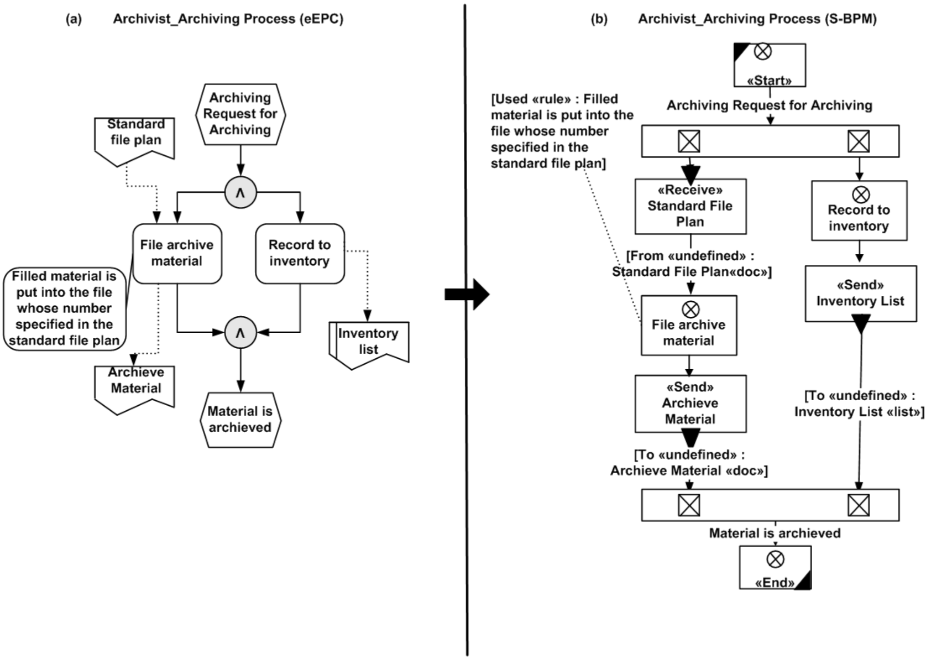


Fig. 12. Transformation of Archiving Process for Archivist

"Events without incoming control flow", "Data with incoming information flow", "Resource object transformation", "Functions followed by an event" and "Events without outgoing control flow" are used in given order to transform Acquiring archive material process in the view of personnel (Figure 13.a-13.b).

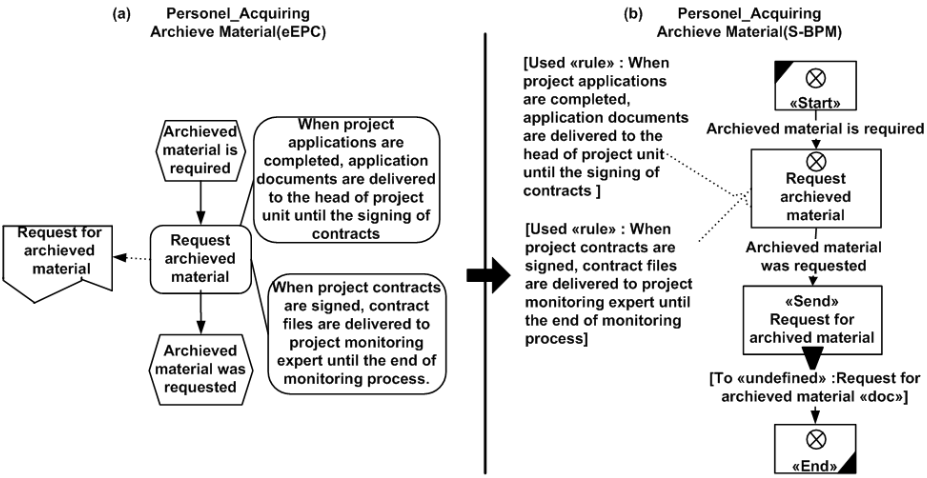


Fig. 13. Transformation of Acquiring Archived Material Process for Personnel

S-BPM model of Achivist_Acquiring archive material process is generated by practicing the following rules respectively; “Events without incoming control flow”, “Data with outgoing information flow”, “Functions without following event”, “Split Function”, “Events without outgoing control flow”, “Functions without following event”, “Data with incoming information flow”, “Split Function”, “Functions followed by an event”, “Events without outgoing control flow” (Figure 14.a-14.b).

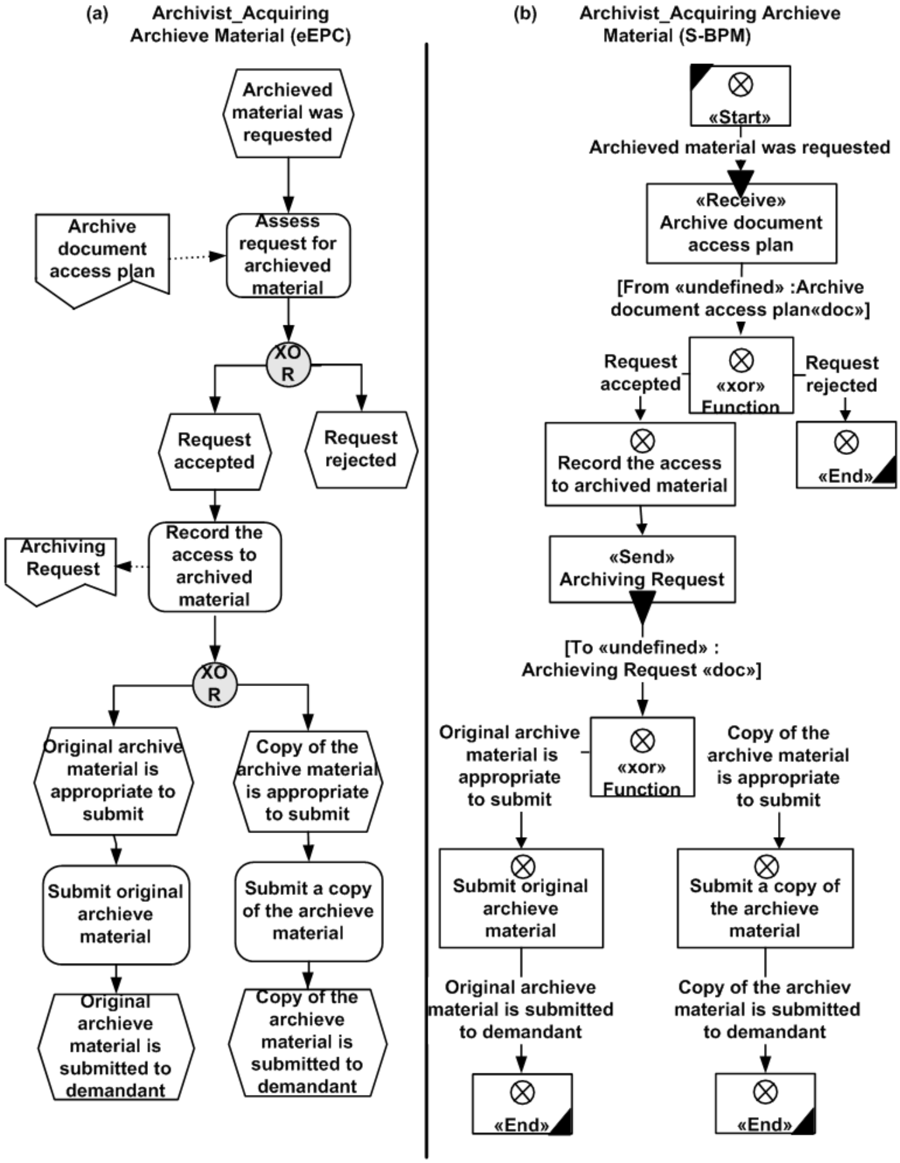


Fig. 14. Transformation of Acquiring Archived Material Process for Archivist

The validity of output S-BPM models are checked manually by traversing the validation rules given in section 3.

7 Discussion and Conclusion

In our case study, eEPC model has been transformed to S-BPM without any information lost by using the defined transformation rules. The generated S-BPM models are valid and semantics of the input model are preserved in output models. As a result of eEPC to S-BPM transformation, individual models for each subject are generated. The output model names give information about subject and related process. However, interactions of the subjects and message flow between them are missing. User can only get this information by tracking the generated models and finding the common points (events) manually. In order to provide traceability, SBD for input process should also be generated. That generation is out of this paper's scope, it is left as a future work.

Duplication is another problem which occurs in the conversion of sending and receiving messages. If there is a function which includes "receive" keyword in its description and an incoming information flow connected to it, information flow part is converted as receive message with «receive» annotation and conversion of function part also includes "receive" keyword which refers to the same data. The duplication problem also occurs in the conversion of outgoing information flow connected to a function that contains "Send" keyword. In order to avoid this problem, description of functions is also taken into account. Information flow and related function is considered as a different pattern and that pattern is mapped to "Receive Message" or "Send Message" element directly. On the other hand, there is no way to show all flow information in a single S-BPM element (e.g. send-receive element) if there are more than one information flow related to the function. Thus, possible duplicates are not handled in order not to complicate transformation.

During the case study, we have had some observations for more understandable model generations. These remarks can be summarized as follows:

- Each function should be followed by an event and each function should be triggered by an event in the input model. Otherwise, null transitions will be occurred in the model.
- Each data object should be related to a subject. Otherwise, subject information will be marked as «undefined».
- In case of eEPC model belongs to only one actor, no information lost occurs in transformation. Otherwise, subject's interactions will be lost.

As a conclusion, in this paper we have defined metamodels (based on MOF) and validation rules for eEPC and S-BPM. The main contribution of the study is defining a transformation approach and it is divided into two phases. In the first phase, individual eEPC models are generated and then eEPC to S-BPM transformation is performed as a second phase. Common eEPC patterns and rules for each of them are defined as guidelines. These guidelines are validated on a real world case study and they are evaluated to be used in the development of the extension to UPROM (work in progress).

References

1. Aguilar-Savén, R.S.: Business Process Modeling: Review and Framework. International Journal of Production Economics (90), 129–149 (2004)
2. Scheer, A.W: ARIS- Modeling Methods, Meta-models, Applications. Springer, Berlin (1998) (in German)
3. Scheer, A.W.: ARIS - Business Process Modeling, 3rd edn. Springer (1999)
4. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management (2012)
5. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
6. Singer, R., Zinser, E.: Business Process Management — S-BPM A New Paradigm for Competitive Advantage? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE 2009. CCIS, vol. 85, pp. 48–70. Springer, Heidelberg (2010)
7. Sendall, S., Kozaczynski, W.: Model Transformation the Heart and Soul of Model-Driven Software Development. IEEE Software 20(5), 42–45 (2003)
8. Böhme, C.: bflow* Toolbox-an Open-Source Modeling Tool (2011)
9. van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41(10), 639–650 (1999)
10. van Hee, K.M., Oanea, O., Sidorova, N.: Colored Petri Nets to Verify Extended Event-Driven Process Chains. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 183–201. Springer, Heidelberg (2005)
11. Nüttgens, M., Feld, T., Zimmermann, V.: Business Process Modeling with EPC and UML: Transformation or Integration? In: Schader, M., Korthaus, A. (eds.) The Unified Modeling Language—Technical Aspects and Applications. Physica-Verlag, Heidelberg (1998)
12. Tscheschner, W.: Transformation from EPC to BPMN. Oryx Research. Potsdam, Germany (2010)
13. Handy, B., Dirndorfer, M., Schneeberger, J., Fischer, H.: Methods of Process Modeling in the Context of Civil Services by the Example of German Notaries. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 281–295. Springer, Heidelberg (2011)
14. Sneed, S.: Mapping Possibilities of S-BPM and BPMN 2.0. In: Oppl, S., Fleischmann, A. (eds.) S-BPM ONE 2012. CCIS, vol. 284, pp. 91–105. Springer, Heidelberg (2012)
15. Coşkunçay, A., Aysolmaz, B., Demirörs, O., Bilen, Ö., Doğan, İ.: An Approach for Concurrent Business Process Modeling and Requirements Analysis. In: Symposium on Software Quality and Software Development Tools, İstanbul, Turkey (2010)

Design of a Subject-Oriented Reference Model for Change Management

Patrick Garon[1], Arnd Neumann[2], and Frank Bensberg[3]

[1]T-Systems International GmbH, Leipzig, Germany
[2]T-Systems International GmbH, Leinfelden-Echterdingen, Germany
{Patrick.Garon,Arnd.Neumann}@T-Systems.com
[3]Hochschule für Telekommunikation Leipzig (HfTL), Dept. of Business, Leipzig, Germany
Bensberg@hft-leipzig.de

Abstract. The ITIL is an established presence in ICT practice, which allows some scope for domain-specific implementation. The complexity of existing IT solutions and ever shorter development times are causing increasing problems in implementing the ITIL processes. This paper takes up this problem with reference to Change Management and shows how it has been addressed in practice with the aid of subject-oriented Business Process Management (S-BPM) and reference modeling. The description of the problem is followed by a discussion of the conceptual foundations of the ITIL and the central features of S-BPM, which is used to design a conceptual reference model for the domain of Change Management. This model is presented with focus on the involved subjects, their interactions and behavioral patterns. We substantiate the practicability of the model using a generic example and finally conclude potential benefits.

Keywords: Change Management, Reference Modeling, ITIL, S-BPM, IT Service Management.

1 Introduction

Because of technological and economic developments, IT service providers have to produce more and more complex IT solutions in shorter and shorter development cycles if they are to succeed in the marketplace (see [24], p. 247). This phenomenon is currently referred to as “dynaxity,” a mixture of dynamic change and complexity [11]. It can also be observed in practice that the processes of producing IT services are increasingly driven by internal IT Governance policies, while the resulting customer satisfaction is only incidentally addressed. In light of this situation, IT service providers can fall into a complexity trap: the increased complexity of the services demanded by the market and the host of internal rules to be complied with lead to rising costs, so the resulting gains show a declining trend.

The reaction of management to this development is often to reduce the available staff resources to safeguard the company’s financial results. To tackle this ever-growing complexity, the players who have to modify and implement the processes involved in IT service production are then often left managing shadow processes [22].

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 74–88, 2014.

Because of a lack of transparency, these often escape management control, so quality problems are almost bound to arise. In the worst case they may fall into a vicious spiral, sustained by declining financial results on the one hand and increasing pressure from management to impose control on the other.

In particular, the identified complexity trap exerts massive pressure upon the domain of Change Management, which is responsible for the efficient handling of all changes of the IT infrastructure. Typically, such tasks require specific and intense communication between different organizational units, such that enterprises are confronted with high coordination costs [14]. In order to gain efficiency in the domain of Change Management, process orientation and adequate process-oriented information systems seem to be suitable means [12].

Widely-used IT management frameworks such as the ITIL offer no constructive solution to unfold process orientation. The ITIL does define Change Management tasks with which tactical and operational changes to IT services can be organized, coordinated and controlled (see [25], p. 181). However, it provides no detailed process models covering the logical sequence of the necessary Change Management activities (see [24], p. 122ff., and [13]). Given the dynamic nature of the market outlined above, such process models must be extremely adaptable [9] and capable of being rapidly rolled out into production in IT organizations [6].

To address the set of problems outlined here, this paper presents a conceptual reference model [21] for ITIL-based Change Management. A reference model has the fundamental potential to reduce the production costs of an implementable Change Management process that meets the company's needs or to optimize an existing Change Management function [1,4]. To meet the requirements of the domains in question, we have based the design of the reference model on the concept of subject-oriented Business Process Management (S-BPM) [7]. As well as transparent modeling of the players involved, along with their activities, this allows processes to be executed without any programming effort (model-to-execute). This means that the processes can be tested and adapted by the players concerned, providing a high degree of organizational flexibility and acceptance. The resulting subject-oriented model for Change Management has been developed in an evolutionary manner over several years within an IT service organization, and is presented here in a generalized form.

This paper begins by introducing some key principles of Change Management. Based on the central features of S-BPM, we then discuss the subjects, interactions and relevant behavioral aspects of the reference model. In order to substantiate the practicability of the designed model, we introduce an example scenario, which subsequently is expanded by a data layer. Finally, a summary is presented and potential benefits of the designed artifacts are concluded.

2 ITIL Principles

The IT Infrastructure Library (ITIL) is a process-oriented collection of best practices for planning, monitoring and managing IT services [25]. The ITIL has developed into a international de-facto standard for IT service providers (see [24], p. 122) and enables IT services to be aligned with current and future requirements.

The core of the ITIL comprises five modules: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement. Change Management forms part of the Service Transition area and is intended to keep the impact of changes on all operational services, and hence on IT costs, to a minimum (see [25], p. 195ff.). It coordinates, controls and establishes the activities needed to manage Changes. The area of Change Management encompasses the live operation that is relevant to service provision, and the associated administrative environment (see [15], p. 45). Each individual Change is entered in the Change Record (RC). This document contains all management-related information, and the history of a particular Change. A Change is a modification or functional enhancement to an existing system (e.g., hardware or software, or IT infrastructure). The details of a Change are gathered as the process is carried out and are then provided by the Request for Change (RFC) [20].

To handle the operation of Change Management, the ITIL provides for various roles (see [18], p. 49). The Change Manager is the highest operational instance and assumes the responsibility for the process. The Change Coordinator supports the work of the Change Manager by helping to define and describe the change in detail. The Change Advisory Board (CAB) is a group of decision-makers (technical and administrative) with a suitable overview of the whole system, which examines the impact of each proposed Change and supports the implementation of the Change (see [15], p. 98).

Although the ITIL claims to provide a comprehensive approach to IT Service Management, clear gaps appear when we compare the contents of the ITIL with the tasks of production management (see [24], p. 123). There are significant shortcomings in the management of the production process in particular. For example, the ITIL provides little or no coverage for production planning and production control. Only production change has reasonable coverage, so there is scope for improvement overall.

The ITIL does not include any industry- or company-specific recommendations. Although the generic description of the best practices discussed in the ITIL allows them to be adapted to a wide range of application areas, this adaptation has to be performed by the company itself (see [24], p. 125ff.). The ITIL is a good framework for creating standardized processes within a company, but the staff are then forced to work according to certain rules which must be compliant with the IT Governance policy, which results in greater organizational costs.

3 Reference Model for Change Management

3.1 S-BPM Fundamentals

S-BPM is a comprehensive method for the correct implementation of an integrated BPM approach (see [7], p. 20). This approach can be considered to be integrated as the individual phases (analysis, modeling, validation, optimization, implementation, verification and acceptance, and operation and monitoring) form a closed loop (see [16], p. 14). The S-BPM notation is based on natural language with complete sentences comprising subject, predicate and object, and in contrast to traditional BPM, it focuses on the subject of a process. Modeling in S-BPM takes places in two stages (see [5], p. 54):

1. Identification of the subjects and their interactions → What are our subjects?
2. Definition of the subject behavior → What does a subject do, and when does it send or receive a message?

The result of the first step is a *subject interaction diagram* (SID), which forms a model with explicit communication relationships structured by the involved subjects. Fig. 1 shows an example subject interaction diagram created with the S-BPM modeling tool Metasonic Suite V. 5.0.1 [19].

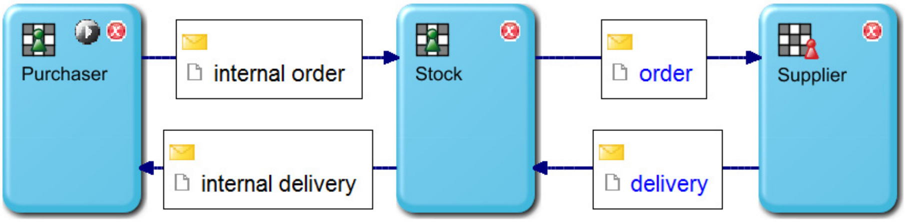


Fig. 1. Example subject interaction diagram (SID)

In the second step, the subjects are refined by modeling their behavior as a sequence of activities and interactions with the aid of states and transitions. Several subjects can act in parallel and synchronize themselves by means of messages (see [5], p. 54ff.). The behavior of every subject can be specified using three states and transitions: *Send*, *Receive* and *Function*. Five symbols are used in S-BPM notation (Fig. 2).

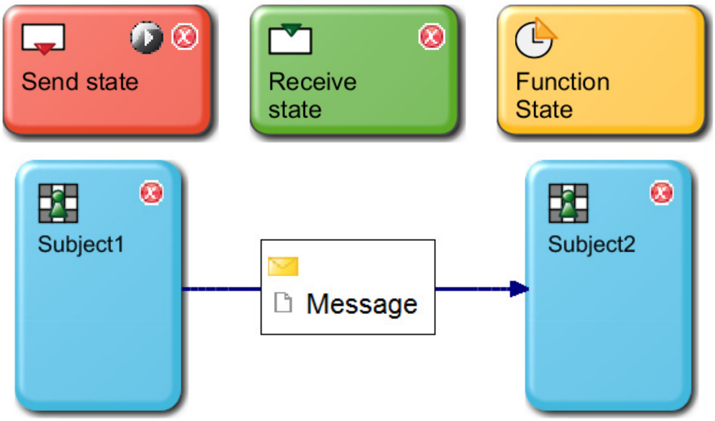


Fig. 2. Symbols in S-BPM notation

The *subject behavior diagram* (SBD) shown in Figure 3 models the processing of an order. After the receipt of the order (Receive state), the stock level is checked (Function state) and the control flow branches. Individual states can be refined, e.g., to assign the activities to external applications, information objects or business rules (see [5], p. 56).

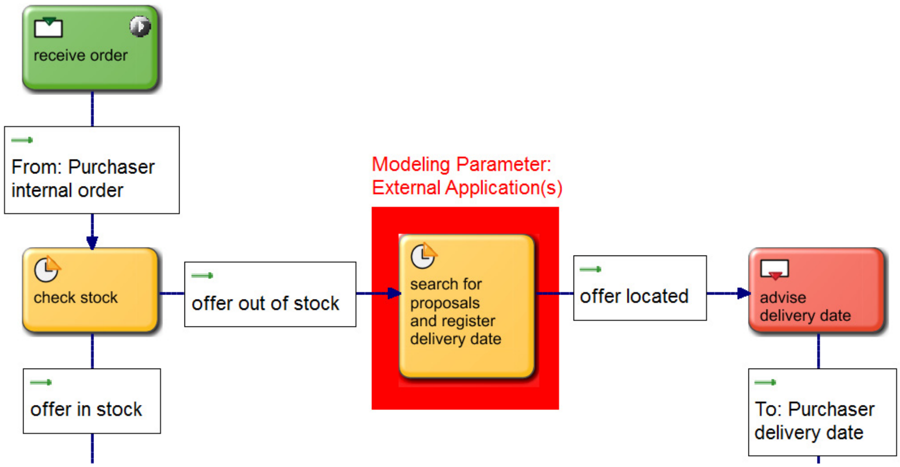


Fig. 3. Example subject behavior diagram (SBD)

The simple notation used in S-BPM makes the resulting models very easy to understand. The models can also be executed without any additional programming effort (model-to-execute). This allows models to be tested in the actual work process and adapted to suit. These attributes are the reason why S-BPM is applied below to the domain of Change Management.

3.2 Subjects and Interactions within Change Management

The subject-oriented reference model (SID) shown in Figure 4 is based on a Change Management process that has been developed and optimized over many years by a system integrator for the live running of SAP Basis services. Because of the practical application, some elements differ from the relevant literature, but they are tried and tested and therefore constitute an adequate reference for different domains.

To reflect the generic character of the reference modeling activity, we will dispense with describing all the elements of Change Management in detail. Because of the many special rules and actions in the case of an incident, this area is not considered at this time.

From the SID we can see the following subjects involved in Change Management: the Change Requester, Change Coordinator, Change Implementer, Change Manager and Change Approver.

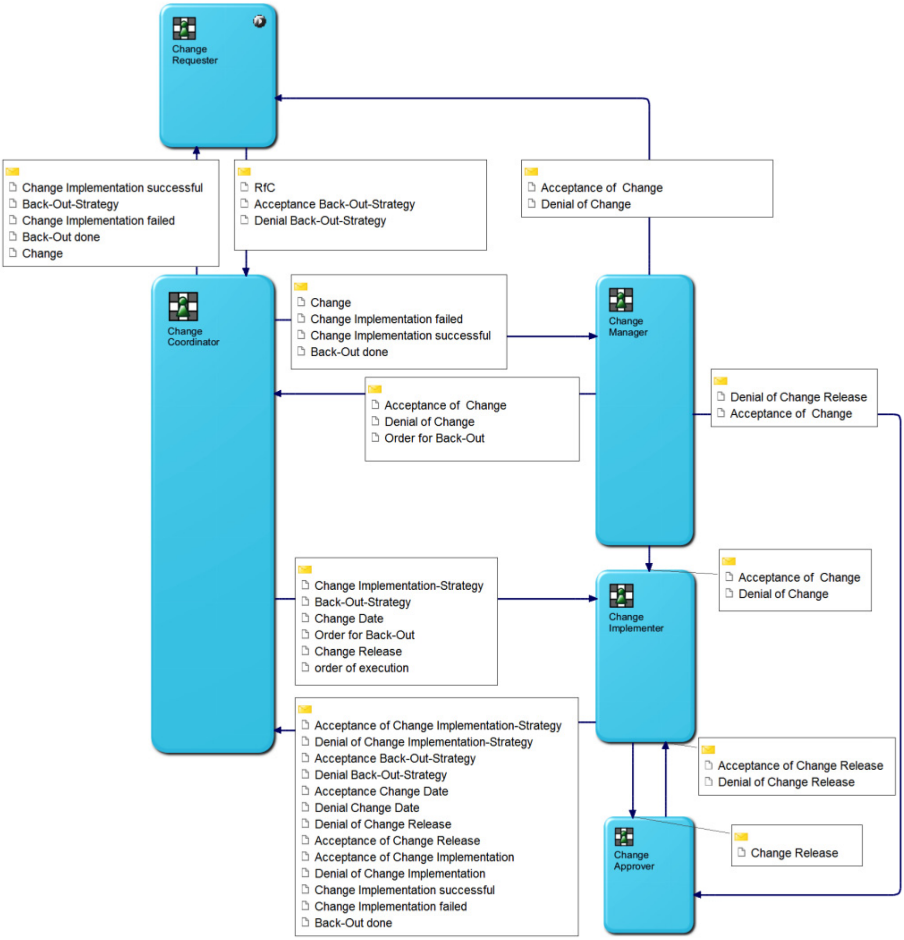


Fig. 4. Subject interaction diagram for the Change Management process

Changes may be internal or external. In the external case, the *Change Requester* will engage in an active dialog with the customer, enter the customer requirements in the Request for Change (RfC) and provide this to the Change Coordinator. In the internal case, it has to be decided whether the functional area into which the Change falls should handle the tasks of the Change Requester itself or entrust the production of the RfC to an external person or group. The Change Requester is then informed of significant elements of the change or asked for approval.

In this model, the function of *Change Manager* is of a purely administrative nature. This person only intervenes to provide guidance and instruction when the Change absolutely needs it: for example, where Service Level Agreements are not adhered to or an unsuitable time window has been chosen. The Change Manager only comes in at a relatively late stage, as most of the organizational tasks are handled by the

Change Coordinator. In S-BPM subjects can be defined as multi-subjects, allowing them to represent the behavior of a team or a group. The Change Manager, modeled as an administrative multi-subject, may represent the functions of the CAB, covering all significant groups and persons such as administrators, technicians, Service Level Managers, operations staff, etc. That is why the CAB is not modeled. Whether a CAB subject is needed is a decision for the individual company.

The Change Coordinator and the Change Implementer play a crucial role in Change Management, as can be seen from the large number of transitions between these two subjects. The *Change Coordinator* is responsible for organizing and coordinating most of the steps needed to produce and implement the Change. They are in contact with almost all subjects, to gather all the necessary technical, administrative and operational information on the Change. In close collaboration with the Change Implementer they draw up the Back-Out Strategy and the Change Strategy, containing all significant details of the Change, such as its criticality, priority, affected components, procedure, etc. The Change Coordinator also handles the scheduling and publishes a Change Release once all the information is to hand. Once the Change Release has been confirmed by the Change Manager, there is no longer anything to stop the Change being executed, and the Order for Change is given directly to the *Change Implementer*. In the event of failure, the Change Implementer immediately informs the Change Coordinator, who in turn communicates this result and awaits a back-out order from the Change Manager in order to have the original situation restored by the Change Implementer.

The *Change Approver* represents the individual functions involved and bears the primary responsibility for checking the Change at the technical level. The Change Approver may be one or more people.

The relationships between the Change Coordinator (CC) and Change Implementer (CI) subjects from the outline reference model are described in more detail below by way of example.

3.3 Examples of Behavior by Selected Actors

The CC receives an RfC from the Change Requester (CR). Once he has checked the format and content of the data in the RfC, he prepares a Change Implementation Strategy as shown in Figure 5. The Change Implementation Strategy defines the technical requirements and the resources and functions required, the criticality, the service impact and the Change type. Apart from detailed descriptions and definite categorizations, the CC and CI then cover different aspects of the Change Strategy, which need to be drawn up and confirmed in a relatively short time in line with the Change objective.

When the CC has completed his part of the Change Implementation Strategy, he sends it to the CI. As can be seen from Figure 6, the CI receives the Change Implementation Strategy prepared by the CC, checks it for completeness and makes any necessary changes. He then accepts or rejects the Change Strategy. In both cases, the information goes back to the CC, who either starts the next task in the process or has

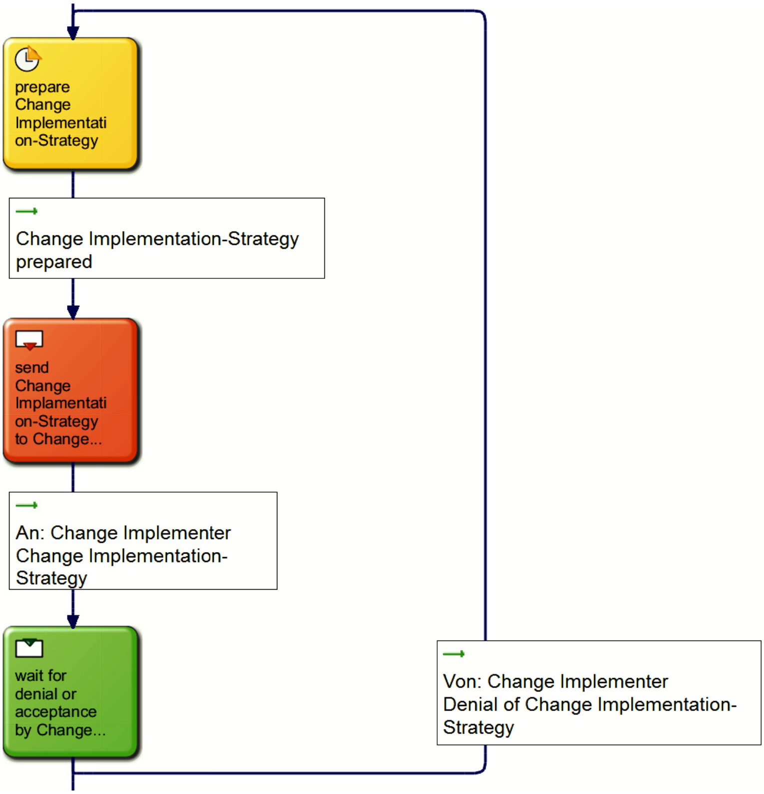


Fig. 5. Change Coordinator receives and prepares the Change Implementation Strategy

to make amendments to the Change Strategy. For this, the S-BPM model includes a loop to jump back to the Function state *prepare Change Implementation Strategy* in the case of rejection. Until the CI accepts the Change Strategy, the CC cannot continue the process.

The subject behavior model for both subjects is determined by these loops. There are always transitions between the CC and the CI, which may be positive or negative in character. Such coordination procedures can demand a great deal of communication in practice and so should be validated and adapted in the operational context.

3.4 Validating the reference model

We will now present a functionality within the S-BPM modeling tool Metasonic Suite, which can be used to execute the reference model for validation purposes. Instantiating a Change Management process allows the activity flow to be tracked, tested and controlled with the aid of a web-based interface (Fig. 7).

On the right of Figure 7 we can see which user is logged in. Beneath the menu bar is a navigation area showing from left to right the processes in which the user is active, and as which subject. At the moment *Patrick Garon* is logged on, and is active in

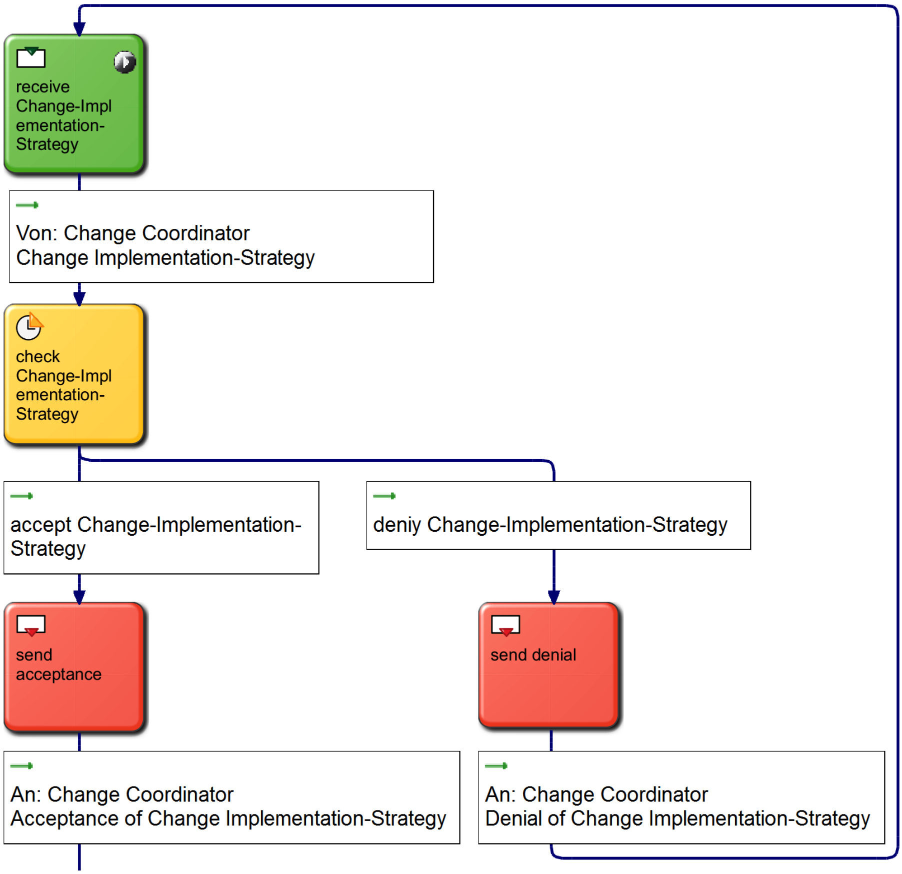


Fig. 6. Input and commitment to the change strategy by the Change Implementer

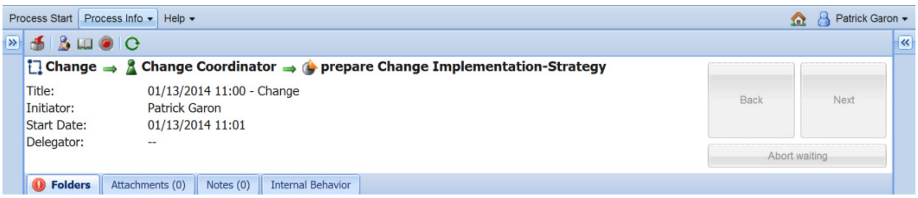


Fig. 7. Change Coordinator: "prepare Change Implementation Strategy"

the *Change Process* as *Change Coordinator* in the function *prepare Change Implementation Strategy*.

Once the CC has delivered his part of the Change Implementation Strategy, he clicks on the Next button to trigger the Send state *send Change Implementation Strategy to CI* and sends this to the Change Implementer. He then passes into the Receive state *wait for denial or acceptance by CI* (see Fig. 8). The process is now taken forward by the Change Implementer. Fig. 9 shows the Change Implementer that a Change Management process has been instantiated, so a message has arrived.

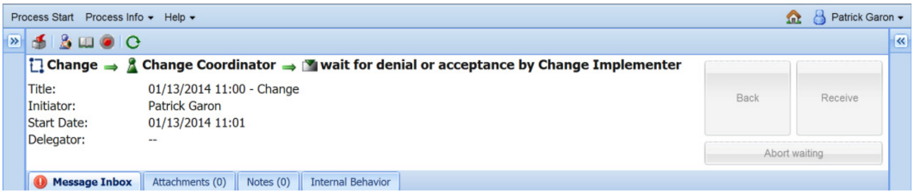


Fig. 8. Change Coordinator: “wait for denial or acceptance by CI”

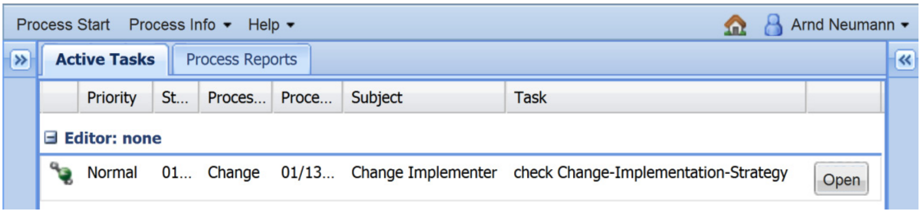


Fig. 9. Change Implementer: “receive Change Implementation Strategy”

After opening the message, the CI switches from the Receive state *receive Change Implementation Strategy* to the Function state *check Change Implementation Strategy* (Figure 10 shows the same basic view as Figure 7).

To continue the process, the *Next* button is pressed once more. A dialog box then opens, asking the CI for a decision. As defined previously in the process model and shown in Figure 11, he can choose between acceptance and rejection. Whichever way the decision goes, a message with the relevant content will pass back to the CC in both cases.

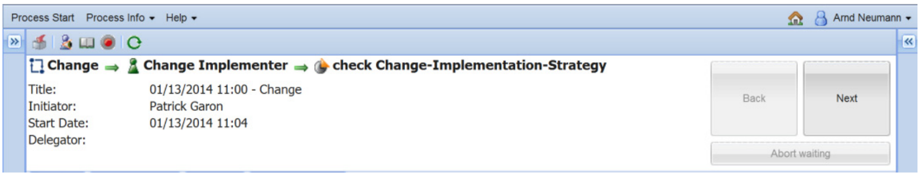


Fig. 10. Change Implementer: “check Change Implementation Strategy”

In this case the CI has accepted the Change Implementation Strategy and triggered the corresponding Send state. The system registers this and, in response to the message sent by the CI, the CC switches from the Receive state *wait for denial or acceptance by CI* to the Function state *prepare Back-Out Strategy*, as this represents the next step in the process. In the case of rejection by the CI, the CC switches from the Receive state *wait for denial or acceptance by CI* back to the Function state *prepare Change Implementation Strategy* and the process step described above starts over, until the Change Implementation Strategy is finally accepted.

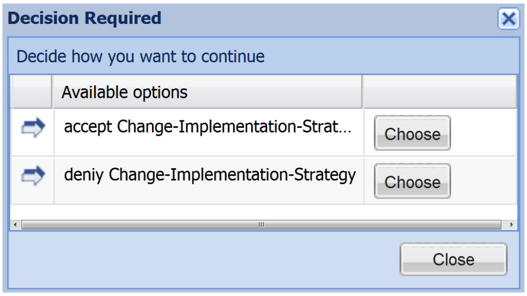


Fig. 11. Change Implementer: decision on the Change Implementation Strategy

3.5 Data Layer Design for Change Management

In order for a Change to be implemented and the Configuration Management Database (CMDB) to be brought up to date on completion of the Change, adequate documentation is required. The ITIL defines the change process for this as a form-based process. The form should include following items (see [15]):

- the objective and justification for the Change,
- the date, the status, the priority, the criticality and the expected effect of the Change,
- the Configuration Items (CIs) affected, any possible downtime, a Back-Out Strategy, and necessary activities before, during and after the change,
- the resources including the necessary technical staff, and a staff deployment plan.

To enable the CMDB to be updated to reflect the actual position, these details must be complete, valid, in line with their definitions, consistent, and as atomic as possible. It must also be possible to keep a history of the individual Changes for future management decisions, which is why high-quality data is essential [10].

In practice, however, the data organization continues to cause substantial quality problems. Gartner suggests that up to 25 percent of the data in the Fortune 1000 companies is incorrect and incomplete, and that 80 percent of the data in these companies is unstructured and decentralized (see [17], p. 317). The documents relating to Change Management likely are no exception.

We will now take an example case to show how the reference model is expanded to include a data layer. The data layer is represented by the business objects which, once they have been configured, are integrated directly into the SBD models with subject-specific views. The views indicate which attributes are relevant and whether they can only be read or also updated. In the example, the Change Coordinator has to fill in the view shown in Figure 12 to prepare the Change Implementation Strategy. Along with text fields, business objects can also be defined as lists, “forcing” the user into a defined selection from a case register. For the Change type, for example, the CC can only choose between major, significant and minor. Once the CC has saved his input, he can send it to the CI.

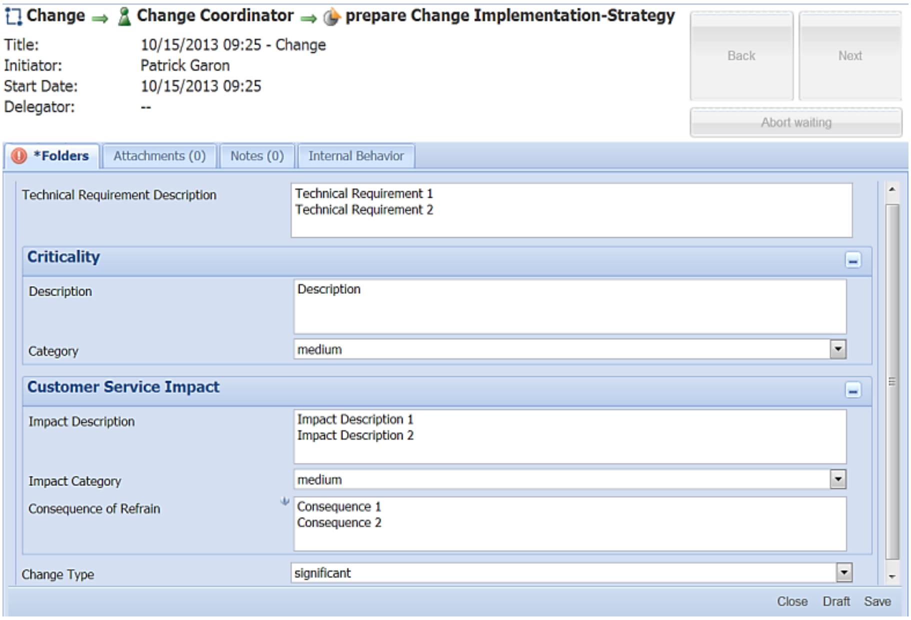


Fig. 12. Data view for the Change Coordinator to prepare the Change Implementation Strategy

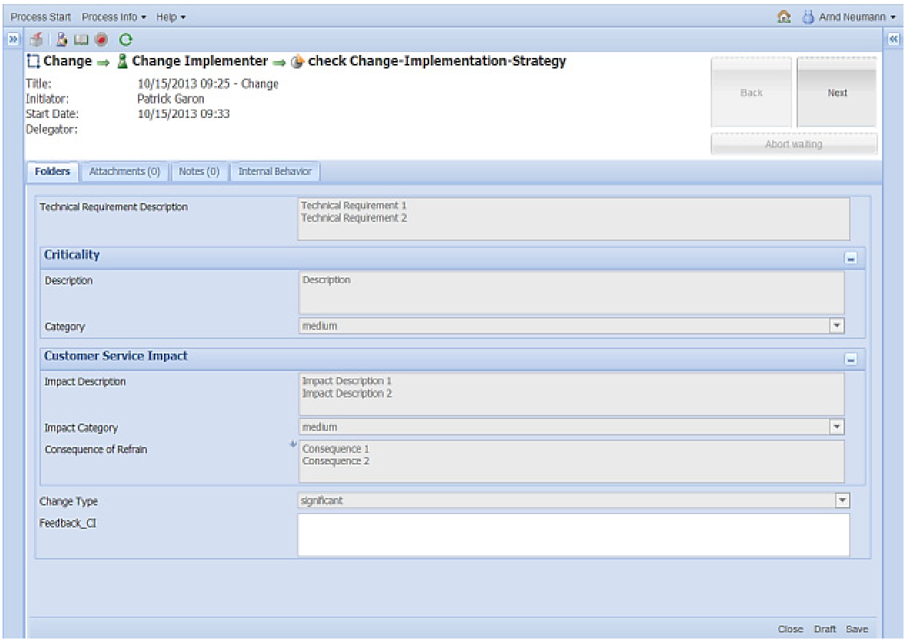


Fig. 13. Data view for the CI to display the Change Implementation Strategy prepared by the CC

After opening the message, the CI sees the view depicted in Figure 13. He cannot change the data provided as in our case since this is not covered by his access rights, which is why the details entered by the CC are grayed out. He can however give feedback on the Change Implementation Strategy and reject or confirm this in the next step. If he confirms the Change, feedback is not mandatory, but in the case of rejection it is essential. After the CI has entered his feedback, he saves his input and sends it back to the CC, who then makes appropriate changes and returns the data to the CI.

As the example shows, a subject-oriented approach to Change Management can provide end-to-end documentation of the Change using a managed data store containing a set of relevant business objects. The business objects hold all process-related data, such that a coherent basis for operational decision making is established.

4 Conclusion

The ITIL is a de-facto standard and should be implemented in practice in a process-oriented way. In the basic form of the ITIL, however, there are evident gaps as – to the best of our knowledge – no complete and adequately detailed process descriptions are available as reference models. The subject-oriented reference model for the Change Management process presented here provides a design template which can be reused as an application model with the aid of the S-BPM mechanisms and tools. Institutions can select and adapt components of the model. Because the S-BPM models are executable (model-to-execute), the results of the modification can be validated directly, providing a high degree of agility in process definition and rollout.

The proposed model can be used by managers, analysts and consultants in the domain of IT service management for different purposes:

1. The model artifacts could be used as blueprint in order to (re-)organize the domain of Change Management in IT service centers. In particular, the designed models provide relevant actors and corresponding behavioral patterns [23] in order to design a modern, process-oriented organization.
2. The reference model could serve as a guideline to evaluate and enhance software solutions to support the Change Management process with regard to the application and data layer. In practice, application support for ITIL processes is predominantly realized by use of office tools like spreadsheet calculation software which induces severe data quality problems [3]. These problems can be tackled by using the presented reference model with the associated data layer and its business objects. As a consequence, the degree of maturity of the IT service organization is expected to increase.
3. In addition, the model can also be used as teaching tool in industry and university. Since the model is executable, the complex procedures of communication and coordination in the domain of Change Management can demonstrated close to reality. Beyond that, process trainings [2] based on S-BPM can enable students to walk through ITIL-based processes by use of a computer supported, cooperative web based interface, such that learning effectivity could be positively stimulated.

Future research has to concentrate on the practical evaluation [8] and refinement of the presented model, for example by carrying out case studies and applicability evaluations in IT service organizations.

References

1. Ardalani, P., Houy, C., Fettke, P., Loos, P.: Towards A Minimal Cost Of Change Approach For Inductive Reference Model Development. In: Proceedings of the 21st European Conference on Information Systems (ECIS), Utrecht, Netherlands (2013)
2. Back, G., Daniel, K.: Process Training to Support Change Necessary within Scope of Process Implementation. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 48–61. Springer, Heidelberg (2011)
3. Baskarada, S.: How Spreadsheet Applications Affect Information Quality. Journal of Computer Information Systems 51, 77–84 (2011)
4. Fettke, P., Loos, P.: Perspectives on Reference Modeling. In: Fettke, P., Loos, P. (eds.) Reference Modeling for Business Systems Analysis, pp. 1–20. IGI Global, Hershey (2007)
5. Fleischmann, A., Schmidt, W., Gilbert, O.: Subjektorientiertes Geschäftsprozessmanagement. HMD – Praxis der Wirtschaftsinformatik 266, 52–62 (2009)
6. Fleischmann, A., Schmidt, W., Stary, C., Augl, M.: Agiles Prozessmanagement mittels Subjektorientierung. HMD – Praxis der Wirtschaftsinformatik 290, 64–76 (2013)
7. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management. Springer, Heidelberg (2012)
8. Frank, U.: Evaluation of Reference Models. In: Fettke, P., Loos, P. (eds.) Reference Modeling for Business Systems Analysis, pp. 118–140. IGI Global, Hershey (2007)
9. Goeken, M., Alter, S., Milicevic, D., Patas, J.: Metamodelle von Referenzmodellen am Beispiel ITIL – Vorgehen, Nutzen, Anwendung. In: Fischer, S., Maehle, E., Reischuk, R. (eds.) Informatik 2009: Im Focus das Leben, Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. LNI, vol. 154, pp. 3701–3714. GI (2009)
10. Grob, H.L., Reepmeyer, J.-A., Bensberg, F.: Einführung in die Wirtschaftsinformatik. Vahlen, München (2004)
11. Hamilton, P.: Dynaxity – Management von Dynamik und Komplexität im Softwarebau. Springer, Berlin (2007)
12. Hammer, M.: What is Business Process Management? In: van Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1, pp. 3–16. Springer, Heidelberg (2010)
13. Hintsch, J., Turowski, K.: Towards Implementing IT Service Management in an ERP for the IT Service Industry. In: Grabis, J., Kirikova, M., Zdravkovic, J., Stirna, J. (eds.) Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013), pp. 83–94. Riga, Latvia (2013)
14. Irani, Z., Ghoneim, A., Love, P.: Evaluating Cost Taxonomies for Information Systems Management. European Journal of Operational Research 173, 1103–1122 (2006)
15. Köhler, P.: ITIL – Das IT-Servicemanagement Framework. Springer, Heidelberg (2007)
16. Kubernus, R.: Innovatives Geschäftsprozessmanagement durch Subjektorientierung: S-BPM ermöglicht ein durchgängiges Round-Trip-Engineering in Echtzeit. Diplomica, Hamburg (2013)
17. Laudon, K.C., Laudon, J.P., Schoder, D.: Wirtschaftsinformatik – Eine Einführung. Pearson, München (2010)

18. Lienemann, G.: ITIL – Change Management: Hinweise und Vorgehensweisen aus der Praxis. Heise, Hannover (2006)
19. Metasonic. Metasonic site, http://www.metasonic.de/
20. mITSM. Change Management, http://www.mitsm.de/itil-wiki/prozess-beschreibungen-deutsch/change-management
21. Schermann, M., Böhmann, T., Krcmar, H.: Explicating Design Theories with Conceptual Models: Towards a Theoretical Role of Reference Models. In: Becker, J., Krcmar, H., Niehaves, B. (eds.) Wissenschaftstheorie und Gestaltungsorientierte Wirtschaftsinformatik, pp. 175–194. Physica, Heidelberg (2009)
22. Schultz, M.: Enriching Process Models for Business Process Compliance Checking in ERP Environments. In: vom Brocke, J., Hekkala, R., Ram, S., Rossi, M. (eds.) DESRIST 2013. LNCS, vol. 7939, pp. 120–135. Springer, Heidelberg (2013)
23. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action Patterns in Business Process Models. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 115–129. Springer, Heidelberg (2009)
24. Zarnekow, R.: Produktionsmanagement von IT-Dienstleistungen: Grundlagen, Aufgaben und Prozesse. Springer, Heidelberg (2007)
25. Zarnekow, R., Hochstein, A., Brenner, W.: Service-orientiertes IT-Management: ITIL-Best-Practices und Fallstudien. Springer, Berlin (2005)

Cross-Organizational and Context-Sensitive Modeling of Organizational Dependencies in $\mathcal{C} - \mathcal{ORG}$

Alexander Lawall, Thomas Schaller, and Dominik Reichelt

Institute for Information Systems at Hof University, Alfons-Goppel-Platz 1, 95028 Hof, Germany
{alexander.lawall,thomas.schaller,dominik.reichelt}@iisys.de

Abstract. Almost every application used to run a business relies on a model of the organization structure, the roles and the actors in order to define access rights or assign tasks. This article proposes a novel approach to organizational modeling. It also describes a way to connect internal and external organizational models in order to implement cross-organizational processes. It demonstrates the approach on two examples. One of them is a cross-organizational business process and the other a joint research project. The paper includes a description of the metamodel that constitutes the approach for context-sensitive modeling. It shows concrete language expressions that describe sets of actors and how these expressions are interpreted on the organizational model. The article concludes with a short overview of a prototypical implementation of the system $\mathcal{C} - \mathcal{ORG}$.

Keywords: cross-organizational, inter-organizational, context-specific, metamodel, organizational model, formal language.

1 Introduction

For years, companies have been facing an increasing complexity of their environment. In order to be viable (cf. [1]), their organization structure has to be flexible enough to react to those changes (cf. [2]). Otherwise they will be eliminated from the market. So it is no big surprise that a lot of companies changed their structure from traditional stereotypes like hierarchies or tensor to more flexible concepts like process or virtual organization forms (cf. [3, p. 277]). Nowadays the organizational structure is driven by the work in project teams, global teams, networks and global teams in networks (cf. [4]). Especially cooperative structures of independent partners like virtual organizations[1] are arising (cf. [4]). They bundle the core competencies of all involved companies over spacial distances [3, p. 278].

[1] A virtual organization has the potential of a traditional organization without having a comparable institutional frame. They go beyond intra- and inter-organizational limitations, cf. [5].

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 89–109, 2014.

Traditional structures for organizing a company like hierarchies, matrices, etc. are partly disused, but still exist in companies (cf. [6]), though new organization paradigms are arising. In addition to these intra-organizational aspects, cooperation paradigms like supply chain management or joined product development make it necessary to pay special attention to inter-organizational (same as cross-organizational) processes and structures ([7, pp. 4]).

Almost every application used to run a business relies on a model of the organization structure, the roles and the actors in order to define access rights or assign tasks. This article proposes a novel approach to organizational modeling. It focuses on representing cross-organizational interactions and context-specific organizational relations.

In order to consistently demonstrate key issues, section 2 introduces a practical cross-organizational business process. We consider this process from the perspectives of the different involved parties, but with a strong emphasis on cross-organizational communication.

Chapter 3 then introduces extensions to an existing approach described in [8]. It proposes new entity types and sets of relationship types for describing cooperations. After that the foundations of the metamodel and a corresponding query language is presented.

The following chapter illustrates the proposed concepts by applying them to the example described in section 2. It also shows how temporary cooperations, e.g. projects, can be represented within the frame of the metamodel.

In order to answer questions to concrete organizational models (e.g. the definition of actors in workflow management systems), different algorithms are used to traverse the model. Section 5 describes these algorithms, especially with regard to the novel concepts introduced in section 3.

We conclude with the presentation of a prototypical implementation (section 6) and by giving an outlook on topics for future research (section 7).

The appendix contains a more detailed and limited view on aspects discussed in section 2.

2 Motivating Example

This section describes a cross-organizational business process – a purchase. First, we consider the process stakeholders from an abstract perspective. In the following, we focus on the different internal *subjects*, i.e. behaviors as shown by the process stakeholders (cf. [9]). For clarity purposes, we omit parts of the individual subjects' behavior and concentrate on cross-organizational interactions.

Figure 1 depicts the interaction of the involved subjects on a high level of abstraction.

We now consider the subjects described in fig. 1 in more detail. Figures 8 (cf. appendix A) and 2a are representations of internal subjects within the *customer* subject.

The start subject of the whole process is depicted in figure 8. An employee of the university (the process *initiator*) decides that they need to purchase an

Fig. 1. Information flow between subjects

article. They fill out a request for purchase, which is then reviewed by their *supervisor*. If the supervisor approves the request, the initiator forwards it to the *purchase department*. The employees of this department handle the actual purchase order process, and after a while, the process initiator receives the package and a delivery receipt from the *distributor*. They then validate the receipt and the delivery. They also forward the receipt to the purchase department as signal to finalize the order process (omitted in the figure).

The following subjects are of relevancy:

- Initiator
- Supervisor
- Purchase Department
- Distributor

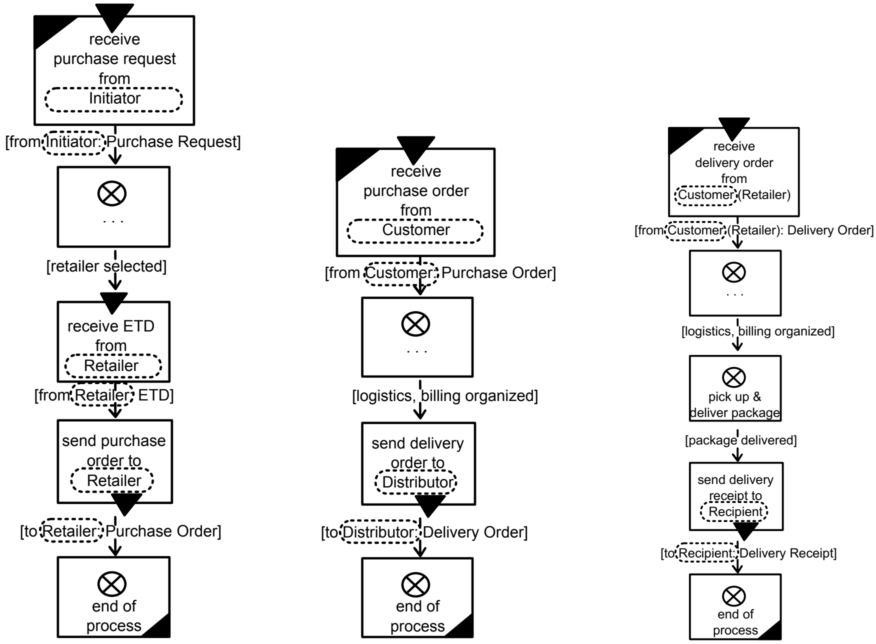
Fig. 2a shows the actions that need to be taken by the subject *purchase department*, once they receive a purchase request from the process *initiator*. In the depiction, we focus on interactions with other subjects that are described in our context. Missing steps, indicated by "...", include the actual comparison of retailers and internal accounting affairs. A purchase might be time-critical, so one criterion for the selection of a retailer can be the request for an estimated time of delivery (ETD). This ETD is received by the *purchase department*'s clerk.

After selecting a *retailer*, the clerk sends the concrete purchase order to them. This concludes this limited view of the purchase department's role in the overall process. From this perspective, only the Retailer is a relevant new subject.

At this point, the subjects contained in the *customer* subject in figure 1 are described adequately. We continue by describing the purchase processing enacted by the *retailer* (cf. fig. 2b).

The reception of a purchase order from the *customer* triggers the actual handling of the purchase. As previously, we focus on cross-organizational interactions. Omitted process steps include organizing the logistics, billing and actual handling of the ordered products. In the final step of this process, the delivery order is sent to the *distributor*, indicating that the package is ready to be delivered to the customer.

Figure 2c picks up at this point from the distributor's perspective. Please keep in mind that here, the subject *customer* does not denote the same subject as in fig. 1. This is a result of the shifted perspective. The *customer* of the distributor is actually the *retailer* that sends them the delivery order. After receiving the delivery order from their *customer*, the distributor organizes their



(a) Customer: purchase department

(b) Retailer: purchase processing

(c) Distributor: delivery order processing

Fig. 2. Internal subjects of the involved organizations

billing and logistics for the delivery. Again, these internal steps are omitted in the depiction. At some point, an employee actually picks up the package from the sender and delivers it. Additionally, they hand the delivery receipt to the *recipient*. As we omit internal steps like handling the recipient countersigning the delivery, the process is at an end here.

The following additional subjects are of relevancy:

– Customer: Actually the *retailer* in the cross-organizational view (fig. 1)
– Recipient: The recipient of the delivery, actually the *customer* in fig. 1

3 Metamodel

This section describes the metamodel for cross-organizational models. It includes entity and relationship types to model organizational requirements. Additionally, the relationship types can be restricted by different kinds of constraints.

The metamodel for modeling organizational requirements consists of sets of entity type sets $\mathcal{V} = \{\mathcal{V}_{internal}, \mathcal{V}_{external}, \mathcal{V}_{cooperation}\}$ and sets of relationships types $\mathcal{R} = \{\mathcal{R}_s, \mathcal{R}_o, \mathcal{R}_u\}$. The formal specification of the basic metamodel is described in [8].

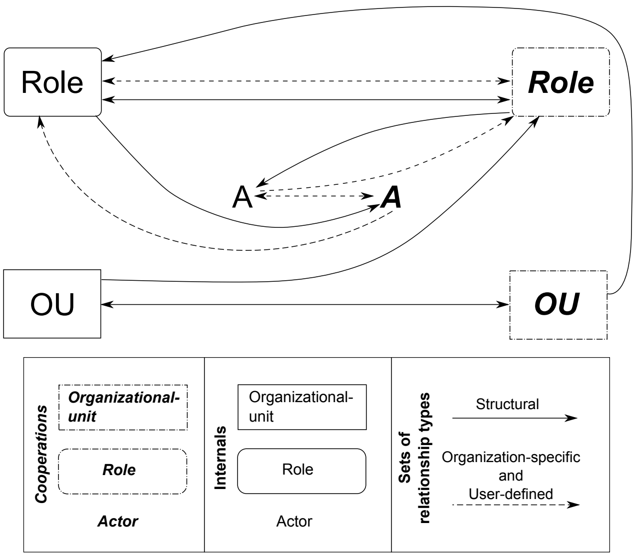


Fig. 3. Excerpt from the metamodel focussing on cooperations and internals

The focus of this paper is the cross-organizational (equals to inter-organizational) context. Cross-organizational means that more than one organization form a cooperation. There is a difference between time-persistent and time-limited (temporary) cooperations. In the example shown in figure 1, the cooperation is a time-persistent cooperation. The *customer*, *retailer* and *distributor* are involved in the process for each concrete purchase.

These requirements have to be met by the metamodel. It has to support modeling both permanent and temporary federations in order to allow interconnecting organizations.

Entity- and Relationship-Types

All entity type sets of $\mathcal{V}$ include the entity types organizational-units $\mathcal{OU}$, roles $\mathcal{ROLE}$ and actors $\mathcal{ACTOR}$. The entity types of $\mathcal{V}_{internal}$ represent the elements of the internal organizational model (cf. [8] and [10]). Entity types that denote externals $\mathcal{V}_{external}$ and cooperations $\mathcal{V}_{cooperation}$ are an extension to the set of entity sets $\mathcal{V}$. The set $\mathcal{V}_{cooperation}$ includes all entity types used to specify cooperations (time-persistent and time-limited). Externals $\mathcal{V}_{external}$ are used to distinguish between internal $\mathcal{V}_{internal}$ (based on the "own" organizational model) and external $\mathcal{V}_{external}$ (all involved organizations, except the "own" one) organizational model entity types. External entity types are needed to interconnect (over cooperation entities) external with internal entities.

The relations between internal and external entities have to be constrained context-specific. This means that they are only valid in a given situation. A more detailed explanation of this concept is given in section 4.

The metamodel consists, besides of entities, of the set $\mathcal{R}$ of relationship type sets. This set can be broken down into *structural* $\mathcal{R}_s$, *organization-specific* $\mathcal{R}_o$ and *user-defined* $\mathcal{R}_u$ relationship type sets.

The *structural* set of a relationship type

$$\mathcal{R}_s = \{\mathcal{HAS}\}$$

acts as an "IS PART OF" relation and is used to relate recursive organizational-units, organizational-units to roles and roles to actors.

The *organization-specific* set of relationship types

$$\mathcal{R}_o = \{\mathcal{HAS\_DEPUTY}, \mathcal{HAS\_SUPERVISOR}, \mathcal{REPORTS\_TO}\}$$

are partitioned in:

- $\mathcal{HAS\_DEPUTY}$ to model deputyship between entities,
- $\mathcal{HAS\_SUPERVISOR}$ to represent the supervisor relationship, and
- $\mathcal{REPORTS\_TO}$ relations that are used to specify the duty of reporting.

All these *organization-specific* relationship types connect role to role, actor to actor or actor to role. The same rules apply between external and between cooperation entities.

The *user-defined* set of relationship types $\mathcal{R}_u$ is freely definable. Relationship types that are needed can be defined and are included in the traversal process. For example, we can define a relationship type $\mathcal{HAS\_DRIVER}$. When a query looking for drivers of an actor is interpreted on the organizational model, the corresponding set of drivers results from evaluating the $\mathcal{HAS\_DRIVER}$ relations. The *user-defined* set of relationship types $\mathcal{R}_u$ allows extending the relationship types as needed.

The aforementioned sets of relationship types $(\mathcal{R}_s, \mathcal{R}_o, \mathcal{R}_u)$ can be used to interconnect the different sets of entity types, such as entity types of internals $\mathcal{V}_{internal}$, externals $\mathcal{V}_{external}$ and cooperations $\mathcal{V}_{cooperation}$. The rules defined above are the same for this interconnection. Figure 3 shows these rules in a graph-based manner. For clarity, the figure only shows rules applicable between different sets of entity types, i.e. cooperations $\mathcal{V}_{cooperation}$ and internals $\mathcal{V}_{internal}$. It omits rules that apply within the same entity type (i.e. between internals). All permutations between the three sets (internals, externals and cooperations) are subject to these rules.

Constraints on Relationship Types

The relations of the two sets of relationship types $\mathcal{R}_o$ and $\mathcal{R}_u$ can be restricted with a set of constraints $\mathcal{C}$[2].

$$\forall \Gamma_{r_1,r_2}(\Gamma_{r_1,r_2} \in \mathcal{R}_o \vee \Gamma_{r_1,r_2} \in \mathcal{R}_u) : \Gamma_{r_1,r_2} \subseteq (\mathcal{ROLE} \times \mathcal{ROLE}) \times \mathcal{C} \quad (1)$$
$$\forall \Gamma_{a_1,a_2}(\Gamma_{a_1,a_2} \in \mathcal{R}_o \vee \Gamma_{a_1,a_2} \in \mathcal{R}_u) : \Gamma_{a_1,a_2} \subseteq (\mathcal{ACTOR} \times \mathcal{ACTOR}) \times \mathcal{C} \quad (2)$$
$$\forall \Gamma_{a,r}(\Gamma_{a,r} \in \mathcal{R}_o \vee \Gamma_{a,r} \in \mathcal{R}_u) : \Gamma_{a,r} \subseteq (\mathcal{ACTOR} \times \mathcal{ROLE}) \times \mathcal{C} \quad (3)$$
$$with\ a, a_1, a_2 \in \mathcal{ACTOR}, r, r_1, r_2 \in \mathcal{ROLE}$$

[2] The set of relationship types $\mathcal{R}_s$ can not be restricted in this fashion.

The set $\mathcal{C}$ includes the empty symbol ε to make constraints optional[3]. The constraints are possible on relations between roles (1), between actors (2) and between actor and role (3). Examples of concrete constraints are shown in figures 4 and 5. Constraints are assigned to relations and reduce the solution space when traversing relations. Traversal takes place when evaluating a query (language expression). The constraints $\mathcal{C}$ can be distinguished as follows:

- *Context*-based: If the context of the query is equal to the context on the relation, the traversal follows the relation.
- *Attribute*-based: If the attribute of a concrete entity fulfills the constraint (predicate) on the relation, the entity is retained in the result set. Otherwise it is removed. For the detailed algorithm see section 5.
- *Parameter*-based: If the parameter of the query fulfills the constraint (predicate) on the relation, the traversal follows the relation.

Language Expression for Constraints

The constraint $c \in \mathcal{C}$ on a relation can be formulated as

$$[< context > [.]][ATT. < attribute >< operator >< value >] \qquad (4)$$

$$[< context > [.]][< parameter >< operator >< value >] \qquad (5)$$

Context is an optional term that can be combined with attribute and parameter based constraints. This means that context-specific attribute / parameter constraints can only evaluate positively if the context is correct. "*ATT*" is a special terminal symbol to distinguish between attribute and parameter based evaluation (cf. language expression (4)). "*ATT.*" is mandatory for defining a predicate based on attributes. An example for attribute-based constraints is the language expression "$ATT.HiringYear > 2$". It is also possible to assign only the context to the relation, depicted in figure 5. This context-specific constraint is independent of attributes and parameters.

The main difference between constraints based on attributes and parameters is that the values of the entities' attributes are stored in the organizational model. The attribute based constraints are evaluated purely based on model-internal information. Parameters, in contrast, are passed from outside the organizational model and evaluated on the predicates on the relations. The syntax of the language expression for constraints based on parameters is shown in (5). The external parameter is formulated as part of the WITH-clause, described in [11, Figure 4].

Role-Dependent Relationship Type

The role-dependent relationship type is used to "constrain" a relationship type of the sets $\mathcal{R}_o$ and $\mathcal{R}_u$.

$$\forall \Psi_{a_1,a_2}(\Psi_{a_1,a_2} \in \mathcal{R}_o \vee \Psi_{a_1,a_2} \in \mathcal{R}_u) : \Psi_{a_1,a_2} \subseteq (\Gamma_{a_1,a_2}) \times \mathcal{ROLE} \qquad (6)$$

$$\forall \Psi_{a,r}(\Psi_{a,r} \in \mathcal{R}_o \vee \Psi_{a,r} \in \mathcal{R}_u) : \Psi_{a,r} \subseteq (\Gamma_{a,r}) \times \mathcal{ROLE} \qquad (7)$$

[3] Concrete relations in the model without constraints are generally valid.

A "basic" relation $\gamma \in (\Gamma_{a_1,a_2} \cup \Gamma_{a,r})$ is only active if the source is an actor that acts in the role r assigned to γ. Functions (8) and (9) specify the assignment from γ to $\psi \in (\Psi_{a_1,a_2} \cup \Psi_{a,r})$. Formulas (10) and (11) specify the assignment from ψ to r.

$$f_{\Gamma_{a_1,a_2}} : \Gamma_{a_1,a_2} \rightarrow \Psi_{a_1,a_2} \tag{8}$$
$$f_{\Gamma_{a,r}} : \Gamma_{a,r} \rightarrow \Psi_{a,r} \tag{9}$$

$$f_{\Psi_{a_1,a_2}} : \Psi_{a_1,a_2} \rightarrow \mathcal{ROLE} \tag{10}$$
$$f_{\Psi_{a,r}} : \Psi_{a,r} \rightarrow \mathcal{ROLE} \tag{11}$$

This is shown in figure 4 as "basic" relation between ARB and P and the role-dependent relation between this relation and the role *Lecturer*. Section 4 demonstrates role-dependent traversal by example.

Language Expression for Role-Dependent Traversal

In the previous section, we describe relations that are only valid if an actor assumes a given role r. There are two ways to specify which role an actor assumes in a given query:

1. Explicit definition within the query: Using the "AS" terminal symbol, such as "ARB" AS $Lecturer$, actors can be assigned to roles. In the example, ARB acts as *Lecturer*. This makes the relation between ARB and P active.
2. Implicit definition: If actors are declared using an expression of the form $Researcher(Research\ Group\ A)$, the role they assume is implicitly contained in the query. In the example, the resulting actors ARA and ARB act as *Researcher*s. The relation between ARB and P is inactive.

4 Cooperation and Context

This section shows the concepts of section 3 by examples. These examples refer to figures illustrating realistic scenarios.

4.1 Purchase Example

The following considerations discuss the cross-organizational purchase process described in 2. In order to illustrate concepts from section 3, we revisit the subjects that receive messages in the process. We show by example how they can be declared using language expressions. We then proceed to retrace the concrete lookup of actors belonging to these subjects. Base for these considerations is the organizational model depicted in figure 4.

The figure shows the organizational model of the three organizations involved in the purchase as seen by the *customer* located within the "University". The

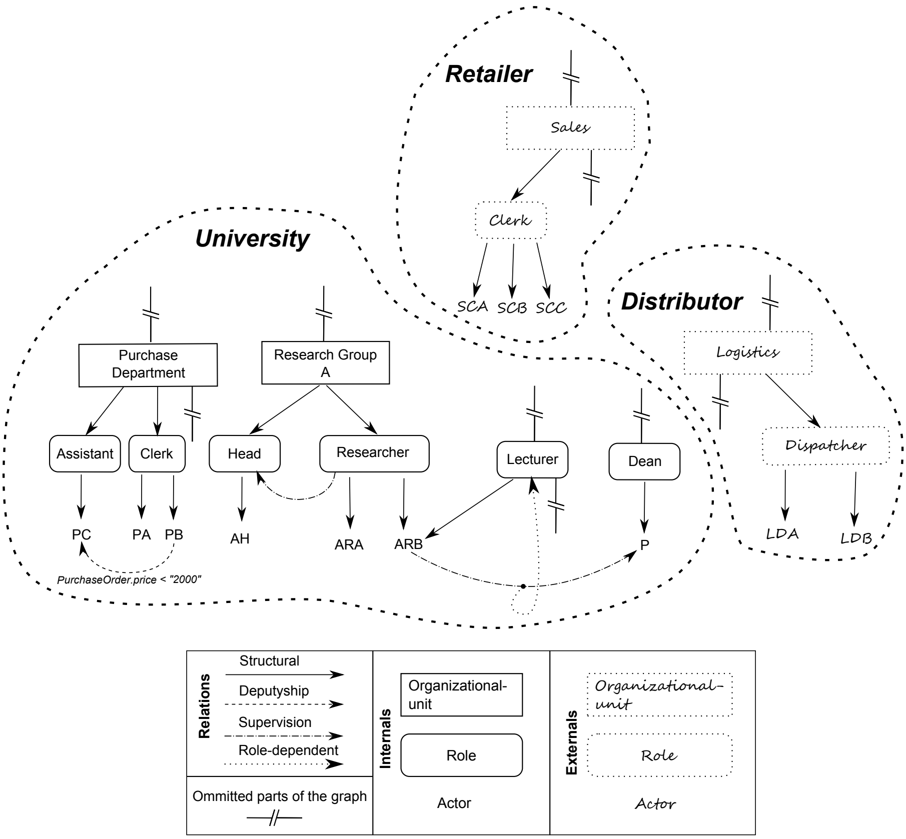


Fig. 4. Organizational model from the customer's perspective

entities of the external organizations "Retailer" and "Distributor" are denoted as *externals*.

The models of the external organizations are limited to actors that interact with the "University".

– For the "Retailer", three concrete actors are modeled: *SCA*, *SCB* and *SCC*. All of them fulfill the same role, *Clerk*, within the same organizational-unit, *Sales*.
– On the "Distributor" side, two actors are defined: *LDA* and *LDB*. They both are *Dispatcher*s within the same organizational-unit, *Logistics*.

We represent the internal organization of the "University" as a subgraph of the complete model. The subgraph is limited to entities relevant to the example. It consists of the organizational-units *Purchase Department* and *Research Group A*. Within the *Purchase Department*, *PA*, *PB* are *Clerk*s and *PC* is *Assistant*. Within *Research Group A*, *AH* is *Head* of the group, *ARA* and *ARB* are *Researcher*s. Additionally, the "University" contains the roles *Lecturer* and *Dean*.

P is *Dean*, *ARB* is *Lecturer* in addition to his role as *Researcher*. This concludes the structural composition of the "University". The entities are connected by structural relations.

The organization-specific relations used in the model of the "University" are:

1. the supervision from the role *Researcher* to *Head*
2. the deputyship from the actor *PB* to *PC*
3. the supervision from *ARB* to *P*

(1.) is a generally valid relation. The deputyship relation (2.) is constrained. It is only valid, if the predicate *price is lower than 2000* — in the context *PurchaseOrder* — is true. The supervision relation (3.) is role-dependent. It is only valid, if *ARB* acts as a *Lecturer*.

Definition of Actors

The problem of resolving actors to subjects arises only in the *send* state (cf. [11]). A subject enters *send* state. The set of possibly receiving actors (the subject) is determined by evaluating the language expression. All the actors contained in the set can decide to *receive* the message. The subjects addressed in the *send* state in figures 2a, 2b, 2c and 8 are assigned to actors by language expressions (queries). These define actors that are responsible for a task. In section 2, we found that the following subjects exchange messages:

- *Customer* perspective
  1. **Initiator** denotes the person that started the process. This is a concrete actor, for the example we assume that *ARB* initiated the process. Consequently, the language expression for this subject is the literal "*ARB*" that resolves to the concrete actor *ARB*.
  2. **Supervisor** refers to the supervisor or the initiator. This is the first occurrence of a language expression where a lookup has to occur. An expression to formalize this actor description is $SUPERVISOR(\text{“}ARB\text{”})$. The literal "*ARB*" remains from step 1, and we want to find their *SUPERVISOR*. Running this query on the organizational model returns *AH*, who is the general supervisor of the researchers in research group A.
  3. **Purchase Department** is the subject that handles purchase requests. In the example model, this means all *Clerk*s of the *Purchase Department*. For failover purposes, the business process management system can also supply additional information to the model[4]. An expression for this subject is $Clerk(Purchase\ Department)$ $WITH\ price = \text{“1500”}\ AND\ context = \text{“PurchaseOrder”}$. The *price* parameter can be used from the concrete process instance, while the *context* "PurchaseOrder" can be a static property of the process template. Running this more complex query on the organizational model returns *PA*

[4] In this case a parameter and a context, cf. section 3.

and *PB* if they are available[5]. If both these actors are unavailable, however, the organizational model could use the additional information to determine *PC* as replacement actor[6]. For further details on the traversal procedure, see section 5.

4. **Retailer** denotes an external subject — an employee of the retailer that can process purchase requests. An expression for this group of actors from the organizational perspective is $Clerk(Sales)$, which denotes any *Clerk* within a *Sales* organizational-unit. Evaluating this query returns the three external actors *SCA*, *SCB* and *SCC*.
5. **Distributor**, as seen from the customer perspective, is not a subject to send to. Consequently there is no need to specify a language expression for this subject.

– *Retailer* perspective
  1. **Customer**, in the simplified version of the process described in section 2, is also not a subject to send to. In a more detailed process, however, the customer would at least be sent a confirmation of the order. The subject would also be represented as a literal with a value extracted from the received order. This is similar to the **Initiator** example in the customer's perspective.
  2. **Distributor**, from the retailer's perspective, is an external subject that can execute delivery orders. An expression that describes the relevant actors in the example organizational model is $Dispatcher(*)$. It addresses any actors fulfilling the *Dispatcher* role, independent of the organizational-unit they are part of. The result set for this query consists of the actors *LDA* and *LDB*.

– *Distributor* perspective
  1. **Customer** is a subject that acts similar to the **Customer** subject in the retailer's perspective. In the simplified version of the process, it is not sent any messages.
  2. **Recipient** is the subject that receives the package and the delivery receipt. For the example, it is represented by the literal "*ARB*". This means that the initiator of the process is carried as a process variable all the way through the external organizations.

So what happens if there exist internal and external entities with the same name? A conflict arises, when a role within an organizational-unit exists in both the internal and external organizations. Imagine an additional internal *Sales* organizational-unit, staffed with *Clerk*s, within the "University" depicted in fig. 4. Then the language expression $Clerk(Sales)$ resolves to all — external, internal and cooperation — actors that fulfill the role *Clerk* within a *Sales* organizational-unit.

This may not always be the desired behavior. We do not want to order supplies from our own sales team. So we need a way to address only external sales clerks.

[5] The simpler expression $Clerk(Purchase\ Department)$ would yield the same result.

[6] This would not be possible with the simple expression.

This can be done by using complex queries as described in [11]. We can subtract the set of internal actors from the result set returned by $Clerk(Sales)$. If we assume an organizational unit *University* (omitted in the figures), we can modify the language expression as follows: $Clerk(Sales)$ NOT $Clerk(University)$.

The same problem not only applies to internal and external organizational structures, but to cooperation structures as well.

4.2 Temporary Cooperation – A Sample Project

In the previous example, we addressed external actors that were not connected to the internal organization. In order to demonstrate a tighter form of cooperation, we introduce a new example. We consider a joint research project that is run by the "University" and a "Company".

Due to the nature of a project as "temporary endeavor" [12, p.5], the cooperation on the organizational level is restricted in time as well. Figure 5 illustrates the organizational structure of the project. It also shows relevant sections of the internal and external organizational model. The internal structures contain the *Research Group A* from the purchase example. Please note that some internal relations that are not relevant to this context have been omitted from the figure for clarity purposes.

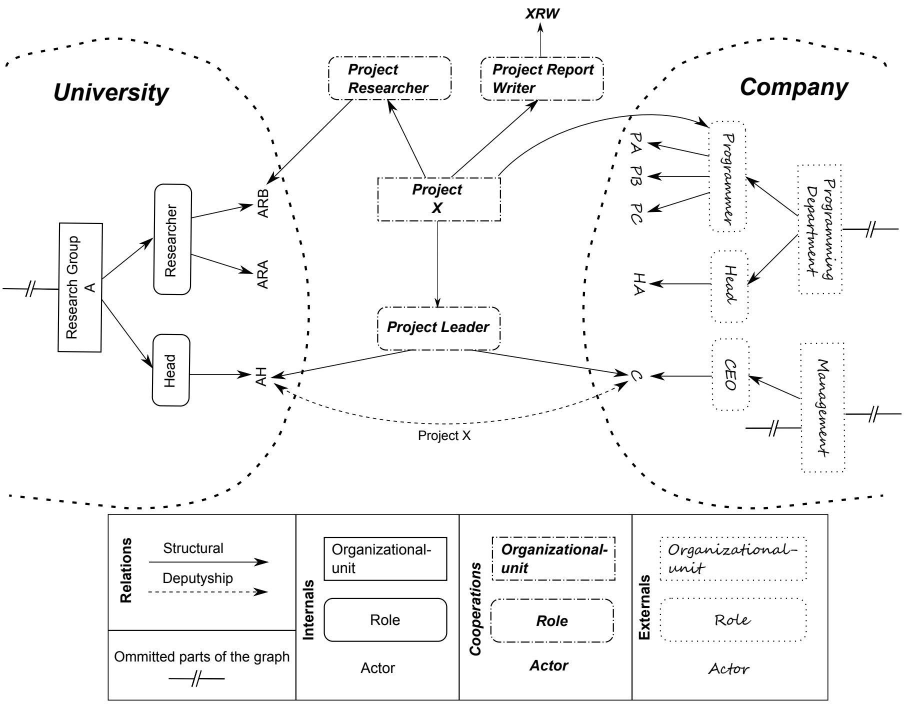


Fig. 5. Cross-organizational project

The *Project X* is represented as an organizational-unit in between the two cooperating organizations "University" and "Company". As such, it is neither internal nor external but a cooperation entity[7]. Within the project, there are the three roles *Project Researcher*, *Project Report Writer* and *Project Leader*. *ARB* is the only *Project Researcher*, *XRW* is the only *Project Report Writer*. The role *Project Leader* is shared between *AH* and *C*. As we can see, most of the actors that fulfill the roles of the joint project are members of organizations, except for *XRW*. *XRW* is a special employee that is employed specifically for the project and for the duration of the project.

As mentioned before, the structure on the "University" side basically consists of *Research Group A* with its *Head*, *AH*, and its *Researchers ARA*, *ARB* and *ARC*. Only *ARB* and *AH* participate in the project and fulfill their roles therein.

The "Company" contributes manpower to the project by providing the *Programmer*s from the *Programming Department* — *PA*, *PB* and *PC*. The *Head* of the organizational-unit, *HA*, is not involved. The *CEO C*, however, shares the *Project Leader* role. The role *CEO* is part of the *Management* organizational-unit of the "Company".

We just determined that *AH* and *C* share the role *Project Leader*. For practical purposes, they decided to be each other's deputy. If one of them is unavailable, the project team remains capable of acting. That is why they established a deputyship relation between each other. As the metamodel discussed in section 3 demands a context constraint, this relation is limited to matters regarding *Project X*. This prevents *C* from having authority on any issues that arise in the "University" outside the scope of the project. If we consider the purchase example from section 2, *C* can not act as supervisor of *ARA* and counter-sign their purchase request.

5 Algorithms to Identify Actors

The subject of this section are the algorithms for traversing the model concerning structural, organization-specific and user-defined relations. We illustrate the algorithms by describing the procedure of interpreting the sample language expressions of section 4 on the organizational model.

A part of the algorithms are excerpts that are described in a less formal manner in [11].

The knowledge hierarchy is used to define different levels of organizational knowledge. The hierarchy is specified from general organizational "rules" (top level) to most specific ones (bottom level). More details on this topic can be found in [8]. In this paper, the consideration of the knowledge hierarchy is reduced to a minimum. Thus, just the resulting actors out of the knowledge levels are listed in the traversal algorithms.

The result set of the traversal algorithms consists purely of actors (if it is not empty). In the following examples, the "Result" is only formed by uniting sets of actors. Intermediate results of traversals are excluded.

[7] As such, it belongs to the entity type $\mathcal{V}_{cooperation}$ discussed in section 3.

5.1 Definitions

Definition 1. $\mathbb{M}_{entity} \rightarrow_{relation-type} \mathbb{M}_R$*: Resulting set* $\mathbb{M}_R$ *of entities listed after traversal of entities connected by* $\rightarrow_{relation-type}$ *and starting from* $\mathbb{M}_{entity}$ *(the cardinal number of set* $\mathbb{M}_{entity}$ *is one, meaning exactly one entity is in that set)*

Definition 2. $\mathbb{M}_{entity} \rightarrow^{\mathcal{C}}_{relation-type} \mathbb{M}_R$*: This definition is analogous to the one before.* $\mathbb{M}_R$ *is the resulting set which lists entities after traversal.* $\mathbb{M}_{entity}$ *is the same as before. The traversal for entities which are possible candidates for the result set will* ***only*** *be done if the constraint* $c \in \mathcal{C}$ *is fulfilled.*

Definition 3. $\mathfrak{expression} \Rightarrow^* \mathbb{M}_{actors}$*: Set of actors* $\mathbb{M}_{actors}$ *resulting after traversal based on* $\mathfrak{expression}$

Definition 4. $\mathbb{M}_1 \cup \mathbb{M}_2 \cup \mathbb{M}_3 \Rightarrow \mathbb{M}$*: Set* $\mathbb{M}$ *results after* $\bigcup_{i=1}^{3} \mathbb{M}_i$*, with* i*: knowledge level. Knowledge level 3 (template level) is in this paper excluded to reduce complexity (for detail see [8], [10]).*

Definition 5. X^k*: Actor* X *results out of considering knowledge level* k *(*$\emptyset^i$ *means that no actor found regarding knowledge level* i*)*

5.2 Structural Traversal

The section describes the traversal of structural relations, instances of the relation type $\mathcal{HAS} \in \mathcal{R}_s$. The interpretation of the language expression starts with a lookup within the index (e.g. organizational-units $I_{\mathcal{OU}}$, roles $I_{\mathcal{ROLE}}$ or actors $I_{\mathcal{ACTOR}}$). If the interpretation is not terminated, the further processing of the traversal is done by following relations.

Example 1: $\boldsymbol{\mathcal{ACTOR}}$

Language Expression: "ARB"

1. Index $I_{\mathcal{ACTOR}}$ lookup for ARB
2. Result: $\{ARB\}$

Examples 2 and 3: $\boldsymbol{\mathcal{ROLE}(\mathcal{OU})}$

Language Expression: $Clerk(Sales)$

1. Index $I_{\mathcal{OU}}$ lookup for $Sales$
2. $\{Sales\} \rightarrow_{\mathcal{HAS}} \{Clerk\}$
3. $\{Clerk\} \rightarrow_{\mathcal{HAS}} \{SCA, SCB, SCC\}$
4. Result: $\{SCA, SCB, SCC\}$

Language Expression: $Dispatcher(*)$

1. Index $I_{\mathcal{OU}}$ lookup for $*$ [8]
2. $\{Sales, Logistics, Purchase\ Department, Research\ Group\ A\} \rightarrow_{\mathcal{HAS}} \{Dispatcher\}$
3. $\{Dispatcher\} \rightarrow_{\mathcal{HAS}} \{LDA, LDB\}$
4. Result: $\{LDA, LDB\}$

5.3 Explicit Search for a Specific Relation

The explicit search for organization-specific relations is valid for instances of $\mathcal{HAS\_SUPERVISOR} \in \mathcal{R}_o$ and $\mathcal{REPORTS\_TO} \in \mathcal{R}_o$ relation types. The lookup within indexes for the start of the interpretation is omitted (cf. section 5.2).

Example 4: $\mathcal{SUPERVISOR}(\mathcal{ACTOR})$

Language Expression: $SUPERVISOR(\text{“}ARB\text{”})$

1. Embedded Expression: “ARB” $\Rightarrow^* \{ARB\}$
2. Next Resolution: $SUPERVISOR(\{ARB\})$
   (a) $\{ARB\} \rightarrow^{\mathcal{C}}_{\mathcal{HAS\_SUPERVISOR}} \emptyset^1$
   (b) $\{ARB\} \rightarrow_{\mathcal{HAS}} \{Researcher, Lecturer\}$
   (c) $\{Researcher\} \rightarrow_{\mathcal{HAS\_SUPERVISOR}} \{Head\} \rightarrow_{\mathcal{HAS}} \{AH^2\}$
   (d) $\{Lecturer\}$ results in $\emptyset^2$
3. Result: $\emptyset \cup \{AH\} \Rightarrow \{AH\}$

The traversal step $\{ARB\} \rightarrow^{\mathcal{C}}_{\mathcal{HAS\_SUPERVISOR}} \{P\}$ is **not** included. The supervisor relation is role-dependent and if no role is specified in the language expression the constraint is by default violated. The example in section 5.4 shows one possibility to fulfill the constraint.

The continuing search proceeds analogously, for both the relations $\mathcal{HAS\_SUPERVISOR}$ and $\mathcal{REPORTS\_TO}$ (e.g. supervisor of supervisor, ...). The resulting set of the embedded expression(s) is the origin for the following resolution, and so on.

The organization-specific relations, except the $\mathcal{HAS\_DEPUTY}$ relation, are traversed across all levels of the knowledge hierarchy. The result set is formed by following the relations on actor, role and template level to a depth of 1.

5.4 Implicit Search

The traversal concerning instances of the $\mathcal{HAS\_DEPUTY} \in \mathcal{R}_o$ relation type is different to the other organization-specific relations. The deputy relation traversal terminates as soon as the concerned knowledge level is processed and the result set is not empty.
The following sample expression describes the traversal procedure for deputies.

[8] The semantics of $*$ is that all entries of the index are assigned to the resulting set.

Example 5: $\mathcal{ROLE}(\mathcal{OU})$
***WITH* <parameter>=<value> *AND CONTEXT* =<value>**

Language Expression (*PA, PB* are unavailable): $Clerk(Purchase\ Department)$ $WITH\ price$ =“1500” $AND\ CONTEXT$ =“PurchaseOrder”

1. Expression: $Clerk(Purchase\ Department) \Rightarrow^* \{PA, PB\}$
   (a) $\{PA\}$ is unavailable thus[9] $\emptyset^1 \cup \emptyset^2 \Rightarrow \emptyset \cup$
   (b) $\{PB\} \rightarrow^{\mathcal{C}}_{\mathcal{HAS\_DEPUTY}} \{PC^1\}$ [10]
2. Result: $\emptyset \cup \{PC\} \Rightarrow \{PC\}$

Example 6: $\mathcal{SUPERVISOR}(\mathcal{ROLE}(\mathcal{OU}))$

Language Expression: $SUPERVISOR(Lecturer(*))$

1. Embedded Expression: $Lecturer(*) \Rightarrow^* \{ARB\}$
2. Next Resolution: $SUPERVISOR(\{ARB\})$
   (a) $\{ARB\} \rightarrow^{\mathcal{C}}_{\mathcal{HAS\_SUPERVISOR}} \{P^1\}$ [11]
   (b) $\{ARB\} \rightarrow_{\mathcal{HAS}} \{Researcher, Lecturer\}$
   (c) $\{Researcher\} \rightarrow_{\mathcal{HAS\_SUPERVISOR}} \{Head\} \rightarrow_{\mathcal{HAS}} \{AH^2\}$
   (d) $\{Lecturer\}$ results in $\emptyset^2$
3. Result: $\{P\} \cup \{AH\} \Rightarrow \{P, AH\}$

The language expressions discussed previously are just a small subset of possible statements. They serve to illustrate key aspects of the traversal algorithms. For further information on the grammar of the language, refer to [11].

6 Prototype

This section explains the structure of the editor that manages, among others, the organizational model. For clarity, the example screenshot depicts only the organizational model of the organization “University”. The “*IS*” relations in the organizational graph are a result of implementation considerations that improve the runtime behavior. They are analogous to the “*HAS*” described on the conceptual level described in section 3.

The editor, depicted in figure 6, can be broken down into different partitions. The left part, the “navigation” area, displays the different entities of the organizational graph shown in the center. The green icons labeled “O” indicate organizational-units, blue icons labeled “F” are roles (corresponding to functional-units described in [8]) and red icons labeled “A” are the actors. The tree structure on the left shows

[9] There is no deputy relation, concerning the actor (PA) or their role ($Clerk$), on any different knowledge level.
[10] The constraint “$PurchaseOrder.price$ < ”2000”” is fulfilled by the value of the parameter and the context is valid.
[11] The role *Lecturer* is explicitly stated in the language expression. Consequently, the constraint (role-dependent relation) is fulfilled.

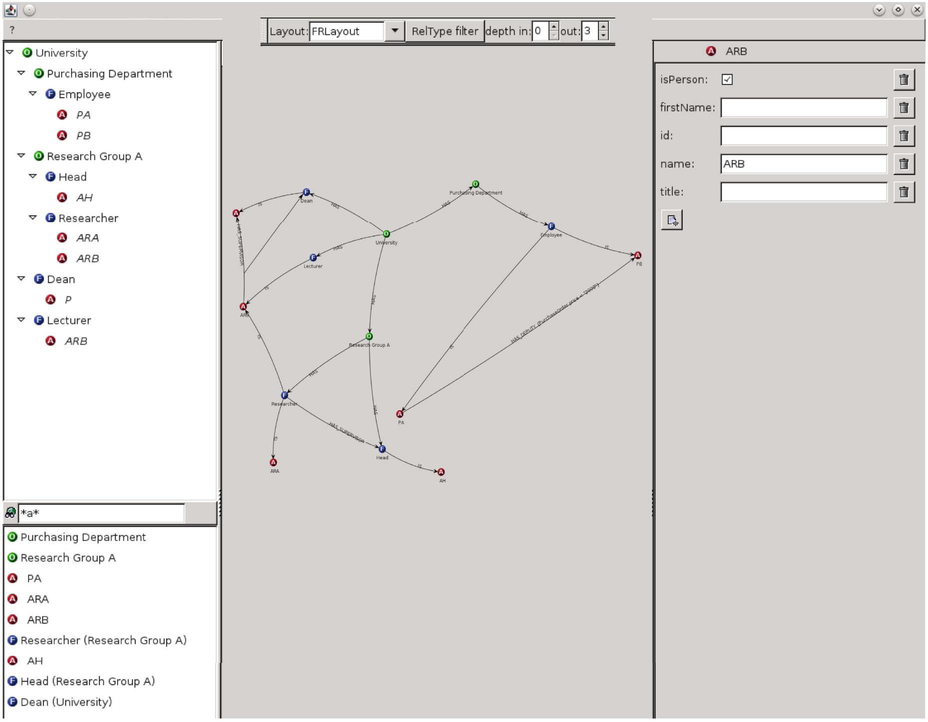


Fig. 6. Screenshot: The graph-based editor of $\mathcal{C} - \mathcal{ORG}$

the organizational structure as quasi-hierarchical composition. Roles are represented as aggregation of actors, organizational-units are aggregations of roles and organizational-units. The drawback of this form of representation is that entities may be shown multiple times. The actor *ARB* is listed twice as he is a *Researcher* as well as a *Lecturer*. He is just displayed twice but stored only once in the organizational model (cf. figure 7).

The tree can be used to quickly navigate to specific entities of the organizational graph (cf. center of figure 6 and figure 7).

By clicking on any icon, the organizational graph focuses on the (sub-)graph of the model. The GUI elements on top configure the depth of this graph (*in*coming and *out*going edges). The depth setting affects only structural relations. It has no influence on organization-specific, user-defined and role-dependent relations. In this screenshot the outgoing depth is 3. The topmost organizational-unit "University" was selected so that the figure shows the resulting (sub-)graph. Furthermore, all organization-specific, user-defined and role-dependent relations, which have relations within this subgraph, are included. An example is the constrained deputy relation between the actors *PA* and *PB*.

If the models of organizations are bigger and more complex than this easy example, the search area on left bottom helps to find entities. The search supports looking for the exact entity (e.g. *ARB* and *Research Group A*). It can also be used to search based on text segments of the entity name (e.g. *Univ** and **a*). A combination of both search strategies is depicted in figure 6. Clicking on a single search result leads to the same functionality as clicking on an item in the "navigation" part.

The right partition is used to edit entities and relations which are selected in the organizational graph (center partition). This "edit" partition can also be used to add or remove attributes of entities. The administration of constraints on relations is also done in this area. In this example, the actor *ARB* is selected within the organizational graph. Consequently, the "edit" area shows *ARB*'s attributes and their values as depicted in 6.

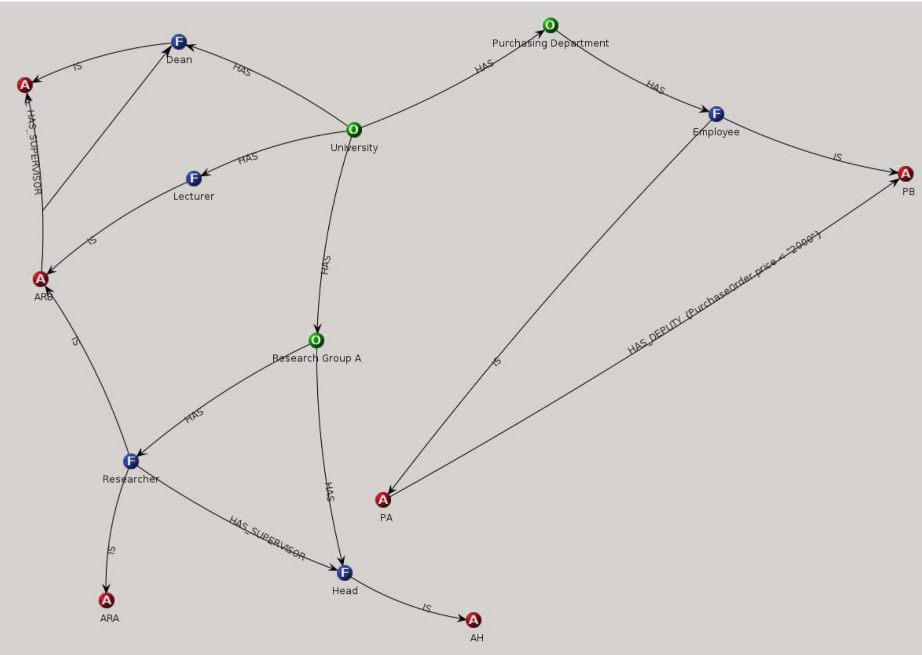


Fig. 7. Screenshot: The organizational structure

The center area of the editor is the main area for displaying the organizational model as arbitrary directed graph (cf. figure 7). This makes it more powerful than the tree structure of the "navigation", as it is not limited to hierarchical structure relations. It can directly show the organization-specific, user-defined and role-dependent relations.

The "graph" partition can be used to connect entities within the organizational model. There are different ways to do so. The first is that a user clicks on an entity and drags a line to an entity. Then, a pop-up dialogue is shown and the user has to decide on a relation type. The relation types are the ones described in section 3 (structural, organization-specific and user-defined).

After the selection of the relation type, the metamodel is used to verify the consistency of the model. If the selected relation type between the selected entities violates the metamodel, the user is informed by an error message and can change it immediately. If the model is consistent with the metamodel, it can be stored in the organization server[12].

Another option to connect entities with relations is to right-click on an entity. After clicking the pop-up menu item "Connect...", the user can select the entity to connect, as well as type and direction of the relation (incoming / outgoing). The selection of the entity can also be done with the aid of a search function. The consistency check is the same as described previously.

In general, the editor is a GUI that enables the user to perform the following operations on the model:

- *Create* entities and relations.
- *Read* subgraphs of the organizational model and display them in an intuitive manner.
- *Update* entity attributes and constraints on relations. The start and end entities of relations can be altered as well.
- *Delete* relations and entities while maintaining consistency[13].

7 Conclusion and Outlook

The approach we described in this contribution makes it possible to describe participants in cross-organizational business processes. This can be done based on their organizational context, as opposed to the total enumeration of members of roles required in traditional approaches. External subjects can be resolved to concrete actors in the same consistent way. They need not be treated as complete black boxes.

In addition to representing cooperations based on business processes, our approach is able to represent cooperations on an organizational level. This is especially true for joint projects that represent a temporary structural relation between two or more organizations but are not defined by reoccurring choreographies. As both internal and external organizational entities and relations are kept within the same model, it is possible to transfer external structures into the internal structures. This may be relevant when considering the acquisition of one organization by another or the foundation of a holding company.

The metamodel enables us to declare role-dependent relations. This means that the relations that are valid for a concrete actor is determined by the role the actor assumes at the time. If we consider the organizational model from the example, it makes a difference whether *ARB* requests a purchase as a *Researcher* or as *Lecturer*. Different people will be their supervisor and responsible for approving the purchase.

[12] The organization server stores the concrete organizational model of the companies (cf. [10, fig. 2]).

[13] Entities can only be deleted if they are no longer interconnected.

There are two major directions of research that will be pursued in the future. First, the language can be extended to allow for more fine-grained control over the traversal. For auditing purposes, it makes sense to extend the language expressions so that it is possible to ask for all *possible* deputies, supervisors, etc. of actors.

The second area of research is more directly related to the cross-organizational aspects of the approach. In such scenarios there should be a way to reference entities from other *models* in a concrete model. This would allow organizations to "publish" sections of their organization in a formal way based on the metamodel described above. This could serve as a formal description of organizational interfaces.

References

1. Vahs, D.: Organisation: Einführung in die Organisationstheorie und -praxis. Schäffer-Poeschel (2007)
2. Fulda, H.: Neue Organisationsformen und ihre informationstechnische Realisierung. PhD thesis, Universität St. Gallen (2001)
3. Meier, A., Stormer, H.: eBusiness & eCommerce. Springer, Heidelberg (2012)
4. Krcmar, H.: Informations management. Springer, Heidelberg (2010)
5. Wüthrich, H., Philipp, A., Frentz, M.: Vorsprung durch Virtualisierung. Gabler, Wiesbaden (1997)
6. Schulte-Zurhausen, M.: Organisation. Franz Vahlen (2005)
7. Schwarzer, B., Krcmar, H.: Neue Organisationsformen - Ein Führer durch das Begriffspotpurri. In: IM Information Management, vol. 9, pp. 20–27 (1994)
8. Lawall, A., Reichelt, D., Schaller, T.: Intelligente Verzeichnisdienste. In: Barton, T., Erdlenbruch, B., Herrmann, F., Müller, C. (eds.) Herausforderungen an die Wirtschaftsinformatik: Betriebliche Anwendungssysteme, AKWI 2011, pp. 87–100. News & Media, Berlin (2011)
9. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Brger, E.: Subjektorientiertes Prozessmanagement - Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern. Hanser (2011)
10. Lawall, A., Schaller, T., Reichelt, D.: An approach towards subject-oriented access control. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 33–42. Springer, Heidelberg (2012)
11. Lawall, A., Schaller, T., Reichelt, D.: Integration of dynamic role resolution within the s-bpm approach. In: Fischer, H., Schneeberger, J. (eds.) S-BPM ONE 2013. CCIS, vol. 360, pp. 21–33. Springer, Heidelberg (2013)
12. Larson, E.: Project Management: The Managerial Process. McGraw Hill (2010)

A Detailed Purchase Request

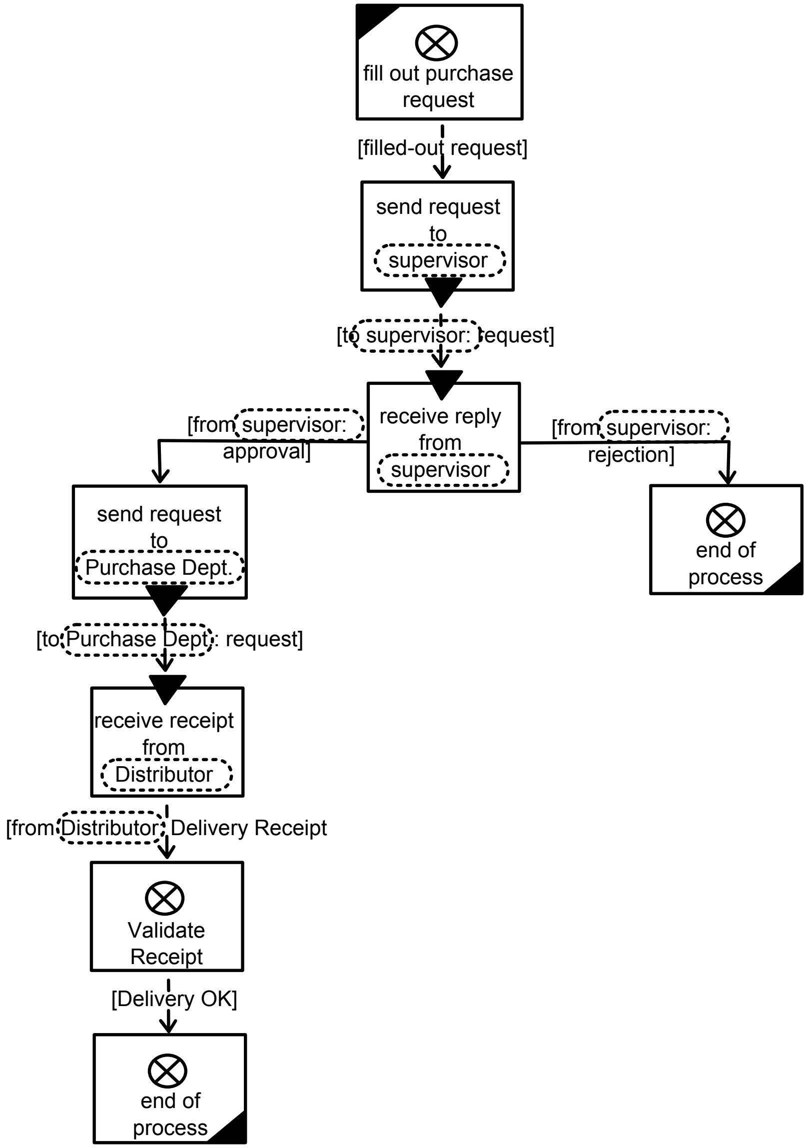


Fig. 8. Purchase request process within the *Customer* subject

Multi-agent Solution for Business Processes Management of 5PL Transportation Provider

Anton Ivaschenko

Magenta Technology Development Center, Samara, Russia
anton.ivashenko@gmail.com

Abstract. This paper describes a model and technology for business processes management of Fifth Party Logistics (5PL) provider in transportation industry. The proposed model provides formalization of transportation orders' assignment in the form of events able to consider the flexibility of actors' interaction. The solution is based on implementation of overlay networks for actors generated analyzing the variability of orders assignment. The results are illustrated by the description of multi-agent software-as-a-service solution implementing the proposed approach.

Keywords: Multi-agent technology, Fifth party logistics, 5PL, transportation, scheduling, decision making support.

1 Introduction

One of the most up-to-date trends in transportation logistics introduces 5PL (Fifth Party Logistics) concept [1], which is based on implementation of a number of services for customers and shippers provided by the specially designed software platform. 5PL platform is open for new transportation companies and even drivers and helps them negotiating with customers in integrated information space. For example, a new coming taxi driver can use a handheld device to register and start receiving new orders according to his current position and capabilities.

One of the main features of such an approach is high uncertainty in number and time of resources available. Each transportation company or even driver becomes an independent actor with its own objectives and constraints and is motivated mainly by his own interest. This makes it impossible to implement standard approaches for scheduling and business-process managements based on a support of the solid business process for all the orders and executors in the system.

One of the solutions can be close to subject-oriented approach for business processes management (S-BPM), which conceives a process as a collaboration of multiple subjects organized via structured communication [2]. There can be proposed a model for interaction of actors (subjects) in integrated information space of 5PL operator, which can be implemented using multi-agent software. In this paper we describe one of the possible solutions of this problem.

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 110–120, 2014.

2 State of the Art

The concept of Fifth Party Logistics (5PL) is introduced by analogy with 3PL that provides the process of outsourcing of transportation resources and 4PL that describes a concept of Lead logistics integrator (still there is no general definition of it in the business world) [1]. 5PL provider owns no transportation resources itself but makes available a special service able to link suppliers and buyers. This service is based on the IT infrastructure, which plays the general role in 5PL business. Customer representatives, transport managers, shippers, carriers, and even drivers become users of a certain IT platform. The purpose of this platform is to allocate incoming orders to appropriate resources, consolidate them improving consolidation and reducing idle time and generate efficient schedules for drivers and vehicles.

This idea is close to the popular SaaS (Software as a Service) business model, according to which software and associated data are centrally hosted on the cloud. Such service becomes attractive for small transportation companies and allows outsourcing dispatching functions for large logistics operators.

Still to ensure high efficiency of 5PL service both for customers and executors in terms of time and costs there is a request to implement modern technologies of business processes management based on decentralized architectures, distributed intelligence and multi-agent technology. This happens because of the increasing number of decision makers, high uncertainty and dynamics of changes, and flexibility of decision making logic. The example of using multi-agent technology for business processes simulation can be found at [3]. Also the described approach generalizes our experience of multi-agent solutions development for transportation logistics [4, 5].

As soon as the information space provided by 5PL software platform can be treated as a complex network of continuously running and co-evolving actors (or subjects), the whole solution can be based on holons paradigm [6] and bio-inspired approach [6]. This paradigm and approach offer a way of designing adaptive systems with decentralization over distributed and autonomous entities organized in hierarchical structures formed by intermediate stable forms. It's implementation in practice requires development of new methods and tools for supporting fundamental mechanisms of self-organization and evolution similar to living organisms (colonies of ants, swarms of bees, etc) [7].

The actors compete and cooperate, coordinate and adapt their behaviors, aggregate their services to users and take various requirements individually. Each event that occurs here can influence the whole network and needs a collaborative reaction from all subjects that take into account personal objectives and constraints of each decision making member. Another requirement for the decision making process based on subject' negotiation is that the final decision can require a complicated and time consuming process of data exchange between the actors. That's why it should be managed to consider time factor and assure functioning in real time.

One of the recent developments in this area [8] introduces featuring a clear separation between the local planning performed by the individual vehicles and the global coordination achieved by negotiation. To solve such a kind of problems there can be implemented a special functionality for a statistical analysis based on recent developments in cross-correlation analysis of non-equidistant time series [9]. The models and methods of such analysis were successfully probated in social management and can be reused for management of multi-agent negotiations.

Also in this area there can be used the event processing techniques [10] for an effective continuous processing of time sensitive data in control centers. This technology deals with the analysis of streams of continuously arriving events with the goal of identifying instances of predefined meaningful patterns (complex events). Event processing offers a variety of special operations that are applied on events (e.g., event filtering, projecting, aggregating, splitting, transforming etc.), and enables a special (the event-driven) interaction model.

In many cases however, real-time awareness provided by event processing is not sufficient; real time actions need to be triggered not only by events, but also upon evaluation of additional background knowledge [11]. This knowledge captures semantic metadata descriptions (the domain of interest), and the context related to critical actions and decisions. Its purpose is to be evaluated during detection of complex events in order to on-the-fly enriched events with relevant background information or to propose certain intelligent recommendations in real time.

In this paper we introduce a model for event-based description of actors' interaction using 5PL provider and formalize a group of key performance indicators (KPIs) that help understanding its efficiency considering the uncertainty and influence of time factor inherent to its business. This approach can help rising the subjects' income and reduction of the decision making time and as a result increasing the service appeal of 5PL software platform for its users.

3 Event-Based Model for 5PL Business Processes Representation

Let us consider a generalized business model where orders (or jobs) w_i are proceeded to actors (or subjects) u_j. Any actor can be assigned to perform any order, incurring some cost that may vary depending on the exact assignment. It is required to perform all the orders by assigning exactly one order to each actor in such a way that the total cost of the assignment is minimized. The centre is introduced as a solid dispatching agent that offers the orders to actors and ensures the effectiveness of the whole system.

The objective of the order agent is to be proceeded by any actor available on time (particular the KPIs can be formulated as "early average absorption"). The actor's objective is to receive the most corresponding orders with the highest relevance.

Let us set the following order lifecycle events, represented by Boolean variables:

$e^*(w_i, t^*_i) \in \{0, 1\}$ – appearance of w_i, t^*_i is the time of its appearance;

$e(w_i, u_j, t_{i,j}) \in \{0, 1\}$ – offer of w_i to u_j at time $t_{i,j}$;

$e'(w_i, u_j, t'_{i,j}) \in \{0, 1\}$ – assignment of w_i to u_j at time $t'_{i,j}$;

$e''(w_i, t''_i) \in \{0, 1\}$ – escape of w_i at time t''_i in case of order rejection.

The cost of order w_i execution by actor u_j is $c_{i,j}$. It is determined by the actor and proposed to the center. Let us assume that one actor cannot execute several orders at a time. The allocation problem for this model can be represented as

$$\begin{aligned} &\sum_{i=1}^{N_w}\sum_{j=1}^{N_u} e'\left(w_i, u_j, t'_{i,j}\right)\cdot c_{i,j} \to \min, \\ &\sum_{j=1}^{N_u} e'\left(w_i, u_j, t'_{i,j}\right) = 1, \quad i = 1..N_w, \end{aligned} \tag{1}$$

$$\sum_{i=1}^{N_w}\sum_{j=1}^{N_u} e^*\left(w_i, t^*_i\right)\cdot e'\left(w_i, u_j, t'_{i,j}\right)\cdot\left(t'_{i,j} - t^*_i\right) \to \min\ , \tag{2}$$

where N_w is the total number of orders and N_u is the total number of actors.

For the order flow $e^*\left(w_i, t^*_i\right)$ there should be developed a strategy (schedule of offers) $e\left(w_i, u_j, t_{i,j}\right)$ for a set of u_j that will reach (1) and (2).

From the other side, each actor considering the order flow $e\left(w_i, u_j, t_{i,j}\right)$ should decide on the strategy $e'\left(w_i, u_j, t_{i,j}\right)$ that comes out at

$$\forall u_j : \sum_{i=1}^{N_w}\sum_{k=1}^{N_u} e\left(w_i, u_j, t_{i,j}\right)\cdot e'\left(w_i, u_j, t'_{i,j}\right)\cdot\left(1 - e'\left(w_i, u_k, t'_{i,k}\right)\right)\cdot c_{i,j} \to \max\ . \tag{3}$$

In case the actor starts execution as soon as the order is allocated the following limitation is valid:

$$\forall u_j : \sum_{i=1}^{N_w}\sum_{l=1}^{N_w} e'\left(w_i, u_j, t'_{i,j}\right) e'\left(w_l, u_j, t'_{l,j}\right)\cdot\left(1 - \theta\left(t'_{l,j} - t'_{i,j}\right)\cdot\theta\left(t'_{i,j} + \Delta t_{i,j} - t'_{l,j}\right)\right) = 0\ ,$$

where $\theta(x)$ – Heavyside step function [12]: $\theta(x) = \begin{cases} 0, x < 0 \\ 1, x \ge 0 \end{cases}$.

The problem (1, 2, 3) is introduced as a problem of "proactive allocation". Its direct solution is not possible as soon as the number and availability time frames of resources and orders changes with time. To prove it there can be specified the following logic. Firstly, the statement (3), being summarized by u_j, results in a contradiction with (1). Secondly, to solve (1) one needs to fix the number of events considered, but at any moment of time t^* there is no information about the events $e^*\left(w_i, t^*_i\right) : t^*_i > t^*$, and there cannot be proposed any substantial approach on how to pick-out the orders w_i accepted by 5PL platform for scheduling in real time.

So there should be developed new approaches for project management of scheduling business processes.

The following challenges can be specified for a 5PL provider:

- attraction of customers and executors in order to increase the number of options for each order allocation;
- enforcing interaction conditions to support competition and cooperation between the users of 5PL platform, which is beneficial for them;
- estimation and analysis of key performance indicators (KPIs) of 5PL business processes in order to increase the level of service.

Considering the features of the proposed model and in reliance on the theory of constraints there can be formalized the following KPIs:

Total costs of the orders being allocated to a certain time frame:

$$C = \sum_{i=1}^{N_w} \sum_{j=1}^{N_u} e'\left(w_i, u_j, t'_{i,j}\right) \cdot c_{i,j} \,, \tag{5}$$

Average resources utilization:

$$L = \frac{1}{N_u} \sum_{i=1}^{N_w} \sum_{j=1}^{N_u} e'\left(w_i, u_j, t'_{i,j}\right) \cdot \Delta t_{i,j} \,, \tag{6}$$

where $\Delta t_{i,j}$ – is the duration of loading of u_j by w_i in case of corresponding allocation.

Bound order set:

$$\Theta = \frac{1}{N_w} \cdot \sum_{i=1}^{N_w} \sum_{j=1}^{N_u} e^*\left(w_i, t^*_i\right) \cdot e'\left(w_i, u_j, t'_{i,j}\right) \cdot \left(t'_{i,j} - t^*_i\right). \tag{7}$$

Statements (1 – 3) together with KPIs (5 – 7) form the event-based model for 5PL provider and can be used in multi-agent systems for simulation and evaluation of business processes in transportation logistics.

4 5PL Business Processes Management Technology

Functioning of 5PL provider contains a number of interrelated business processes for supply chain members, which proves the advantages of application of subject-oriented approach in this area. While developing multi-agent solutions for transportation logistics [4] we usually have specified two groups of agents representing demands and supplies and negotiating in order to find trade offs. For example, there can be introduced the driver agents that strive to maximize utilization and the order agents that try to allocate at lower costs.

For each group of agents there can be proposed a separate business process, which includes a number of states connected by the relations of precedence. The order can be input into the system, stay for a while in events queue, become scheduled to a certain resource, proceed to execution state after it starts and finalize being done or rejected in case of exception. Each driver repeats iteratively the states of being free or busy with order with some intermediate states of pick-ups, drops and idle moves. The generalized scheduling problem involves determination of a consistent combination of these states across the planning horizon in the near future.

Special aspects of 5PL service make it necessary to consider these business processes from a different point of view. In case of drivers' flexibility to take a certain order or reject it there can be evaluated no consistent planning horizon – the time interval of resources availability and the list of orders to be done. The center offers the orders to appropriate resources with no guaranty that they will be accepted for execution. That's why it should evaluate and analyze the probability of orders acceptance and allocation.

In transportation logistics much is determined by the network of geographical locations and roads. In case the order is associated by a location of appearance or pick-up the probability of its allocation is dependent upon the density of executors in the near area. Considering the KPIs (5 – 7) the same approach can be introduced in terms of time. The higher is the number of resources free from loading and waiting for new orders at a certain location, the higher is the probability of orders being allocated at this point.

Still we should consider two constraints:

- executors waiting too long would escape (as soon as the 5PL platform is open for entry and exit);
- drivers would congregate at locations with the highest density of orders (this is valid for taxi business: e.g. taxi drivers are attracted to airports and tourist sights).

Both factors influence the service level and lower the force of attraction of 5PL provider. To overcome this challenges the dispatching center of 5PL provider should attract the actors' interest and aggregate drivers at proper areas.

The event flow of orders' appearance $e^*(w_i, t^*_i)$ in non-equidistant. According to the statement (3) of the problem definition there should be the minimum time between order income and assignment.

The way to solve this problem is to specify such an event flow of assignments $e'(w_i, u_j, t'_{i,j})$ that will reduce the variability of assignments in reaction to the irregular event flow of incoming orders with minimum waiting time. This is similarly to providing line balancing in production project management. One of the solutions runs out of the combination of (6) and (8) – we suggest introducing the rhythmical assignment:

$$\rho'_k(k \cdot \Delta\tau) = \sum_{i=1}^{N_w} \sum_{j=1}^{N_u} e'(w_i, u_j, t'_{i,j}) \cdot \theta(k \cdot \Delta\tau - t'_{i,j}) \cdot \theta(t'_{i,j} - (k-1) \cdot \Delta\tau) \rightarrow const. \quad (8)$$

where $\Delta\tau$ is the time sampling interval,
N_w is the total number of orders and N_u is the total number of actors,
and $\theta(x)$ – Heavyside step function (introduced above).

This schedule of assignments sets up the moments, before which the center needs to determine the options and send the offers to the actors according to the priority of waiting time. The statement (8) can be used in practice to prioritize the orders taken out for scheduling.

Therefore the set of offers $e(w_i, u_j, t_{i,j})$ can differ for each actor, and the same orders can be offered for several different actors. In addition to this the representation of the current transportation network for each actor can vary as soon as the center provides limited information about the orders and the current transportation situation. Let us call such subset of data a Virtual Overlay Network – a graph, which contains nodes representing geographical locations and links simulating distances between them. For every node there can be specified a set of orders with the same pick-up point corresponding to this node. The virtual overlay network can be specific for each actor describing the current situation in the individual scene and change with time.

This concept is given by analogy with peer-to-peer (P2P) networks in telecommunications where an overlay network is usually a computer network which is built on the top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network.

One of the possible technologies implementing virtual overlay networks is P2P outsourcing [13] based on the series of virtual auctions among the executors. Getting the incoming events of orders' appearance the centre generates offers or bids $e(w_i, u_j, t_{i,j})$ to the actors (the candidates are determined according to geography, their current position and adaptive planning horizon). One order can be offered to several actors. The actors that have received a number of offers can choose the most profitable for them and commit the assignments $e'(w_i, u_j, t'_{i,j})$. This negotiation scheme allows putting into practice the schemes that force competition among the actors (to win the most profitable order) and cooperation between them (by creating temporal groups in the areas of orders' high frequency). In such a way the center does not distribute all the orders directly to actors, but initiate a competition between them.

It should be mentioned, that generating of $e(w_i, u_j, t_{i,j})$ means not only developing a number of bids (what order to offer to which actor), but also decision on time when to send them. Thus the center can hold back recent orders and give higher priority to the orders that are waiting for a long time. To make this decision the center should consider the indicators, introduced above.

The proposed approach is illustrated in Fig. 1. In the given example the order 5 (that came later than others) is reduced from the view of Actor 1. Besides the set of orders 2, 3, and 4 are hidden from Actor 2. As a result both actors are interested in order 1. Is should be mentioned that the hidden data will appear for the actors with time, so in any case they retain an option to wait till the situation changes.

Fig. 2. describes the results of simulation carried out for a transportation network. Several strategies were implemented to compare directive allocation with management based on constructing of overlay networks. The first graph represents the dependence of effectiveness of orders allocation (ratio of total load time to total idle time). The results are quite close depending on the number of locations. The second graph presents average decision making time – one can see that introducing overlay networks allows to minimize the complexity of decision making and as a result to reduce time spent by orders to wait for an allocation, which supports the statement (7). Average decision

making time characterizes time frame between the time of appearance and assignment for each order. In case it is lower the KPI for the bound orders set (7) is minimized.

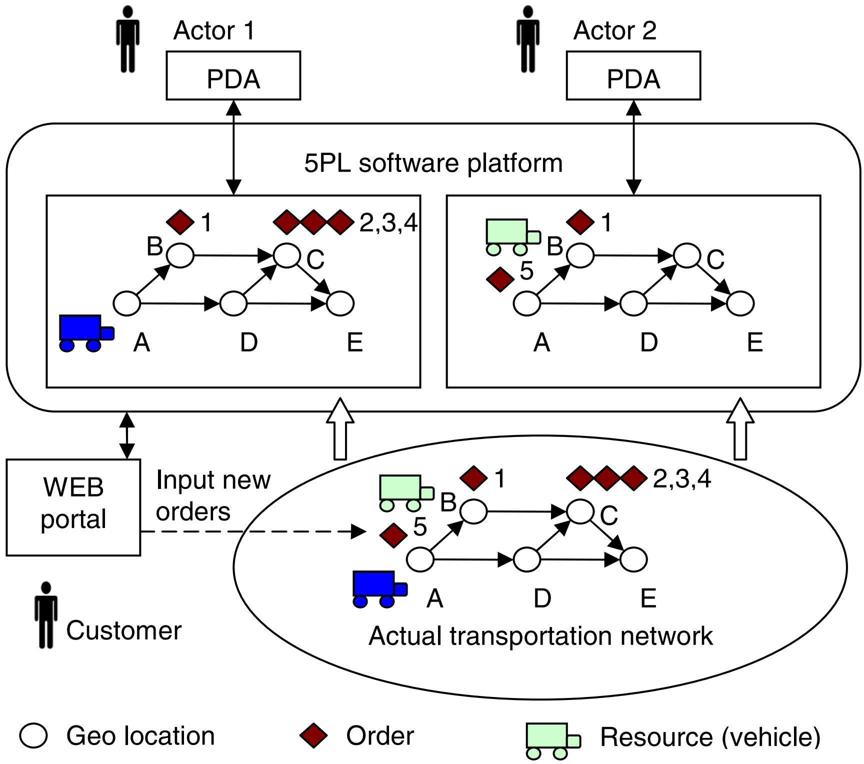


Fig. 1. 5PL provider software platform architecture: Both actors-executors are stimulated to compete for order in location A

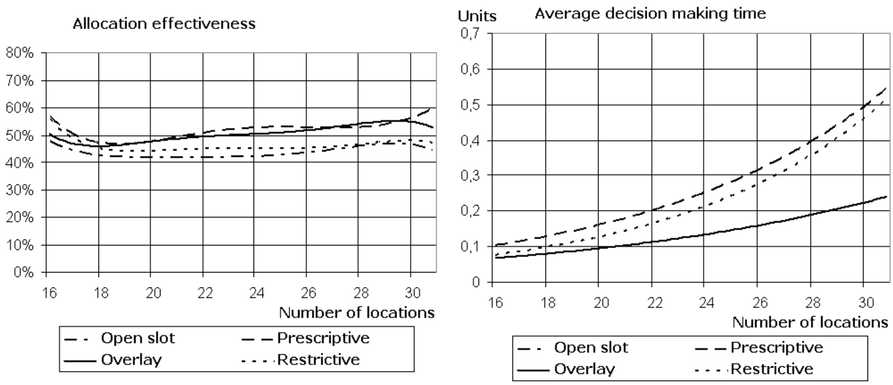


Fig. 2. Simulation results

5 Maxoptra Solution

The described approach was implemented in the Maxoptra Web based solution (see Fig. 3) that functions on a Software as a Service (SaaS) basis and is available from any PC with Internet access. Scheduling screen is presented in Fig. 4.

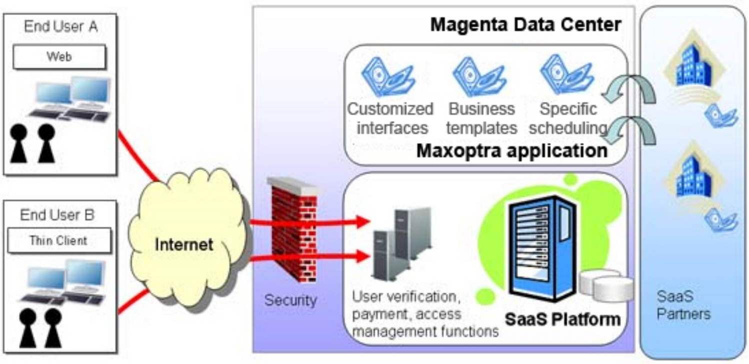


Fig. 3. Maxoptra logical architecture

The Maxoptra functionality includes:

- order entry: manual order file upload and integration with an existing order management solution;
- manual or automatic jobs scheduling;
- orders allocation to vehicles and optimal vehicle route creation;
- dynamic and batch scheduling support;
- adjustable scheduling objectives: cost reduction, mileage reduction, optimized own resource utilization, etc.;
- manual plan adjustment by means of a ‘drag and drop’ mechanism;
- vehicle capacity planning;
- visualization of: delivery routes on the map, real time mileage and cost KPIs;
- workflow management, printed manifests and management reports;
- planned vs. actual analysis of: vehicle locations, delivery times, followed routes;
- proof of delivery signature captured and password protected administration.

Available resources (including drivers and vehicles) and customer data is captured in a knowledge base (ontology). It is very important to capture statistics and link it to the knowledge about orders and resources (e.g. certain VIP clients prefer to take specific drivers, some drivers prefer certain locations or are not allowed to visit specific locations, etc.) This can be used in order to introduce special client management and pricing schemes and enrich the agents decision making logic.

One of the features is that the drivers are included into the scheduling process not only as providers of actual data, but also as decision makers. When several orders can be scheduled to one driver these can be sent to his handheld device. So that the driver can make a choice between several orders or consolidate them if it is possible. As soon as the same proposals can be sent to different drivers the system can initiate a competitive process of orders assignment (auctioning). In addition to this, the data flow generated for each driver forms a virtual overlay network for him. The number and sequence of orders sent to drivers can be used for manipulation. In such conditions drivers are free to make their own decisions to some extent, but, nevertheless, stay under the dispatcher's control.

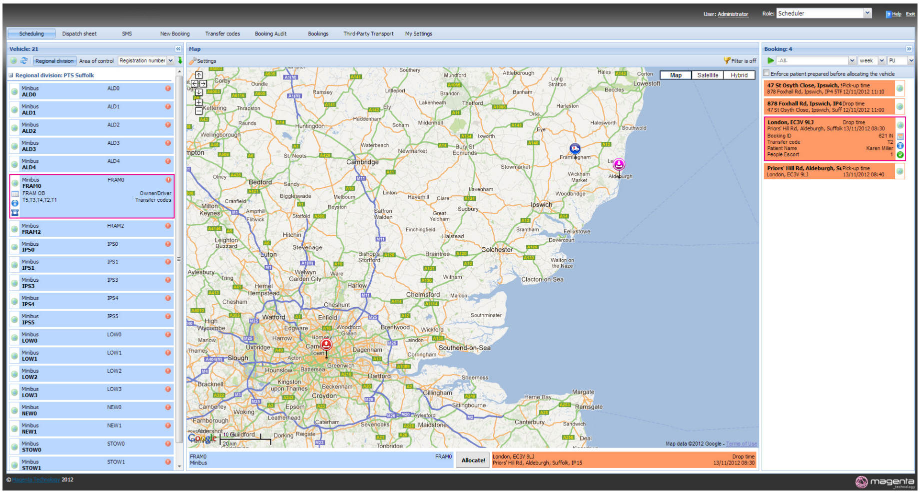


Fig. 4. Scheduling screen

As the result the Maxoptra solution provides:

- advanced booking facilities for customers to enable a quick, streamlined booking process including return bookings;
- automatic real-time dynamic scheduling of orders to ensure effective and cost efficient transportation operations through the following: notification of optimum booking preferences at the time of booking; real time tracking of vehicles' location; fleet optimization based on a variety of criteria to minimize empty mileage and maximize vehicle capacity utilization, reduce operational costs, and meet agreed customer service levels;
- a secure yet easy way to access and manage data through the web, available for vehicle suppliers (to access vehicle details and data on resource availability).

6 Conclusion

In this paper we have proposed a solution for business processes management of Fifth Party Logistics (5PL) provider in transportation industry. This solution is based on multi-agent approach and introduces the event-based model for simulation and analysis of 5PL transportation logistics, the technology for transportation orders proactive allocation based on the implementation of virtual overlay networks, and the multi-agent software solution that illustrates the validation.

One of the main benefits of the proposed solution is that it considers the flexibility of actors' interaction. The users of 5PL software platform act as independent decision makers and require subject-oriented business processes management to be supported. In this paper we have considered some special aspects and peculiarities of such an approach and provided a technology of how to implement it in practice.

References

1. Hickson, A., Wirth, B., Morales, G.: Supply chain intermediaries study, University of Manitoba Transport Institute, 56 p. (2008), http://umanitoba.ca/faculties/management/ti/media/docs/scIntermediariesfinalwithcover(1).pdf
2. Fleischmann, A., Kannengiesser, U., Schmidt, W., Stary, C.: Subject-oriented modeling and execution of multi-agent business processes. In: Proceedings of 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Atlanta, Georgia, USA, November 17-20, pp. 138–145 (2013)
3. Onggo, S.: Agent-based simulation model representation using BPMN 2013. In: Fonseca i Casas, P. (ed.) Formal Languages for Computer Simulation, pp. 378–399. IGI Global, Hershey (2013)
4. Andreev, V., Glashchenko, A., Ivaschenko, A., Inozemtsev, S., Rzevski, G., Skobelev, P., Shveykin, P.: Magenta multi-agent systems for dynamic scheduling. In: Proceedings of the 1st International Conference on Agents and Artificial Intelligence (ICAART 2009), pp. 489–496 (2009)
5. Glaschenko, A., Ivaschenko, A., Rzevski, G., Skobelev, P.: Multi-agent real time scheduling system for taxi companies. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, pp. 29–36 (2009)
6. Leitão, P.: Holonic rationale and self-organization on design of complex evolvable systems, HoloMAS 2009. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 1–12. Springer, Heidelberg (2009)
7. Gorodetskii, V.I.: Self-organization and multiagent systems: I. Models of multiagent self-organization. Journal of Computer and Systems Sciences International 51(2), 256–281 (2012)
8. Kalina, P., Vokřínek, J., Mařík, V.: The art of negotiation: Developing efficient agent-based algorithms for solving vehicle routing problem with time windows. In: Mařík, V., Lastra, J.L.M., Skobelev, P. (eds.) HoloMAS 2013. LNCS, vol. 8062, pp. 187–198. Springer, Heidelberg (2013)
9. Prokhorov, S.: Applied analysis of random processes, Samara scientific center of RAS, 582 p. (2007), http://window.edu.ru/resource/665/58665/files/9_pasp.pdf
10. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream reasoning and complex event processing in ETALIS. In: Semantic Web Journal (2012), doi:10.3233/SW-2011-0053; Special Issue: Semantic Web Tools and Systems
11. Artikis, A., Paliouras, G., Portet, F., Skarlatidis, A.: Logic-based representation, reasoning and machine learning for event recognition. In: Proceedings of the 4th ACM International Conference on Distributed Event-Based Systems, New York, USA, pp. 282–293 (2010)
12. Weisstein, E.W.: Heaviside step function. From MathWorld – A Wolfram Web Resource (2013), http://mathworld.wolfram.com/HeavisideStepFunction.html
13. Ivaschenko, A., Lednev, A.: Time-based regulation of auctions in P2P outsourcing. In: Proceedings of the 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Atlanta, Georgia, USA, November 17-20, pp. 75–79 (2013)

Short Papers

Allowing agility and mastering dynamics of business processes is addressed by the following contributions.

Matthias Kurz and Matthias Lederer show how Subject-oriented Business Process Management can be used to extend an existing Adaptive Case Management approach.

Davut Çulha and Ali Doğru suggest an agile business process development methodology and an estimation formula to assess its efficiency.

Vadim Agievich and Kirill Skripkin present the matrix of change and discrete optimization methods as means to plan BPM changes.

Udo Kannengiesser discusses how Subject-oriented Business Process Management can support the lean production method of value stream design.

S-BPM Groupware denotes the Software-as-a-Service S-BPM suite. It is introduced by Stephan Borgert and Max Mühlhäuser revealing support of cross company process execution and performance scalability.

In the last paper in this section Thomas Müllerleile and Volker Nissen reflect on the economic impact of process acceptance. They use Grounded Theory to construct a theory of process acceptance from empirical data applying qualitative content analysis.

Subject-Oriented Adaptive Case Management

Extending Subject-Oriented Business Process Management to Knowledge-Intensive Cross-Enterprise Business Processes

Matthias Kurz[1] and Matthias Lederer[2]

[1] DATEV eG, Paumgartnerstr. 6-14, Nürnberg
[2] Universität Erlangen Nürnberg, Lehrstuhl Wirtschaftsinformatik II, Lange Gasse 20, Nürnberg

Abstract. Adaptive Case Management is a Business Process Management approach that is quickly gaining the attention of practitioners and scientists. In an effort to examine how Subject-oriented Business Process Management relates to Adaptive Case Management, this contribution proposes extending an existing ACM approach inspired by multi-agent systems with the capability of defining temporal-logical dependencies between tasks using Subject-oriented Business Process Management.

Keywords: adaptive case management; knowledge work; metamodel; subject-oriented process management; multi-agent systems.

1 Introduction

Classical Business Process Management (BPM) offers a rich and tested set of methods for well-structured and routine processes. The Adaptive Case Management (ACM) concept is an increasingly popular concept which promises to bring the benefits of BPM to the area of weakly structured knowledge-intensive business processes [1, 2]. Contrary to classical BPM systems which focus on automating business processes, ACM provides capabilities to adapt processes during runtime. “This form of runtime flexibility allows process participants to respond to challenges or new requirements that were not considered during designing the business processes.” [3]

However, both classical BPM as well as ACM assume that all process participants work towards the same goal. This assumption is increasingly becoming unsustainable as processes are more and more spanning multiple enterprises. In order to address this issue, [3] extends the ACM-approach of [4] based upon the paradigm of Multi-agent Systems (MAS). [3] proposes a distributed approach for managing and supporting knowledge-intensive cross-enterprise processes (KXBP) as well as a corresponding case metamodel. Although being a first step towards a comprehensive KXBP methodology for ACM, [3] currently focusses on defining and breaking down the work within a case. Therefore, the KXBP methodology offers no way to specify the temporal-logic dependencies of tasks.

This manuscript addresses this gap from a conceptual perspective by using the Subject-oriented Business Process Management (S-BPM) approach as a method for

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 123–132, 2014.

defining the temporal-logical dependencies of a KXBP case. As this Subject-oriented ACM (S-ACM) approach relies on the S-BPM methodology, S-ACM not only extends the ACM-based KXBP approach of [3] but also demonstrates how S-BPM and ACM can be combined. Thus, this contribution demonstrates how S-BPM can be applied in knowledge-intensive cross-company business processes that may be changed while executing these processes.

2 Adaptive Case Management

ACM was first made popular by the well-renown book [1]. [2] complements this book with practice reports and case studies. Although major principles of ACM are outlined in [1], it provides no concrete method for using ACM in a real-world environment. This gap has been closed by the method described in [4] that integrates classical BPM and Case Management with the Enterprise 2.0 paradigm. As the method from [4] serves as the foundation of this contribution, this section presents an excerpt from [4].

In ACM, knowledge workers are no longer expected to follow strictly defined business processes regardless of their suitability for a given problem. Instead, they are empowered and encouraged to adapt the case behavior if necessary. According to [4], the case behavior is described using a case process. Therefore, changing the case behavior means changing a case process. This includes changing the processes of running cases. This kind of runtime flexibility is a key characteristic of ACM. It allows case workers to respond to new challenges which arise after a case has been started.

Each case is represented by a case workspace and is assigned objectives the case is expected to achieve. The workspace contains a process which is constituted by a hierarchy of tasks. Tasks assist in coordinating knowledge work between multiple case participants. [4]

Tasks are not the only object type of a case workspace. Workspaces also contain artifacts like documents or hyperlinks. These artifacts may be added to, removed from, or modified in a case workspace. While automation is not the primary objective of ACM, workflows may be linked to tasks in order to provide automation to those parts of a case that are unlikely to change and, therefore, can be automated efficiently. [4]

Every case object may be created from scratch. For improved efficiency, the method advises to use object libraries for storing and retrieving commonly used case objects. Similarly, case workspaces can be instantiated from predefined templates stored in the template library. This instrument allows to standardize and manage similar cases while retaining a high degree of flexibility for the knowledge worker. [4]

Once several instances (cases) of a case template have been completed, it is advisable to review the respective cases for common changes which should be integrated back into the templates. This adaptation of case processes is a vital instrument for the continuous improvement of case templates. [4]

3 Cross-Enterprise Adaptive Case Management

The ACM method described in section 2 provides little guidance how to support the autonomy of the participants from multiple organizations. This section briefly introduces the KXBP approach described in [3] which supports this kind of autonomy.

With "many competing, mutually inconsistent [definitions]" [5] for the terms *multi-agent systems* and *agents* circulating, the KXBP approach is based on the following definition: "Multi-agent systems are those systems that include multiple autonomous entities with either diverging information or diverging interests, or both" [12]. Therefore, an agent in the context of this contribution is an autonomous knowledge worker who pursues both his or her own interests as well as those of his organization while participating in a case along with other agents from his or her own organizations or other organizations. [3]

When multiple organizations participate in one case, there are essentially two types of case objects. *Common case* objects (cco_c) refer to objectives, tasks, and artifacts that are visible to other organizations involved in a case. While common objectives are used to manage and measure the output of the case, common tasks are used to coordinate the case work. The case output or intermediate results are stored in artifacts like documents. *Private case objects* (pco_c) are used by single organizations ($pco_{o,c}$) or their agents ($pco_{a,c}$) in order to efficiently follow their own agenda within the contractual or legal boundaries and to define the internal behavior of an organization within the case. Private case objects are not visible to members of other organizations. [3]

The different types of case objects constitute perspectives on a case. There are three possible perspective types in a case:

- Every case has a common perspective (COP). The COP is constituted by the set of all common case objects of a case (CCO_c). This perspective contains all common objectives and the basic case structure which is shared by all case participants. [3]
- Each organization may optionally have its private organizational perspective (POP) for every case its members contribute to. POPs introduce elements from the organization's own agenda and proprietary best practices that are useful for achieving the objectives. [3]
- Similar to organizations, agents may optionally have a private agent perspective (PAP) for each case they participate in. [3]

These three perspectives are combined when determining the view for an individual participant. For any given agent a of the organization o, the view $v_{c,a}$ on the case c is calculated as follows:

$$v_{c,a \in membersOf(o)} = CCO_c \cup PCO_{o,c} \cup PCO_{a,c}$$

By adding private case objects like objectives, tasks, or temporary documents to the POP / PAP, private perspectives allow defining behavior that is proprietary to and only visible to the respective organization / agent. Common case objects are used to coordinate cross-organizational case behavior and to define common artifacts like documents. The different perspectives may each reside on separate computers in order to avoid the necessity of a central computer that has to be trusted by all participating organizations. This way, each organization can be sure that only common perspectives leave the control of the organization. [3]

4 Subject-Oriented Adaptive Case Management

4.1 Case Behavior

The following sections extend the KXBP ACM approach summarized in section 3 by leveraging the S-BPM method for defining the case behavior which refers to the temporal-logical dependencies between tasks. Therefore, the resulting approach is referred to as *Subject-oriented ACM* (S-ACM) in this paper.

The case behavior comprises (1) the *proposed case process* and (2) *the executed case process*. The suggested way for achieving the case objectives is defined in the *proposed case process* which provides a temporal-logic relation between the tasks of a case and thereby constitutes the case behavior. Proposed case processes are introduced in section 4.3. As emphasized in section 4.2, agents are free to deviate from the proposed case process whenever deemed applicable. The *executed case process* records which tasks have been executed during the case lifetime. While the proposed case process may be modified before[1] and during the entire lifetime of a case, the executed case process is automatically being created and extended during the lifetime of a case. Section 4.4 discusses executed case processes in more detail.

Tasks are only started by agents or by external events. If an external event occurs, a task associated with the event may be started and processed by an automated program agent.

4.2 Allowing Ad-hoc Changes

Contrary to Workflow Management Systems (WFMS) and BPMN 2.0 ad-hoc sub processes [6], proposed case processes only suggest the order of and the dependencies between tasks. In order to allow for run-time flexibility, agents may deviate from the case process and introduce ad-hoc changes. That way, knowledge workers are empowered to adapt the case to the individual requirements of particular cases which were not known when defining a case template.

While ad-hoc changes are vital for ensuring run-time flexibility, they can lead to deadlock situations if the process model is changed ad-hoc. E.g., the task *wait for approval* will never be completed if the preceding task *approve* is deleted by an ad-hoc change. Deadlocks may happen due to dependencies that are less obvious like input preconditions of tasks that expect a certain process state. If agents are be empowered to make ad-hoc-changes to the proposed case processes without having to consider deadlock situations, tasks cannot assume a specific process state. This can be

1 The proposed case process may be defined in a case template which is used for instantiating a particular case. Thus, the proposed case process of a given case may be modified / defined before instantiating this case.

achieved by adding a test to each task. Such tests are invoked automatically whenever a task is executed and check whether the respective task can be executed in the current process state. If not, the task execution is stopped with a message explaining the reason.

4.3 Proposed Case Process

There is a vast number of notations and paradigms for defining the temporal-logic dependencies between tasks in a process (e.g., flow-oriented [7], object-oriented [8], and subject-oriented [9]). With ACM relying upon the cooperation of multiple individuals, S-ACM uses the subject-oriented paradigm and notation proposed by [9] which is "[…] a stakeholder- and communication-oriented paradigm that roots in the observation that humans usually use standard semantics of natural language […] when they describe what they are doing in a business process" [10].

Subject-oriented business processes comprise two types of models. (1) *Interaction diagrams* (ID) show the communication between the roles in a process (subjects in S-BPM) of a process. (2) *Subject-behavior diagrams* (SBD) define the sequence of activities within a subject. This concept of distinguishing between the externally observable behavior of a process role (the communication) and the internal behavior of a process role is highly compatible for defining the case behavior, as the case process is typically used for coordinating the work between multiple agents. By focusing on the communication between subjects, the case process allows coordinating the dependencies between the tasks of the participating agents while ensuring the agents may decide how to complete their tasks.

With multiple individual agents potentially being able to play the same role, roles are assigned to subjects in the proposed case process. Using the S-BPM notation makes introducing ad-hoc changes to the parts of the proposed process affecting the own role simpler for agents playing this role, as these agents typically focus on the process behavior of their own role. This maps well with the concept of autonomous agents introduced in section 1.

Fig. 1 shows the corresponding SBD for the involved roles *project manager* and *requirements engineer*. These SBD only contain the tasks (function states in S-BPM) of the roles and the corresponding communication primitives (send and receive states in S-BPM). Assuming the communication between the roles is necessary for ensuring the temporal-logical dependencies between tasks, these figures depict the minimal possible process defining the case behavior.

Introducing proposed case processes as an instrument for specifying the temporal-logical relationships between tasks means that there are two views on the tasks. This has the following implications.

- The task hierarchy and the proposed case process are two perspectives on the same process and not two synchronized yet distinct models. E.g., whenever a task is added or removed in the proposed case process, this change is immediately visible in the task hierarchy as well.

- Tasks may be created without being assigned or proposed to a role / subject. In the proposed case process, these tasks are associated with the role *unknown*.
- Not all tasks of the task hierarchy must occur explicitly in the proposed case process, as they implicitly are part of the respective subject's SBD. These tasks may be executed at the agents' discretion.

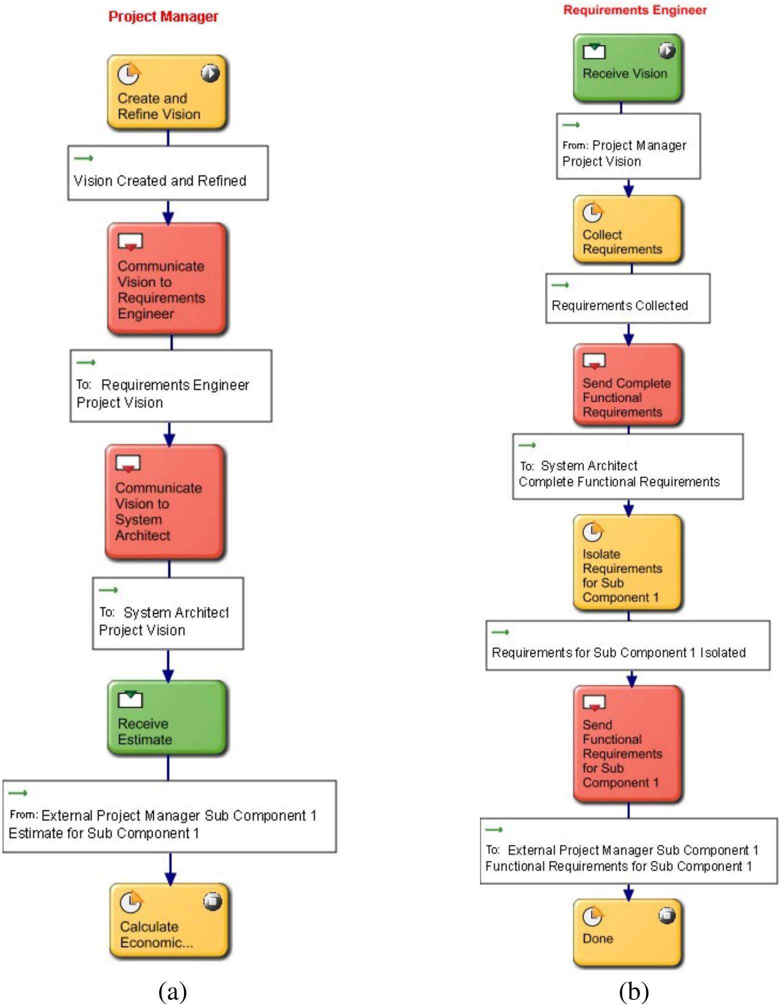


Fig. 1. Proposed case process subject behavior diagram of the roles *Project Manager* and *Requirements Engineer*

4.4 Executed Case Process

The executed case process stores the sequence of tasks which have been executed during a case. It serves three primary purposes:

- During case execution, it represents the current state of the case processes. With the activity stream of the metamodel described in [3] already recording all interactions with a case, the executed case process essentially is a view on the activity stream which filters out all activities that are not directly related to task execution.
- Once a single case has been completed, it serves as an instrument for checking whether the compliance rules have been adhered to. Checking the compliance of tasks is particularly important, as ACM intentionally provides a high degree of flexibility due to the focus on ad-hoc changes. While necessary for knowledge-intensive business processes, this makes violations of compliance rules more likely.
- After multiple cases originating from the same case template have been completed, it is possible to compare the executed case process of the individual cases with the proposed case process of the case template. This allows identifying issues and potential improvements and thereby substantially simplifies continuously improving case templates. [4] refers to this as *cross-case adaptation*.

The executed behavior of a case is amended whenever an agent executes a task. The subject interaction diagram (SID) of Fig. 2 and the SBD of Fig. 3a and Fig. 3b show the executed case process for the proposed case process introduced in section 4.3. As the executed case process contains the executed tasks only, the SBD encompasses no conditional behavior. Similarly, the SID contains only those subjects which already executed tasks. Furthermore, while subjects are assigned to roles in the proposed case process, the agents playing these roles are known in the case process.

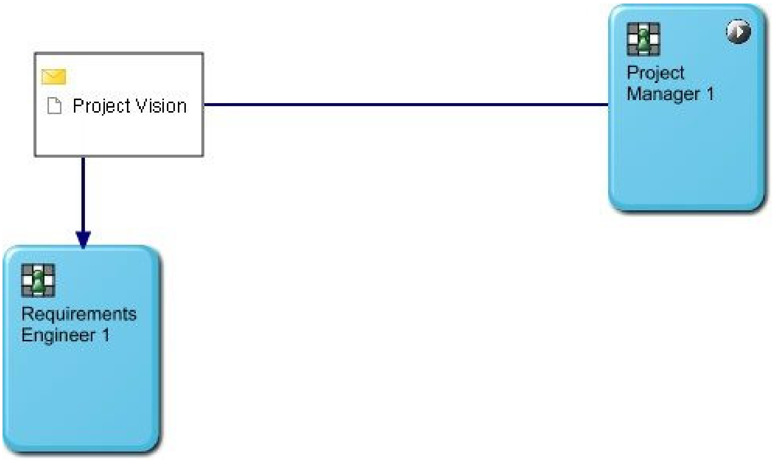


Fig. 2. Executed case process interaction diagram

The task *wait for next step* in Fig. 3a and Fig. 3b is a placeholder which ensures that the SBD is syntactically correct. It indicates that the next action is determined by the agents of the case.

Essentially, the executed case process of S-ACM fulfils a similar function as the ModelAsYouGo approach which allows agents to incrementally build a case process by modifying the internal behavior of subjects representing the respective agents [11].

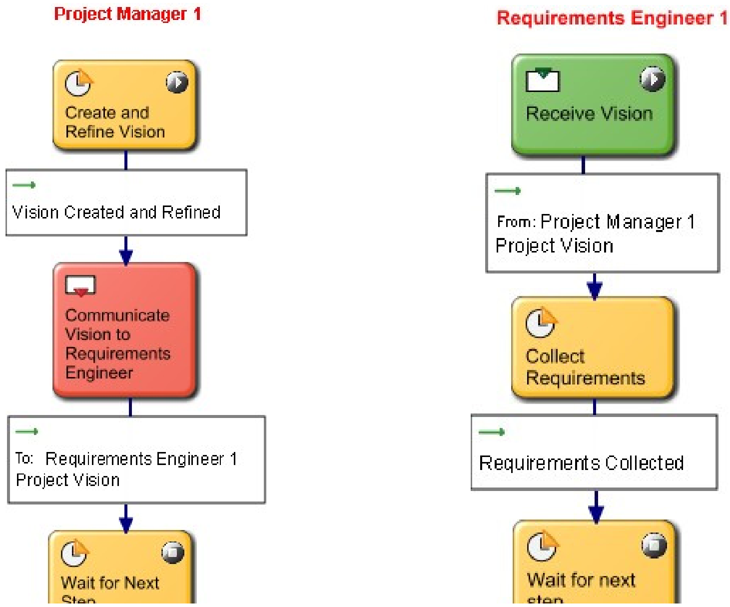


Fig. 3. Executed case process subject behavior diagram of the agents *Project Manager 1* and *Requirements Engineer 1*

4.5 Extended Metamodel

Considering the temporal-logical relationships between tasks in the S-ACM approach requires extending the metamodel described in [3] which provides no means for defining these relationships. Fig. 4 shows how the UML metamodel of [3] needs to be extended. New classes and associations are highlighted with a gray background. The following extensions have been made:

- As external events may start associated tasks of program agents with process the respective tasks (cf. section 4.1), it is necessary to include the *external event* class along with the corresponding relations *associatedWith* and *processedBy*.
- Tasks can have sub tasks refining other tasks. Therefore, the *subTaskOf* association has been added to the *Task* class.
- The proposed case process is added to the metamodel. The *Proposed Case Process* class is associated directly with *Case State*, as (1) associating it with *Phase* would require separate processes for each phase and (2) a case-level process would be required for orchestrating the processes of the individual phases.
- Section 4.3 shows that tasks correspond to S-BPM function states. Therefore, *Proposed Case Process* has an aggregation association with *Task*.

The executed case process is a special view on the activity stream. Therefore, no new elements are introduced to the metamodel for representing the executed case process.

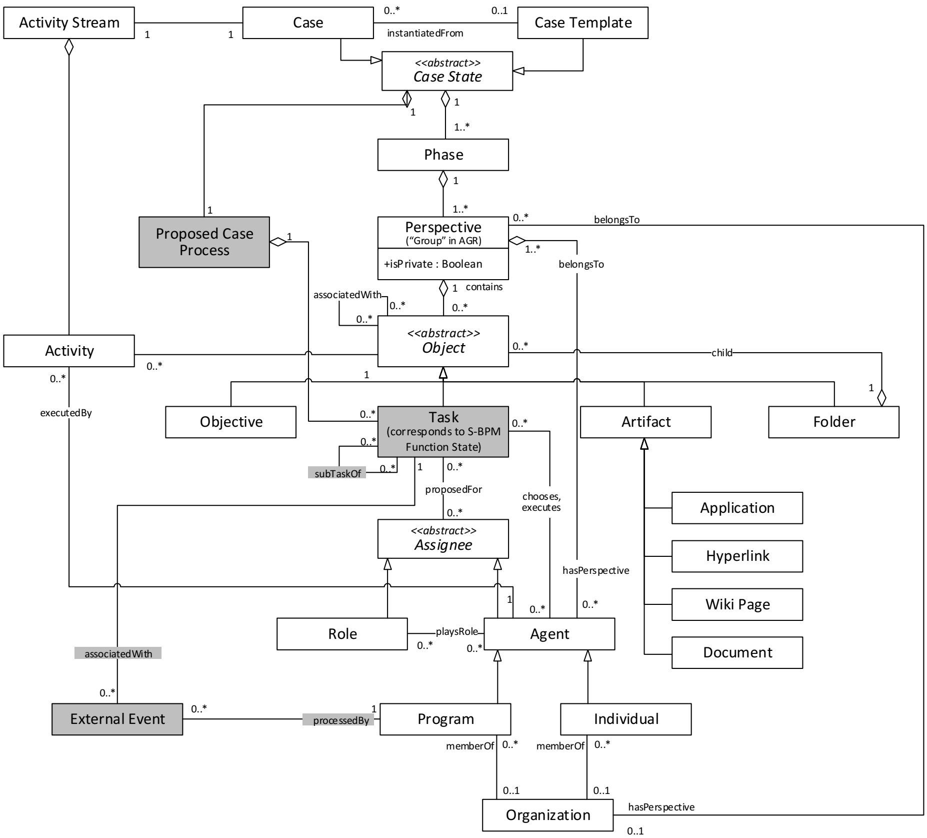


Fig. 4. Extended ACM metamodel

5 Conclusion and Outlook

This paper demonstrates how S-BPM can be applied for defining the temporal-logical dependencies between tasks of knowledge-intensive cross-enterprise processes. S-BPM process models propose an order in which to execute the tasks of the respective case (the proposed case process). In S-ACM, Agents are free to deviate from this proposal at their discretion. With the sequence of executed tasks automatically being recorded in an executed case process, it is possible to check completed cases for compliance with legal requirements or company-specific regulations. Analyzing multiple executed case processes originating from the same case template assists in further improving the proposed case processes embedded in the respective case template.

On the other hand, the proposed approach still has a number of limitations. First, although S-BPM has a graphical notation, this notation needs to be extended in order to address the various aspects of the metamodel. E.g., there currently is no way for visually expressing the relationship between a proposed case process and other case artifacts like documents. The upcoming *Case Management Model And Notation*

(CMMN) of the Object Management Group [11] may give some guidance to the effort of developing such a notation.

Second, a typical operation in ACM is refining tasks by splitting a single task into multiple tasks. While the S-ACM approach provides provisions for this scenario, it is not discussed in this paper due to space constraints. The authors plan to publish an extended version of this publication in the near future. In general the KXBP approach serving as the foundation for S-ACM offers a rich set of concepts that have only been briefly mentioned. In particular, the implications of having multiple perspectives, which may constitute multiple different proposed case processes for different agents, have been covered only briefly due to space constraints.

Finally, a comprehensive set of real-world examples will help analyzing the practicality of the proposed approach. The authors plan to conduct such case studies soon.

References

1. Swenson, K.D. (ed.): Mastering the Unpredictable. How adaptive case management will revolutionize the way that knowledge workers get things done. Meghan-Kiffer Press, Tampa (2010)
2. Layna, F. (ed.): Taming the unpredictable. Real World Adaptive Case Management Future Strategies, London (2011)
3. Kurz, M.: Taming Diversity. A distributed ACM-based approach for cross-enterprise knowledge work. In: Raghavan, V., Hu, X., Liau, C.-J., Treur, J. (eds.) The 2013 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 87–91. IEEE Computer Society (2013)
4. Herrmann, C., Kurz, M.: Adaptive case management. Supporting knowledge intensive processes with IT systems. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 80–97. Springer, Heidelberg (2011)
5. Shoham, Y., Leyton-Brown, K.: Multiagent systems. Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge (2009)
6. Object Management Group: Business Process Model and Notation, BPMN (2011), http://www.omg.org/spec/BPMN/2.0/PDF/
7. Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)" (1992)
8. Dori, D., Goodman, M.: From object-process analysis to object-process design. Annals of Software Engineering 2, 25–50 (1996)
9. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-oriented business process management. Springer, Heidelberg (2012)
10. Fleischmann, A., Schmidt, W., Stary, C.: Subject-oriented BPM = Socially Executable BPM. In: Proceedings of the 15th IEEE Conference on Business Informatics, CBI 2013, pp. 399–406. IEEE Computer Society (2013)
11. Object Management Group: Case Management Model and Notation, CMMN (2013), http://www.omg.org/spec/CMMN/1.0/Beta1/

Towards an Agile Methodology for Business Process Development

Davut Çulha[1] and Ali Doğru[2]

[1] ASELSAN, Ankara, Turkey
[2] Computer Engineering Department, METU, Ankara, Turkey

Abstract. An agile business process development methodology is proposed in this research. An estimation formula is developed in order to assess the efficiency of the new methodology, to yield the required development effort for the traditional case that can be compared to the measured effort with the proposed methodology. Compared with the traditional development of processes, a savings of 27% was achieved. There are currently process development methodologies and limited adaptation work on agile approaches to process redesign. Such existing work do not define a specialized agile methodology for business process development. Existence of many actors renders this field as a complex one where specifying requirements is difficult. Agile approaches may contribute mainly to efficiently gathering desired requirements and may decrease the development time. Also the proposed methodology suggests a critical utilization of training that improves the gathering of quality requirements. Agile requirements gathering, periodic meetings, and incremental and iterative development are observed to be crucial constituents of the proposed methodology during the early studies for applying the methodology to a process in an organization.

Keywords: agile, business process, development methodology.

1 Introduction

A business process is a collection of structured and related steps or activities. Business processes are studied from many different points of view: They are defined, developed, implemented, enacted, configured, and optimized. In other words, management of business processes is required, coining the phrase "Business Process Management" (BPM) [1]. The aim in this article is to propose an agile methodology for developing business processes that is superior to the classical waterfall methodology.

The methodology has been built according to the experiences during the development of business processes using the classical waterfall model. The problems of the classical methodology are identified and an agile approach is proposed to solve them.

Some business processes have been implemented in our organization. They have been implemented through the phases which are analysis, design, and coding. This phased development is actually based on the Waterfall model. During the development it was observed that implementation takes long time and requirements change

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 133–142, 2014.

during "going live". These problems address the need to use agile methodologies with the main expectation of effort saving. The reason is that agile approaches support eliciting better requirements and adapting business processes to the changing conditions.

Some problems occurred in the application of agile methodologies. Business process development is usually more complex than standard software development. Therefore, 2 or 3-week agile iteration durations do not fit business processes. A bit longer periods seem to be more suitable. In addition, a person in the development deals with more than one project at a time where three projects have not been uncommon. Also, daily meetings are time consuming and confusing. Their periods and structure need some arrangements.

The encountered problems demonstrate the need for a specialized development. Since missing requirements can cause reimplementation, gathering of the requirements is very important. Agile approaches should be included to determine the actual requirements of business processes. Also, agile approaches may decrease the development time. In conclusion, applying agile approaches to BPM yielded to develop a specialized agile methodology for business process development.

Related work is presented in the next section. After an introduction to the new methodology, similar work is compared, and application of the methodology is presented. Lastly, conclusions are included.

2 Related Work

A business process is a collection of structured and related steps or activities [10]. There is considerable amount of work on business processes. Surveys on BPM can be found in [1] and [19].

In [11] and [12], a project-oriented development methodology is discussed. The methodology takes an as-is model and tries to optimize it to achieve the to-be model. Another methodology for the business process development relies on web services [13]. In [14] a business process development methodology has been developed that uses UML. Another business process development methodology rests on processes, where they are first class entities [15]. Applications written in BPML [17] will be direct representations of business processes. The work presented in [16] proposes an innovative approach for business process modeling and enactment, which is based on a combination of protocols and policies.

The introduced business process development methodologies are not detailed in the respective papers. Therefore, comparing them was not possible. This situation also addressed that a detailed methodology for business process development is needed.

In recent studies, agile approaches were applied to BPM and social media are used to provide BPM with agility [6] [7] [8] [18]. In all of them, there is a mention of an agile methodology but it is not completely conveyed, except for only a feel of it.

Agile software development methodologies [3] [4] are based on iterative development. Agile methods encourage frequent inspection, adaptation, teamwork, and self-organization to allow for rapid delivery of high-quality software. Scrum [5] is an agile development methodology whose skeleton consists of iterations and daily inspections. Extreme Programming (XP) [4] [9] is another agile software development methodology which advocates frequent releases in short development cycles where new customer requirements can be adopted.

3 The Methodology

3.1 Definition of the Methodology

This methodology is an iterative and agile methodology. It depends on the process framework where there are some framework activities and some umbrella activities. Process framework is explained in [2]. The framework activities form the stages of the methodology. These stages are analysis, design, construction, going to pilot, going to live, and diagnosis. These stages are shown in Figure 1. Transition between stages is not strict and all the stages can be active at the same time with changing densities according to the progress of the development. This is also shown in the figure. From another perspective a stage actually means a set of tasks in this methodology. Therefore, a stage can be considered as a set of related tasks. In addition to the framework activities, there are umbrella activities which are spread over the entire methodology. In other words, umbrella activities are related to nearly all the framework activities. The umbrella activities are project management and training. Also, keeping history of the development is important.

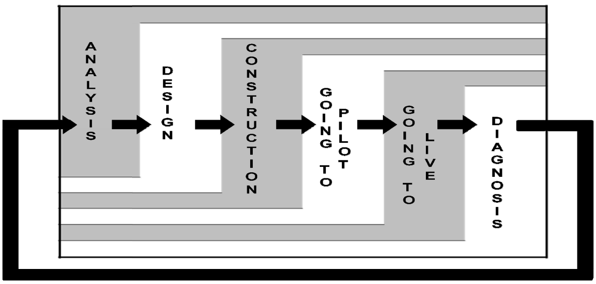


Fig. 1. Stages of the methodology

3.2 Stages of the Methodology

In the analysis stage, requirements are gathered. Use case diagrams are used in this stage. Use cases simply show an actor and its actions. Therefore, working with use cases facilitates determining the roles in a business process.

In order to gather requirements better, site inspection is also included in the analysis stage. Site inspection means examining the business process in its own place. Site inspection also helps to determine the roles of the business process.

It is observed that spike solutions speed up development. Actually they train the stakeholders and support the gathering of better requirements.

In the design stage, process modeling is done using Business Process Diagram (BDP) of Business Process Modeling Notation (BPMN). For fast implementation of the process model, it is assumed that a common screen design is used in all the steps of the process model. Basic input output requirements are determined through this common screen. This screen is referred to as the main input output screen.

In this methodology, any kind of template usage is encouraged because templates can accelerate the development process. In the construction stage, the business process is implemented, verified, and validated. After the pilot stage, live business process starts.

In this methodology, going to pilot is introduced as a stage, because it is observed that pilot application is very important for developing processes. Basically, it provides a preparation for live and increases the success of the implementation. Moreover, the diagnosis stage is also included in this methodology because the diagnosis stage is required in business process management lifecycle [1].

3.3 The Process Model of the Methodology

The process model of the methodology is seen in Figure 2. The model shows the 6 stages of the methodology which are in interaction with the iterations of the methodology. Also, umbrella activities are shown which are spread over the entire methodology.

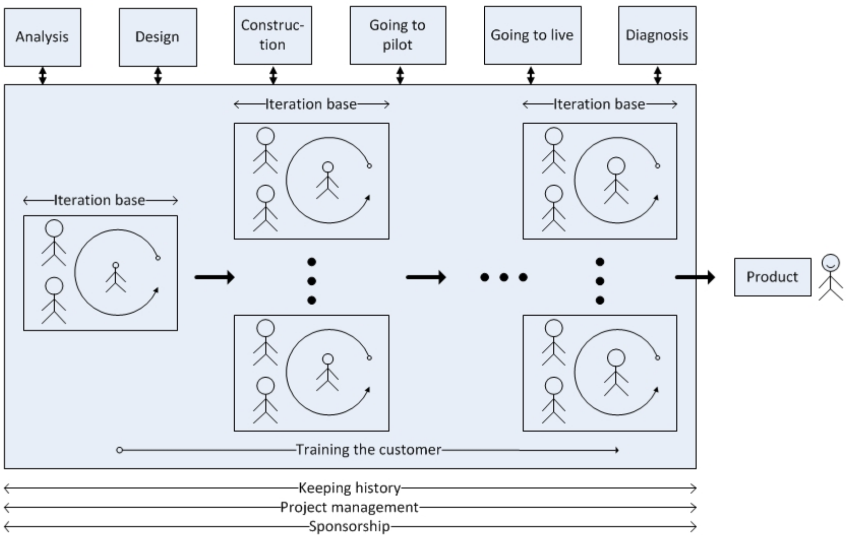


Fig. 2. Process model of the methodology

This methodology requires sponsorship. Usually many organizational units are included in business process development and there are many stakeholders. This increases the complexity of the project and conflicts should be solved by the sponsor when needed.

In this methodology, there are umbrella activities which are light project management, keeping history, and training. Training means training the customer. The more the customer is trained the better the business process is understood. Training is emphasized in this methodology because of the following important reasons:

- Training improves the gathering of quality requirements.
- Training informs people about the process and decrease the development time.
- Training reminds of the decisions taken before, and people do not need to re-solve problems.

3.4 Iteration Base

In this methodology, the 'iteration base' is defined that accounts for incremental development. All the things are done in iteration bases. Figure 3 shows the structure of an iteration base. An iteration base is usually a 5-week period and yields an increment: the process is evolved and something is added to it. In agile methodologies, similar task sets are implemented in approximately in a month. For example, in Extreme Programming a story is implemented in up to a 3-week period [9]. In the Scrum model, a sprint is implemented in 30 days [5]. However, according to the experiences a slightly longer period is more suitable because there are more actors in business processes than in comparable software projects. In short, a period of 5 weeks is chosen for a typical iteration base.

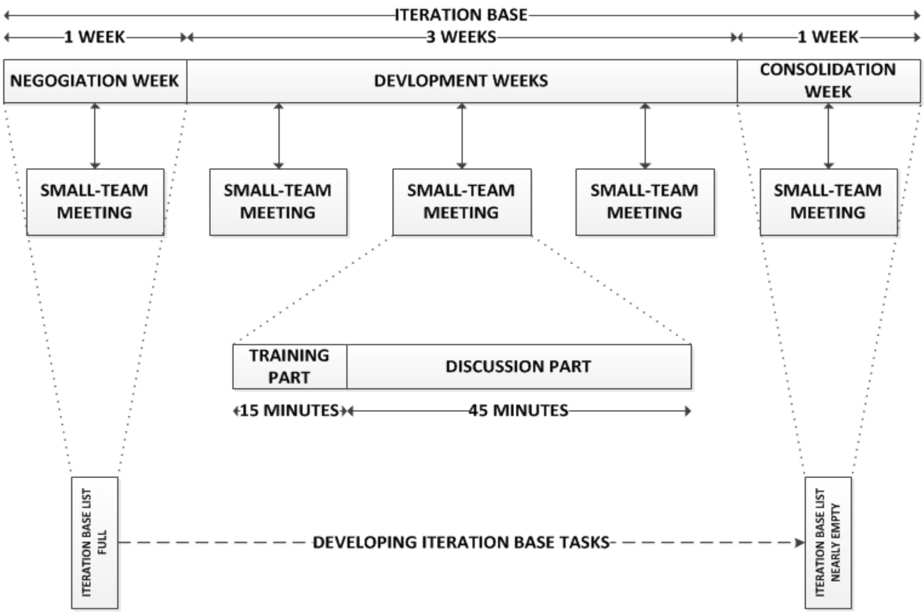


Fig. 3. Structure of an iteration base

A list of requirements is kept like it is done in Scrum [5]. A subset of the list is implemented, which is called an iteration base list, for each iteration base. All the tasks of a project form the project task list. The remaining tasks of the project form the backlog list.

The first week of an iteration base is called the negotiation week. This week of iteration base is used to negotiate with the customer, and prioritize the backlog list and determine the tasks of the iteration base, the iteration base list. The last week of the iteration base is called the consolidation week. In the consolidation week, the iteration base is reviewed and the remaining tasks of the iteration base list are determined and they are added to the backlog list. The intermediate weeks of an iteration base are called development weeks and the tasks of the iteration base list are conducted in these weeks.

An iteration base is usually completed by a small team. Weekly meetings are recommended for small teams. These meetings are called small-team meetings. More than one iteration base can be activated at the same time. Therefore, a person would attend more than one small-team meeting in a week. According to the experiences, more than three meetings in a week are excessive and decrease the development efficiency. For this reason, the project manager should not activate more than three iteration bases related with the same person.

Small-team meetings should be short, i.e., they may take one hour. Small-team meetings are conducted in two parts. The first part is for training and lasts approximately 15 minutes. The remaining is the discussion part.

In the training parts of the meetings, people are informed about the business processes. Especially, the process being developed is introduced to them. If the process has not been developed enough, then the spike solutions of the process can be shown. If neither is available, then other processes can be demonstrated. In the training interval the decisions taken before are repeated also. The aim of this training is to reduce the total development time.

There are also whole-team meetings. Nearly all the stakeholders attend these meetings. These meetings are also structured into training and discussion parts. The whole-team meetings are usually two times longer than small-team meetings. These meetings should be organized when consolidation is needed or when the project needs a refreshment, i.e. drawing attention to the project again.

This methodology is a light weight methodology [21]. Therefore, the documentation requirements should be kept at minimum. However, for each iteration base a simple documentation is good to track the task list of the related iteration base. All the related and required information about an iteration base should be written to this simple document. For the general and required information, again simple but a general document is kept. This document is used to record the interactions and the relations among iteration bases. Moreover, it keeps the history of the development. In other words, every item is added to this document with a date. Therefore, this document glues all the iteration bases in the development.

4 Comparison of Similar Methodologies

The proposed methodology has some similarities with the Rational Unified Process (RUP). First of all, they are both iterative and incremental. Moreover, both of them employ use case diagrams for analysis. However, RUP has strict phases and it is a heavy weight methodology [21]. RUP requires heavy documentation whereas the proposed methodology has light documentation requirements and it is a light weight methodology. In RUP, project management is very important but in the proposed methodology a light project management is included. In RUP, risk mitigation is crucial whereas in the proposed one there is no equivalent.

The proposed methodology is basically an agile methodology. It includes frequent communication, customer involvement, customer satisfaction, short meetings, frequent inspections, product backlog, prioritizing of requirements, cross-functional small teams, self-organization, self-management, self motivation, retrospectives, frequent releases, adaptation, and spike solutions. In addition to these agile approaches, the proposed methodology includes training the customer, keeping history, template usage, main input output screen, site inspection, pilot application stage, and sponsorship.

5 Application of the Proposed Methodology

A new business process has been implemented through the proposed agile methodology and the development effort was recorded. Moreover, nine business processes have been developed using the Waterfall model. Table 1 shows all these implemented business processes.

Table 1. Implemented business processes

Process Name	Applied Methodology	# of Steps(s)	# of Complex Steps(s′)	Effort (person-day)
Consumable Goods Request	Proposed	3	0	34
Purchase Requisition	Waterfall	14	0	130
Insurance Claim	Waterfall	4	0	51
Material Request	Waterfall	5	4	102
Purchase Order	Waterfall	12	0	101
Duty Order	Waterfall	26	0	204
Quality Notification	Waterfall	11	5	156
Quality Tasks	Waterfall	4	2	61
Shipment	Waterfall	3	1	59
Plane Ticket	Waterfall	3	1	52

The implementations are realized in a medium-sized organization. For this study, the guidelines in [20] are used. The main business sector of the organization is electronics. However, these business processes are not related to its business sector. These are general business processes which are related to purchasing, insurance, material request, duty order, quality control, and shipment. These are implemented usually by two analysts and two developers in cooperation with the customer.

In order to compare the efficiency of the proposed methodology, an effort estimation formula was developed. The formula takes the size of the business process and gives the required effort for it development using the Waterfall methodology. To avoid biasing, a simple formula was chosen. The formula is actually a line equation and it is shown in (1). The parameter "x" is the size of the business process and the constants "a" and "b" was found by linear regression.

$$\text{Effort} = a*x + b \tag{1}$$

The number of steps in a business process is simply the best indicator for the size of the business process. Therefore, the number of steps is taken as an indicator for the size of the business process. Moreover, the complexity of a business process also affects the total effort very much. Hence, the number of complex steps is also taken into account. A step in a business process is considered complex if the step requires user interactions except simple user decisions. The sum of the number of steps and the number of complex steps in a business process is taken as the size of the business process.

Table 1 shows also the number of steps and the number of complex steps in the processes. The last column shows the total implementation effort in person-days. According to the table a simple estimation formula is derived using the least squares approach. The sum of the number of steps and the number of complex steps is taken as an explanatory variable of the simple linear regression (1). The values of Waterfall methodology are used in the least squares estimation. Equation (2) shows the estimated formula where s and s′ correspond to number of simple and complex steps, respectively.

$$\text{Effort} = 7.13*(s+ s') + 26.54 \text{ person-day} \tag{2}$$

According to the formula the new process would be implemented in 47 person days using the classical Waterfall methodology. However, this process is implemented in 34 person days using the proposed methodology. This says that there is an effort saving through the proposed methodology. The proportion of the effort saving is calculated in Equation (3) where approximately a 27% effort saving is demonstrated.

$$\text{Effort Saving:} \quad 1 - 34 / 47 = 0.27 = 27\% \tag{3}$$

6 Conclusion

This agile business process development methodology is defined for business processes and stakeholders dealing with more than one project concurrently. It is iterative and incremental. Customer involvement is realized and new requirements are gathered in short development cycles called iteration bases. The project task list is prioritized frequently to queue up the most valuable requirements for iteration bases.

While spike solutions increase the communication between the stakeholders, small teams ensure frequent communications through periodic meetings.

In this agile business process development methodology, sponsorship, lightweight project management, use case diagrams, site inspection, BPD diagrams, template usage, and main input output screen were employed. This specialized methodology gives great emphasis on training, keeping history, going to pilot stage, and determining the roles. Especially training is important because it creates awareness about business processes so that it supports the success of the development.

A formula is derived for estimating the effort required in the development of a business process based on the traditional approaches. The actual effort used in the development of a new business process is compared to the estimated effort if it was developed using the traditional approach. The proposed method showed a 27% effort saving.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Pressman, R.: Software Engineering A Practitioner's Approach, 6th edn. McGraw Hill, New York (2005)
3. Agile Manifesto, `http://agilemanifesto.org/principles.html` (accessed January 12, 2012)
4. Lindstrom, L., Jeffries, R.: Extreme Programming and Agile Software Development Methodologies. Information Systems Management, 41–52 (2004) (summer)
5. Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Washington (2004)
6. Larson, S.: Applying Agile Software Development Methodologies to Business Process Redesign/Management (BPRM). In: Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, March 26-27 (2010)
7. Silva, A.R., Meziani, R., Magalhães, R., Martinho, D., Aguiar, A., Flores, N.: AGILIPO: Embedding Social Software Features into Business Process Tools. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 219–230. Springer, Heidelberg (2010)
8. Bruno, G., et al.: Key Challenges for Enabling Agile BPM with Social Software. Journal of Software Maintenance and Evolution: Research and Practice 23, 297–326 (2011)
9. Extreme Programming, `http://www.extremeprogramming.org/` (accessed January 18, 2012)
10. van der Aalst, W.M.P.: Three Good Reasons for Using a Petri-net-based Workflow Management System. In: IPIC 1996, Cambridge, Massachusetts, pp. 179–201 (November 1996)
11. Weske, M., Goesmann, T., Holten, R., Striemer, R.: Analysing, Modelling and Improving Workflow Application Development Processes. Software Process: Improvement and Practice 6(1), 35–46 (2001)
12. Weske, M.: Business Process Management Concepts, Languages, Architectures, doi:10.1007/978-3-540-73522-9_8
13. Papazoglou, M.P., van den Heuvel, W.: Business Process Development Life Cycle Methodology. Communications of the ACM 50(10) (October 2007)

14. Yamamoto, R., Yamamoto, K., Ohashi, K., Inomata, J.: Development of a Business Process Modeling Methodology and a Tool for Sharing Business Processes. In: Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC 2005). IEEE (2005), doi:0-7695-2465-6/05
15. Smith, H.: Business Process Management—the Third Wave: Business Process Modelling Language (BPML) and its pi-calculus Foundations. Information and Software Technology 45(15), 1065–1069 (2003)
16. Desai, N., Mallya, A.U., Chopra, A.K., Singh, M.P.: Processes = Protocols + Policies: A Methodology for Business Process Development (2004). In: Proceedings of the 14th International World Wide Web Conference, WWW 2005 (2005)
17. Arkin, A., et al.: Business Process Modeling Language (BPML), Version 1.0 (2002)
18. Schatten, A., Schiefer, J.: Agile Business Process Management with Sense and Respond. In: IEEE International Conference on e-Business Engineering (ICEBE 2007), pp. 319–322 (2007)
19. Møller, C., Maack, C.J., Tan, R.D.: What is Business Process Management: A Two Stage Literature Review of an Emerging Field. In: Research and Practical Issues of Enterprise Information Systems II. IFIP International Federation for Information Processing, vol. 254, 1, pp. 19–31 (2007)
20. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), 131–164 (2009)
21. Khan, A.I., Qurashi, R.J., Khan, U.A.: A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies. IJCSI International Journal of Computer Science Issues 8(4(2)), 1694–814 (July 2011), http://www.IJCSI.org

BPM Change Planning Using the Matrix of Change and Discrete Optimization Methods

Vadim Agievich[1] and Kirill Skripkin[2]

[1] National Research University Higher School of Economics, Moscow, Russia
vagievich@hse.ru
[2] Lomonosov Moscow State University, Moscow, Russia

Abstract. Every process-improvement initiative involves some kind of change. But the more complicated the efforts are the more they need tools for managing the change process. A lack of change management tools has been recognized as a substantial reason of low success rates of business process reengineering efforts. The Matrix of Change can help managers identify critical interactions among processes and deal with issues such as how quickly the change should proceed, the order in which changes should take place, whether to start at a new site, and whether the proposed systems are stable and coherent. But one of the disadvantages of the Matrix is its size limitation. The authors describe a way of overcoming the limitation, introduce a formal model of the matrix and formulate the problem of BPM change planning as a discrete optimization problem within the model.

Keywords: BPM, change management, planning, matrix of change.

1 Introduction

The need to consider interconnections during change planning in the enterprise is confirmed by multiple studies. As early as 1990 they were the central subject-matter of the study of the theory of complementary assets of P. Milgrom and J. Roberts [1].

According to [1] complementarity leads to formation of predictable relations between individual types of activity. Relations of complementarity between the changes of technology, demand as well as the structure and scales of an enterprise for the entire XX century kept on creating positive relation between them. Milgrom and Roberts give the following definition of complementarity: “Assets or activities are mutually complementary if the marginal return of an activity increases in the level of the other activity. In other words, if doing (more of) the activity x, the marginal benefits of doing (more of) the complementary activity y increases”.

In the studies of Eric Brynjolfsson and more recent sources instead of the notion “asset” notions “practice” or “organizational practice” are used, defined as a definite way of solving the task an organization has to solve [2]. In the case of BPM change management practices are business processes themselves as well as process groups, goals, principles or other business factors that influence business processes.

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 143–150, 2014.

Effective change management depends on recognizing complements among technology, practice, and strategy. In developing a theory of complements, Milgrom and Roberts showed mathematically how interactions make it impossible to successfully implement a new complex system in a fully decentralized fashion [3]. Instead managers must plan a strategy that coordinates the interactions among all the components of a business system [4]. The task of planning a change strategy considering these interactions is the major function of the Matrix of Change.

2 The Matrix of Change

As the authors of [5] note, the Matrix of Change is the only model which solves the problem of describing complementarities between practices. The Matrix was suggested by E.Brynjolfsson et al. in the article "The Matrix of Change" [4] (Fig. 1). It was developed on the ground of the theory of complements [6] and the concept of the House of Quality [7].

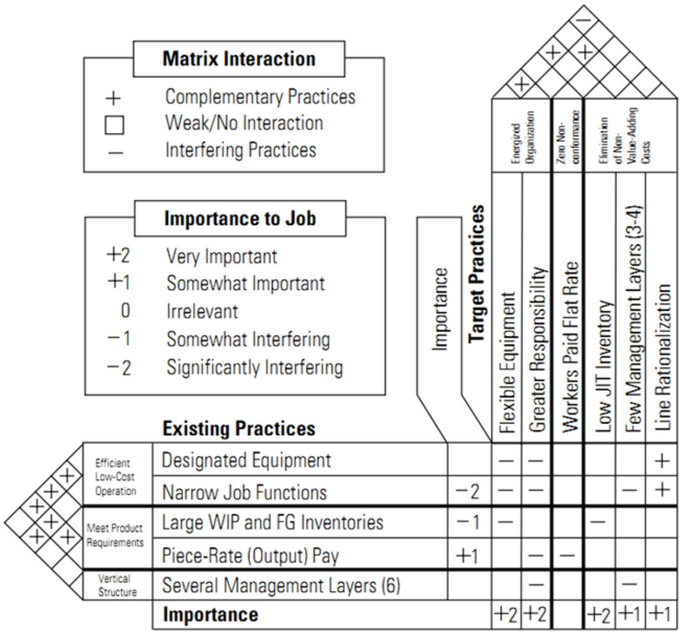


Fig. 1. The Matrix of Change

The Matrix is composed of two interlaid tables. Each consists of a rectangular part – the list of organizational practices and a triangular one, containing data on interactions between the practices. The sign "+" in the cells of the triangle means the complementarity of two practices, the sign "–" means that these practices act in relation to each other as competitors. The horizontal table describes existing practices, the vertical one – the practices that are to be implemented. These complementarities of organizational practices are filled in using expert assessments of the organization's employees. The line and the column "Importance" describe significance of implemented practices under the Likert Scale (from -2 – significantly interfering to +2 – very important). The rectangular at the intercrossing of these two tables specifies the combinability of existing and implemented practices and, respectively, the difficulties of transition from "as is" to "to be".

In order to fill in "as-is" and "to-be" tables managers should first list their goals, business practices, and ways of creating value for customers and then break current practices into constituent processes [4].

Armed with this knowledge of reinforcing and interfering processes, a change agent can use intuitive principles to seek points of leverage and design a smoother transition. The Matrix of Change is a useful tool to answer the following types of questions [4]:

1. Feasibility: Does the set of practices representing the goal state constitute a coherent and stable system? Is our current set of practices coherent and stable? Is the transition likely to be difficult?
2. Sequence of Execution: Where should change begin? How does the sequence of change affect success? Are there reasonable stopping points?
3. Location: Are we better off instituting the new system in a greenfield site or can we reorganize the existing location at a reasonable cost?
4. Pace and Nature of Change: Should the change be slow or fast? Incremental or radical? Which groups of practices, if any, must be changed at the same time?
5. Stakeholder Evaluations: Have we considered the insights from all stakeholders? Have we overlooked any important practices or interactions? What are the greatest sources of value?

3 Requirements and Objectives

The authors of the paper are concentrated on the second type of questions listed above - Sequence of Change. The approach to change sequencing using the Matrix is described in [4] in the form of recommendations. The main recommendations are the following:

- The most easily eliminated practices are those that oppose other existing practices.
- The most easily implemented practices to are those that complement existing ways of doing business (i.e. complement other existing practices).
- Strengthening the old system by new practices in ways that make dismantling the old regime even harder should be avoided.
- The larger the blocks of reinforcing processes, the more difficult they are to change.
- The hardest changes involve the installation of new practices that oppose the greatest number of existing practices. In fact, large new blocks may be impossible to install before the opposing practices are removed.
- In the ideal case, completely independent blocks may be identified and removed separately.

These recommendations may be easily used for small Matrices. But the larger the Matrix the harder its application in practice becomes. Despite the simplicity and efficiency of the approach comprised in the tool, its use in scale change projects is complicated with a large scope of data. Experience shows that one can use the Matrix intuitively only if it does not exceed the size of about 10x10 because the authors of [4] do not propose a ready-to-use formal method or algorithm for the Matrix of Change. Is it possible to build such a method?

Interactions between a practice to be changed and other practices let us judge about the easiness of the change. The recommendations above show that the easiness of separate changes is the main characteristic of the change plan developed using the Matrix of Change. The idea should be used to build a formal method of BPM change planning using Matrices of Change in case of large scale projects.

The objectives of the research may be formulated as follows:

1. Propose a mathematical model containing the data of the Matrix of Change.
2. Develop a formal method of building the best change sequence (i.e. plan of eliminating "as-is" practices and implementing "to-be" ones) according to the interactions between practices. Best sequence maximizes the easiness of changes.

4 The Mathematical Model of the Matrix of Change

First of all the Matrix of Change contains two sets of practices: a set of baseline ("as-is") practices $B = \{b_1, \dots, b_n\}$, where n – the number of baseline practices, and a set of target ("to-be") practices $T = \{t_1, \dots, t_n\}$, where m – the number of target practices, Fig. 2. $X = B \cup T = \{x_1, \dots, x_{n+m}\}$ is a set of all practices ("as-is" and "to-be") of the Matrix.

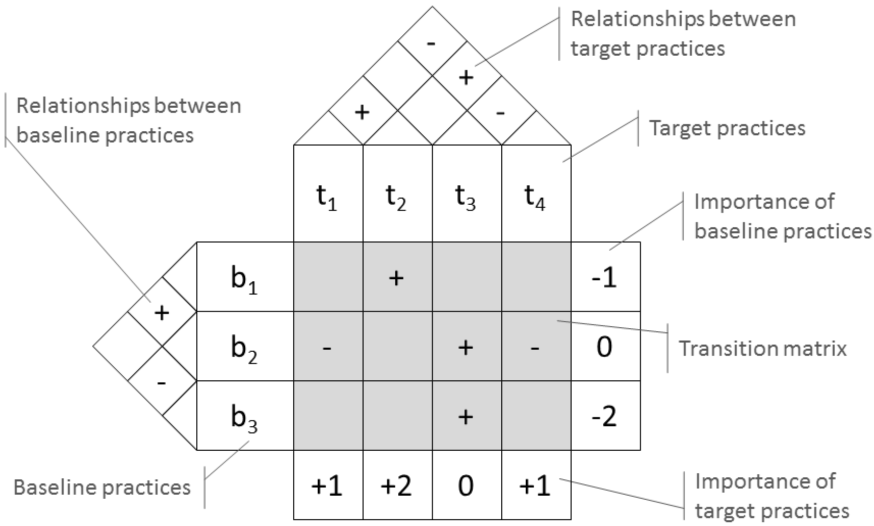


Fig. 2. A mock-up of the Matrix of Change

Let sets V_b and V_t contain the importance of practices of sets B and T accordingly:

$$V_b = \{v_1, \dots, v_n\}, \tag{1}$$

$$V_t = \{v_1, \dots, v_m\}. \tag{2}$$

Brynjolfsson's Matrix of Change does not contain the information about explicit replacement of baseline practices with corresponding target ones. For this purpose the authors propose the use of the scale of so-called "Extended Matrix of Change" described in [5]. Thus possible interactions can be described as a set $R = \{r_1, \dots, r_5\} = \{-2, -1, 0, +1, +2\}$ (the use of "-2" between a baseline and target practice imply explicit replacement).

Interactions between practices can be described as the function

$$r(x_i, x_j) = r_l, \text{ where } r_l \in [1,5]. \tag{3}$$

The Matrix of Change also does not contain the information about desired sequence of some changes that may be needed in practice. This desired sequence can be stated as relationships of partial order for the set X:

$$x_i \leq x_j, \text{ where } i, j \in [1, n+m]. \tag{4}$$

Taking into consideration designations entered above the Matrix of Change can be represented as a weighted undirected painted graph with practices as nodes and interactions as edges. White nodes belong to the "as-is" subgraph G_b and grey nodes belong to the "to-be" subgraph G_t (Fig. 3).

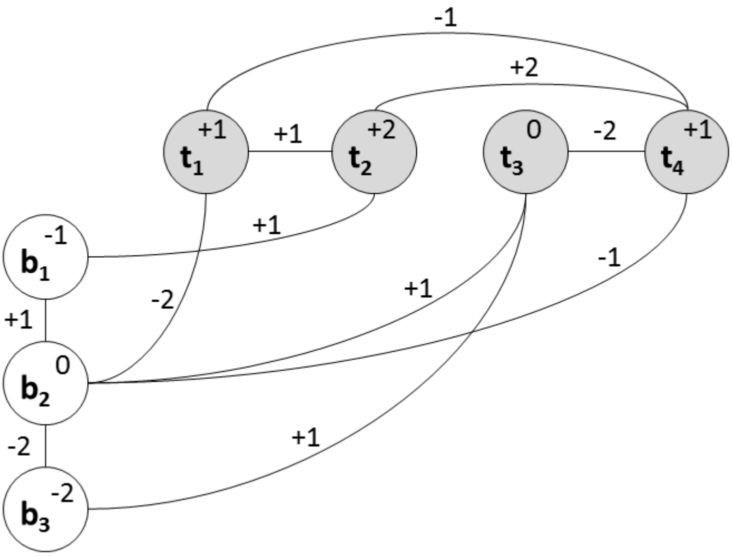


Fig. 3. The Matrix of Change represented as a graph

5 Discrete Optimization Problem Statement

Definition 1. An elementary transformation of graph of practices (or simply "elementary transformation") c_i is implementation of a target practice $t \in T$ or elimination of a baseline practice $b \in B$. The element of the set X corresponding to the elementary transformation c_i is determined by the function $X(c_i)$.

Definition 2. A sequence of elementary transformations is called trajectory $Tr = \{c_1, \ldots, c_{n+m}\}, c_i \prec c_{i+1}$. $Tr(G_b, G_t)$ is a set of all possible trajectories from the graph G_b to the graph G_t.

As it was shown above the objective function should reflect the easiness of changes (or elementary transformations).

The easiness of an elementary transformation c_i is measured in relation to the current state of the system of practices represented by the graph $G_i = (X, V)$. The graph G_i corresponds to G_b in which transformations $c_1..c_{i-1}$ are already made. By this the easiness of an elementary transformation c_i that corresponds to a new target practice can be calculated as follows:

$$l(c_i) = \sum_{j=1}^{n} r(X(c_i), b_j) + \sum_{j=1}^{i-1} \begin{cases} -r\left(X(c_i), X(c_j)\right), \text{if } X(c_j) \in B \\ r\left(X(c_i), X(c_j)\right), \text{if } X(c_j) \in T \end{cases} \qquad (5)$$

The easiness of an elementary transformation for elimination of a baseline practice is the same function but with "minus". The function calculates for an elementary transformation c_i the sum of all interactions between the practice $X(c_i)$ and all practices of the current system of practices G_i (Fig. 4).

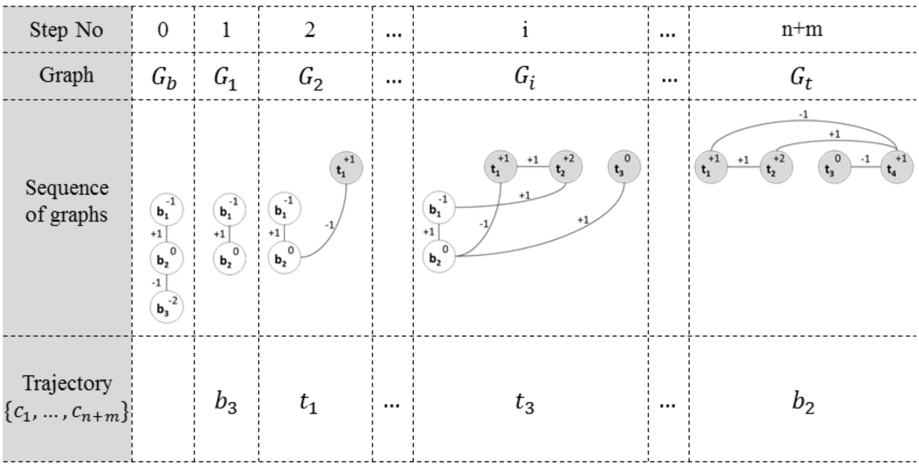


Fig. 4. Transformation of the baseline practices graph into the target one

Consequently, the total rate of the easiness of change for a trajectory Tr_i can be defined as

$$L(Tr_i) = \sum_{j=1}^{n+m} \begin{cases} l(c_j), \text{if } X(c_j) \in B \\ -l(c_j), \text{if } X(c_j) \in T \end{cases} \quad (6)$$

Therefore, taking the formula as the objective function we can formulate the task of finding the best sequence of change for a Matrix as the problem of finding a trajectory $Tr(G_b, G_t)$ for which the total easiness of transformation $L(Tr_i)$ is minimal and minimal easiness of an elementary change $l(c_i)$ is maximal. The latter criterion is introduced in order to eliminate leaps of hardness of an elementary change that can be represented as (-$l(c_i)$).

Let us determine constraints of the optimization task.

As it was shown before some practices have partial order relationships:

$$x_i \leq x_j, \text{ where } i, j \in [1, n+m] \quad (7)$$

Also we should ensure elimination of a baseline practice right before the corresponding target practice if there is a relationship "-2" between them (an explicit replacement). This means that between these two changes there should not be any other changes or for every couple b, t, when $r(b, t) = -2$, there should be a constraint:

$$X(c_i) = b \text{ and } X(c_{i+1}) = t. \quad (8)$$

Eventually the task may be formulated as the following discrete optimization problem:

$$\textbf{Find a trajectory } \boldsymbol{Tr_i \in Tr(G_b, G_t)}\textbf{, where}$$
$$\boldsymbol{L(Tr_i) \rightarrow max,}$$
$$\boldsymbol{min_{1 \leq j \leq n+m}\, l(c_j) \rightarrow max,\ c_j \in Tr_i,}$$
$$\boldsymbol{x_i \leq x_j,\ i, j \in [1, n+m],}$$
$$\boldsymbol{X(c_i) = b\ and\ X(c_{i+1}) = t\ (b, t\colon r(b, t) = -2).}$$

6 Conclusion and Further Work

The authors of the paper develop the approach to business process change management using the Matrix of Change proposed by E.Brynjolfsson in order to overcome the limitations of the tool.

The Matrix of Change is formalized as an undirected weighted graph (practices are nodes and interactions are edges) with two subgraphs (for "as-is" and "to-be" accordingly). The problem of choosing change order is formulated as an optimal "as-is" into "to-be" graph transformation problem. The objective function reflects total easiness of eliminating old and introducing new practices along transformation.

Current work is focused on solution development based on the Branch-and-Bounds method for the problem stated above. In the process of solution development statement of the problem should be complemented with the use of the importance of practices. Further work includes development of corresponding software and detailed description of the method.

References

1. Roberts, J., Milgrom, P.: Economics, Organization, and Management. Prentice-Hall (1992)
2. Skripkin, K.: Organizational Capital of Russian Companies: The Problem of Diversity. In: Innovative Development of the Russian Economy: Regional Diversity: The Sixth International Scientific Conference, Collection of Articles. TEIS, Moscow (2013)
3. Milgrom, P., Roberts, J.: Complementarities and Fit: Strategy, Structure, and Organizational Change in Manufacturing. Journal of Accounting and Economics 19, 197 (1993)
4. Brynjolfsson, E., Renshaw, A., Alstyne, M.: The Matrix of Change. Sloan Management Review 38(2) (1997)
5. Lugachev, M., Skripkin, K., Ananin, V., Zimin, K.: Effectiveness of IT Investments. The Almanach of the Best Works. SoDIT, Moscow (2013)
6. Milgrom, P., Roberts, J.: The Economics of Modern Manufacturing: Technology, Strategy and Organization. The American Economic Review 80(3), 511–528 (1990)
7. Hauser, J., Clausing, D.: The House of Quality. Harvard Business Review 66, 63–73 (1988)

Supporting Value Stream Design Using S-BPM

Udo Kannengiesser

Metasonic AG
Münchner Straße 29 – Hettenshausen, 85276 Pfaffenhofen, Germany
Udo.Kannengiesser@metasonic.de

Abstract. This paper proposes that the lean production method of value stream design (VSD) can be supported by subject-oriented business process management (S-BPM). Among the potential benefits of subject-oriented VSD support are a more accurate and faster data capture and a more effective analysis and enactment of process improvements. The integration of the two approaches is possible because they are based on the same understanding of processes – as a set of asynchronous activities with decentralised control and coordination. The paper describes how some of the fundamental concepts of VSD can be mapped onto S-BPM, including commonly used parameters for analysing existing value streams and some design principles for creating future value streams.

Keywords: Lean production, value stream mapping (VSM), value stream design (VSD), S-BPM.

1 Introduction

Value stream design (VSD) is one of the most widely used methods for identifying and eliminating inefficiencies in production processes [1]. It is based on creating value stream maps for visualising sequences of production steps and a set of metrics such as inventory, transport and waiting times. The data required for populating value stream maps is captured manually, by a lean expert walking along the production line and taking notes on paper. This is a time-consuming process [2, 3] and considers only a snapshot of the process at a particular moment in time. As a result, the data may not be accurate and does not include real-time variations such as changes in customer demand and resource availability [3]. Another drawback of the paper-based VSD approach is that the enactment of process improvements is not directly supported by a process management system. Embedding VSD more systematically in the business process management (BPM) lifecycle has the potential to make lean improvements more effective and sustainable [4].

This paper argues that the overall goal of VSD and other lean production methods – to smoothen the flow of activities within a process – can be supported by the ability of S-BPM to model decentralised process control where individual activities are coordinated asynchronously. This puts S-BPM in a unique position compared to other BPM approaches such as BPMN. Those approaches are based on a view of processes as global, centralised control flows, which is inconsistent with lean thinking.

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 151–160, 2014.

This paper is structured as follows. Section 2 introduces the VSD method and outlines its connection to BPM. Section 3 maps key concepts of VSD, including basic structural elements, lean parameters and design principles, onto modelling constructs used in S-BPM. Section 4 concludes the paper with a discussion of potential benefits.

2 Value Stream Design

2.1 Fundamental Concepts

Identifying, analysing and improving value streams are key activities in lean production [5, 6]. They are most commonly supported through value stream design (VSD) that is a method for identifying and distinguishing value-adding and non-value adding activities [1]. Value-adding activities are commonly defined as those that the customer is willing to pay for [7]. Non-value adding activities are those that create waste and should therefore be reduced or eliminated, where waste is considered anything of the following [8]:

- Overproduction: occurs when more material, parts or products are produced than needed
- Waiting: occurs when an activity cannot proceed until another activity has completed
- Transport: occurs when there is movement of material, work in process (WIP) or finished products, which does not provide direct benefit for the customer
- Overprocessing: occurs when suboptimal machines, tools or product design lead to errors or additional work
- Unnecessary inventory: occurs when there is material, work in process (WIP) or finished products currently not in use
- Unnecessary motion: occurs when human operators, machines or equipment need to be moved unnecessarily
- Defects: can produce scrap and incur costly activities related to correcting the defects

Value stream design is generally viewed as a four-phase process: Stage 1 identifies a product family in which the individual product variants share the same or similar production steps. Stage 2 creates the current-state map, i.e. a diagram showing the as-is value stream including key parameters and lean process metrics such as overall equipment effectiveness (OEE) and total lead time. Stage 3 identifies potential for improvement and creates a future-state map that visualises a re-designed value stream. Stage 4 prepares a work plan for implementing the designed changes in the value stream. The focus of this paper is on stages 2 and 3, as they relate to the key concepts in value stream design and their mapping to S-BPM.

Value stream mapping uses iconic symbols for representing the various aspects of value streams in an easily comprehensible manner. It is usually drawn by pencil on A3 size paper [9]. An example of a (current-state) value stream map is shown in Figure 1.

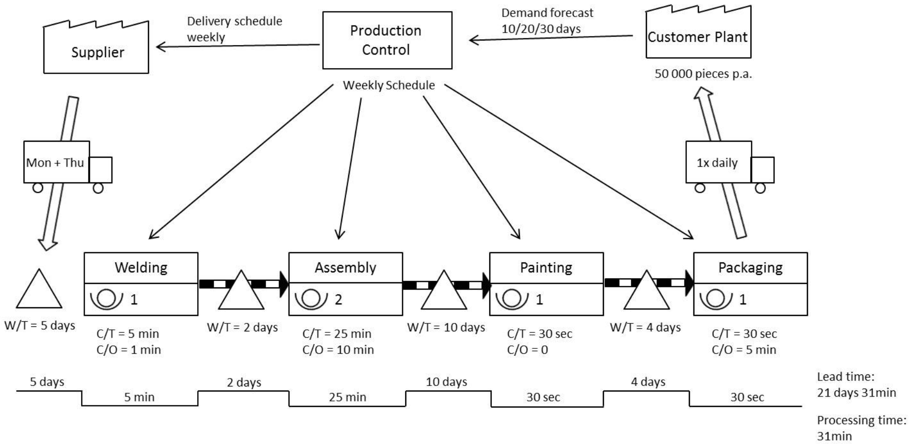


Fig. 1. Example of a value stream map

The overarching goal of value stream design – to streamline the production process – is generally described using the metaphor of "continuous flow". Similar to a smooth and steady stream of water, the continuous flow of production activities implies that within the process there are no or only minimal variations in velocity (i.e., all production steps working at the same speed, otherwise there will be either overproduction/inventory or waiting time) or direction (i.e., no unnecessary or counterproductive steps).

2.2 Decentralised Control

Traditional production lines using centralised planning and control often have difficulties to achieve continuous flow due to variations in customer demand. Decentralised approaches where control is delegated to individual production steps are better suited to react to these variations. There are various techniques such as one-piece flow and pull systems to coordinate the execution of these steps to form a continuous flow. Yet, the paradigm of decentralised control remains fundamental to the lean production philosophy. This has been seen as a major reason why lean production has been poorly supported by IT systems that tend to favour centralised control [10]. These systems are generally too inflexible to accommodate process changes and can thus become obsolete very quickly in lean production systems [3]. The same reason can be assumed for the lack of integration of value stream design and business process management (BPM) systems. There is some work on modelling value streams using BPMN [11]; however, BPMN is still based on the traditional, global control-flow paradigm and does not have well-defined execution semantics.

Support for asynchronous interaction is provided by S-BPM [12], based on the input pool concept that is not available in traditional, control-flow based approaches such as BPMN. An input pool can be viewed as a mailbox for exchanging messages with an active entity (called a "subject") in a process. When a subject is in a function state where it can receive messages (called a “receive state”), it can access its input pool and check for messages. As long as there is no message in the input pool, the subject remains in the receive state. When a message arrives, the subject removes that message from the input pool and follows the transition to the next function state as defined in its Subject Behaviour Diagram (SBD). The input pool can be structured according to behaviour options: The modeller can define how many messages of which type and/or from which sender can be deposited and what the reaction is if these restrictions are violated. The special case in which the maximum size of the input pool is set to zero corresponds to synchronous communication.

The support provided by S-BPM for asynchronous in addition to synchronous interaction has the potential to bring together two seemingly opposing concepts: VSD and BPM.

3 Mapping Value Stream Concepts onto S-BPM

Most concepts of value streams that are to be mapped onto S-BPM can be grouped in three categories: basic structural elements, parameters, and design principles.

3.1 Basic Structural Elements

Erlach [9] identifies six basic structural elements of value streams: production process, business process, material flow, information flow, customer, and supplier.

The *production process* includes all activities that transform raw materials, parts and subassemblies into a final product. In S-BPM, the most appropriate modelling construct for these activities is the subject. This is because subjects allow for the asynchronous behaviours that are the focus of analysis in value stream mapping. The overall production process can then be represented using a subject interaction diagram (SID) that connects the individual activities (i.e. the subjects) with one another via messages. As value stream maps are often concerned with coarse-grained activities [9], the more detailed behaviours of individual subjects (represented in SBDs) are typically not needed for visualising the value stream.

The *business process* includes activities dealing with order processing and production planning and control. They can be represented as subjects and their interactions, within the same SID as used for the production process.

The *material flow* represents the movement, handling and storage of raw material, work in process (WIP) and finished products along the value stream. In S-BPM, the material flow can be represented as messages with associated business objects. At first glance, this does not seem intuitive as messages are usually thought of as conveying information rather than material. However, any piece of information needs a “carrier” that may be a physical part or a container with multiple parts. As industry moves towards an internet of things where every product (or part) is tagged with bar codes or RFID tags, the view of physical objects as informational objects becomes increasingly useful and meaningful. Messages can be thought of as containers for one or more parts, each of which is modelled as a business object. The number of business objects associated with a message then corresponds to a production’s batch size.

The *information flow* includes the exchange of information among individual activities of the business process and the production process. This is modelled in S-BPM using messages and business objects, analogous to the representation of material flow. To clearly distinguish messages representing information flow from messages representing material flow, in this paper we will call the former "control messages" and the latter "material messages".

The *customer* and the *supplier* represent the destination of finished products and the source of raw materials, respectively. They may be organisations that encapsulate their own processes and value streams, and are treated as black boxes. This corresponds to the notion of external subjects in S-BPM.

A SID visualising most of these basic structural elements is shown in Figure 2. It is based on the value stream example presented in Figure 1, and can be seen as a subject-oriented representation of it. In this Figure the grouping of the business process and the information flow on the one hand, and of the production process and the material flow on the other hand, has been done for presentational reasons only. In reality, the two types of processes and the two types of flows are hardly possible to separate from one another, especially for future-state value streams where information flows and production flows are tightly interleaved (as will be shown later in this paper).

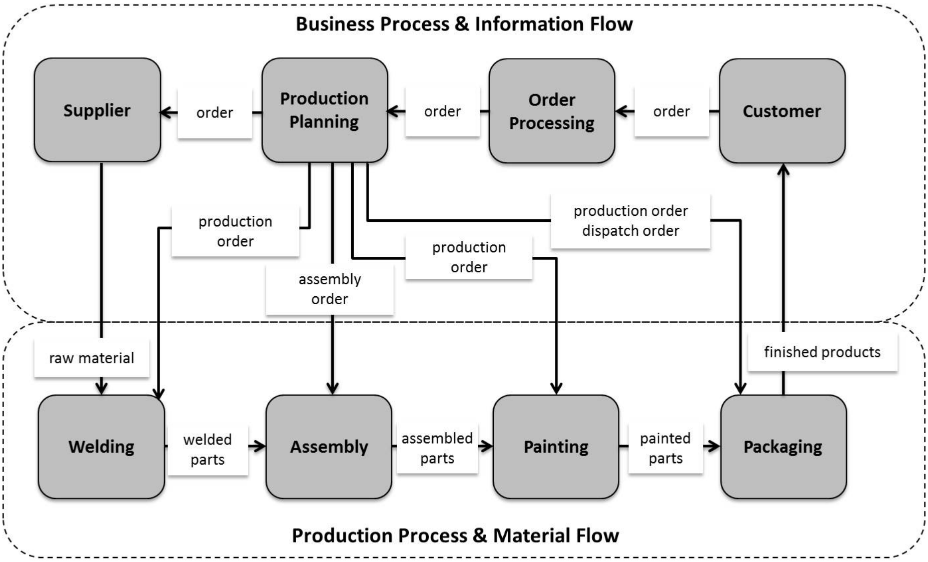


Fig. 2. Subject Interaction Diagram showing the basic structural elements of a value stream

3.2 Parameters

Value streams can be assessed with respect to a number of lean metrics that can be derived from individual parameters. Most approaches to VSD propose the following key parameters: cycle time, changeover time, waiting time, transport time, inventory, rework rate, scrap rate, and number of operators. Most of them can be related to characteristics of individual subject behaviours, as illustrated in Figure 3 for the assembly subject.

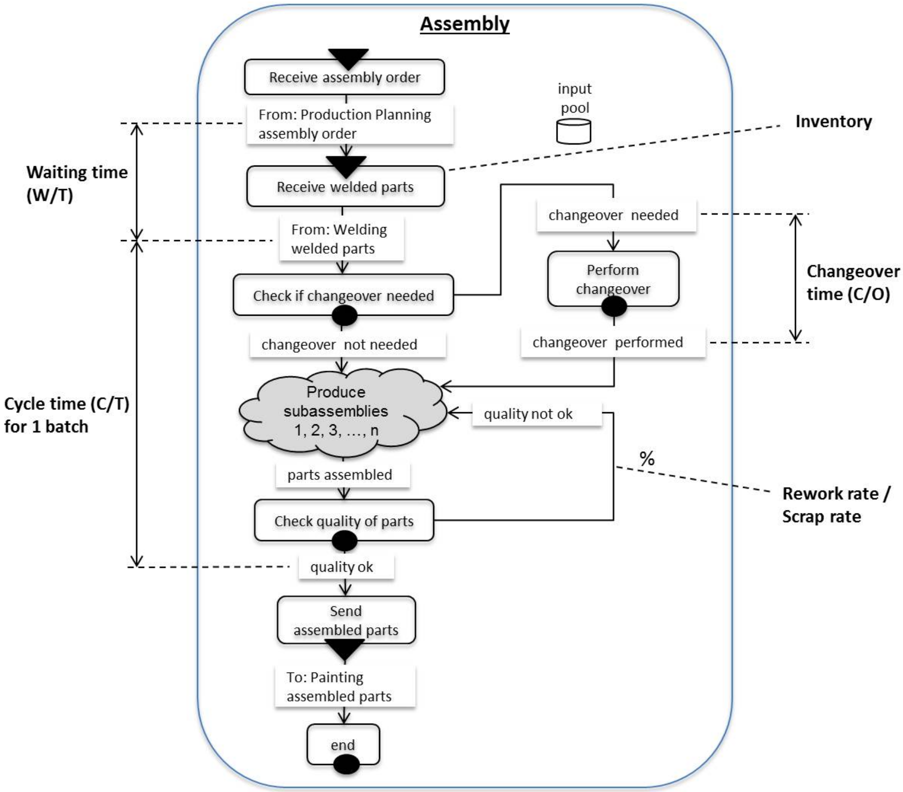


Fig. 3. Value stream parameters derivable from subject behaviour

Cycle time (C/T) is the time it takes to complete a production activity for one unit of the product. Using the SBD, this can be derived from the point in time that a subject has accepted a material message from an upstream subject and the point in time that this subject is ready to send a material message to a downstream subject.

Changeover time (C/O) is the time it takes to perform any necessary preparation (e.g., cleaning and setting up a machine) for switching from producing one product to producing another. This time can be measured from the moment a subject identifies a need for changeover to the moment the changeover is performed.

Waiting time (W/T) is the time that an activity needs to wait for the results of an upstream activity. This time can be measured from the moment a subject commits to executing an activity (by accepting an initial control message such as a production order) to the moment the subject receives the required material message.

Transport time is the time it takes to move raw material, WIP or finished products between two activities in the value stream. Transport can be modelled as a subject that acts as an intermediary between two other subjects in the value stream. The time this subject takes after receiving a material message from the upstream subject until sending it to the downstream subject then corresponds to the transport time.

Inventory represents the number of units of raw material, WIP or finished products stored between activities in the value stream. In S-BPM, this corresponds to the number of business objects associated with material messages in the input pool of a subject.

Rework rate is the relative number (or probability) of defects causing rework. Rework can be represented in a SBD as a loop or exception handling behaviour [12]. Rework rate is then the relative number of business objects for which the loop or exception handling behaviour needs to be executed.

Scrap rate is the percentage of failed WIP or defects that cannot be eliminated through rework. In S-BPM, scrap can be modelled as including those business objects that have been deleted (or, alternatively, marked as "scrap" in a specified field, and sent to a "recycling" subject). Scrap rate can then be calculated as the relative number of deleted business objects with respect to the total number of business objects.

Number of operators represents how many human operators are involved in a production activity. In S-BPM, this corresponds to the number of human agents that have been assigned to a specific subject.

3.3 Design Principles

A number of conceptual design principles have been developed as potential solutions for lean value stream designs [4]. This Section discusses only those principles that have an effect on the process aspects of the value stream and on the way they are represented in S-BPM.

One-piece flow is the closest one can get to realising continuous production flow [9]. This approach aims to reduce batch sizes to just one (work-) piece, each of which is produced in the takt time determined by customer demand. Transport times are reduced to almost zero through a compact and often U-shaped physical layout of workstations. The synchronisation of activities and thus the elimination of waiting times within these production cells are usually achieved by using the same human operator to perform all production activities on the same workpiece. S-BPM can address one-piece flow in two ways: One way is based on the same mappings between value streams and S-BPM as presented earlier, in particular with each activity or workstation corresponding to a different subject. In one-piece flow the same human operator is assigned to all of these subjects. Sending and receiving material messages between these subjects still represents the transport of workpieces from one workstation to another, but without waiting times as the maximum size of all input pools is set to zero (since it is the same operator embodying the sending subject and the receiving subject so there is a "natural" synchronisation). Another way of modelling one-piece flow in S-BPM is based on taking the notion of a production cell as the "integration of formerly separated production processes" [9] literally, by modelling the production cell as one subject. The activities performed at every workstation are then modelled as function states in the subject behaviour diagram of the "production cell" subject.

Supermarkets are buffers that store small quantities of material between production activities. They often have a number of "shelves" for different product variants. The supermarket approach is a pull system that can be used when one-piece flow is not feasible or economic. Supermarkets can be represented as intermediary subjects that receive material from upstream subjects and can distribute this material on request to downstream subjects. Figure 4 shows the example of a supermarket subject that inter-acts with the assembly subject and the painting subject. Shelves can be interpreted as sets of material messages in an input pool that are grouped together according to their message labels indicating different product variants (P1, P2 and P3 in Figure 4). *Kanbans* are part of the supermarket system and represent requests from downstream activities for the production of material by upstream activities. They are modelled as (control) messages in S-BPM. *Withdrawal kanbans* are messages from downstream subjects to a supermarket subject, requesting a specific type of material to be made available from one of the supermarket's shelves. As shown in Figure 5, the supermarket subject responds by taking the requested material message from its input pool and sending it to the downstream subject. If this results in reducing the current number of material items on the respective shelf below a pre-defined minimum, the supermarket subject additionally sends a control message to an upstream subject. This control message represents a *production kanban* that is a request for new material to be produced and sent to the supermarket to replenish the concerned shelf.

Fig. 4. Kanban-based pull system as interactions involving a supermarket as an intermediary subject

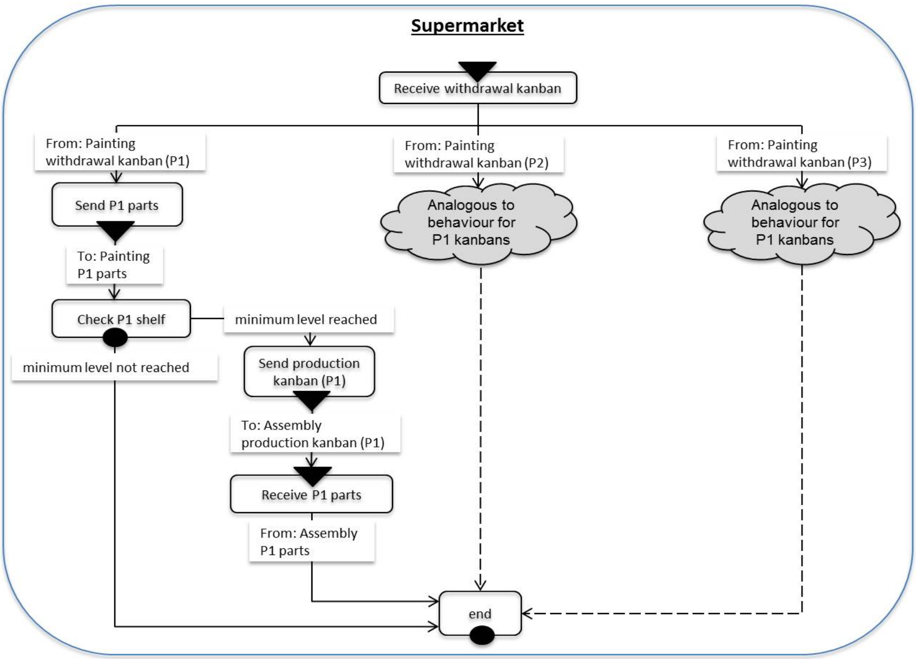


Fig. 5. Subject behaviour of the supermarket

FIFO lanes are small inventories with first-in-first-out (FIFO) processing and a pre-defined maximum of material items to be stored. They can be modelled as input pools in which material messages are time-stamped on entry and then prioritised accordingly. The maximum size of the input pools is specified. When the current number of messages in a pool reaches this specified maximum, any new messages are refused. The sending subject then remains in its sending state until the receiving subject has processed some of the items in the FIFO lane and the new message can be accepted into the input pool.

4 Conclusion

The mapping between VSD and S-BPM brings together two seemingly opposing fields. One is the field of lean production that aims to locally coordinate individual activities to achieve a continuous flow. The other field is BPM that has traditionally focussed on a top-down control of activities. S-BPM, with its view of processes as (potentially asynchronous) interactions between autonomous subjects, opens up the possibility to bridge the two fields and provide benefits to each of them.

Lean production can benefit from subject-oriented process management in two broad areas: capturing as-is processes (or current-state maps) and enacting to-be processes (or future-state maps). As-is processes can be accurately described and analysed when they are executed by a subject-oriented process management system. It reduces the need to rely on manually captured data that can only provide snapshots of a process. Using suitable KPI management systems, value streams can be monitored and analysed for longer timeframes to include the consideration of variations in the data. The design of to-be processes can be supported by subject-oriented simulation, by testing different design alternatives and different production scenarios (e.g. variations in customer demand, product variety, and resource availability). The S-BPM methodology also supports “dry runs” of improved processes prior to their implementation on the shopfloor. These validation runs may also increase workers’ acceptance of process changes and enhance their understanding of lean concepts in general. Finally, the formal underpinnings of S-BPM allow directly enacting process model changes using existing IT standards across the ISA-95 automation pyramid [13].

The benefits of applying lean thinking in business processes have long been recognised; yet, to date this integration has been achieved only on a methodological level [4] due to the conflicting paradigms of lean thinking and traditional BPM approaches. The mappings developed in this paper between VSD and S-BPM can provide the basis for developing computational support for the analysis and design of lean business processes.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme FP7-2013-NMP-ICT-FOF(RTD) under grant agreement n° 609190.

References

1. Rother, M., Shook, J.: Learning to See: Value-Stream Mapping to Create Value and Eliminate Muda. The Lean Enterprise Institute, Cambridge (2003)
2. Hines, P., Rich, N., Bicheno, J., Brunt, D., Taylor, D., Butterworth, C., Sullivan, J.: Value Stream Management. The International Journal of Logistics Management 9, 25–42 (1998)
3. Cottyn, J., Van Landeghem, H., Stockman, K., Derammelaere, S.: A Method to Align a Manufacturing Execution System with Lean Objectives. International Journal of Production Research 49, 4397–4413 (2011)
4. Wagner, K.W., Lindner, A.M.: WPM – Wertstromorientiertes Prozessmanagement. Carl Hanser Verlag, München (2013)
5. Womack, J.P., Jones, D.T.: From Lean Production to the Lean Enterprise. Harvard Business Review 72, 93–103 (1994)
6. Womack, J.P., Jones, D.T.: Lean Thinking: Banish Waste and Create Wealth in your Corporation. Free Press, New York (2003)
7. El-Haik, B., Roy, D.M.: Service Design for Six Sigma: A Road Map for Excellence. John Wiley & Sons, Hoboken (2005)
8. Shingo, S.: A Study of the Toyota Production System from an Industrial Engineering Viewpoint. Productivity Press, New York (1989)
9. Erlach, K.: Value Stream Design: The Way Towards a Lean Factory. Springer, Heidelberg (2013)
10. Powell, D., Binder, A., Arica, E.: MES Support for Lean Production. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds.) Advances in Production Management Systems, Part II. IFIP AICT, vol. 398, pp. 128–135. Springer, Heidelberg (2013)
11. Zor, S., Görlach, K., Leymann, F.: Using BPMN for Modeling Manufacturing Processes. In: Proceedings of the 43rd CIRP Conference on Manufacturing Systems, Vienna, Austria, pp. 1–8 (2010)
12. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management. Springer, Berlin (2012)
13. Müller, H.: Using S-BPM for PLC Code Generation and Extension of Subject-Oriented Methodology to All Layers of Modern Control Systems. In: Stary, C. (ed.) S-BPM ONE 2012. LNBIP, vol. 104, pp. 182–204. Springer, Heidelberg (2012)

A S-BPM Suite for the Execution of Cross Company Subject Oriented Business Processes

Stephan Borgert and Max Mühlhäuser

Technische Universität Darmstadt
Dept. of Computer Science
64289 Darmstadt, Germany
{stephan,max}@tk.informatik.tu-darmstadt.de

Abstract. The execution of cross-company business process models becomes more and more important. This paper introduces a Software as a Service S-BPM suite called S-BPM groupware. It supports cross company execution and is designed for performance scalability.

Keywords: demo, S-BPM, modeling, execution, cross-company.

1 Introduction

The S-BPM Groupware is a S-BPM suite for modeling and executing Subject Oriented Business process models. It supports three important key features needed for modern business process management tools. 1) The suite is a Software as a Service (SaaS) component which offers many advantages for companies. 2) It supports cross company execution of business process models and 3) the actor based backend architecture supports performance scalability. SaaS solutions are normally web based applications which are accessed by web browsers. Many aspects have to taken into consideration for building SaaS solutions like pricing models, distributing option etc. In the present work, we focus on a technical aspects. The S-BPM Groupware consists of three high level components: The frontend, the backend and a repository for behavioral interfaces [6,5]. The behavioral interface repository is in the current state, a simple file storage system with a simple search. These three components form the SaaS application.

This paper is structured as follows. First, the user interface and the technical components of the frontend are introduced. Second, an overview of the backend is given followed by a description of the components. Third the summery is given.

2 Frontend

The frontend is based on modern web technologies and since it is one of the key components, it is explained in more detail. It serves roughly three purposes. Firstly, to model S-BPM processes, secondly, to produce so called behavioral interfaces which can be uploaded to and downloaded from a behavioral interface repository. Finally, it serves as user interface for the process execution.

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 161–170, 2014.

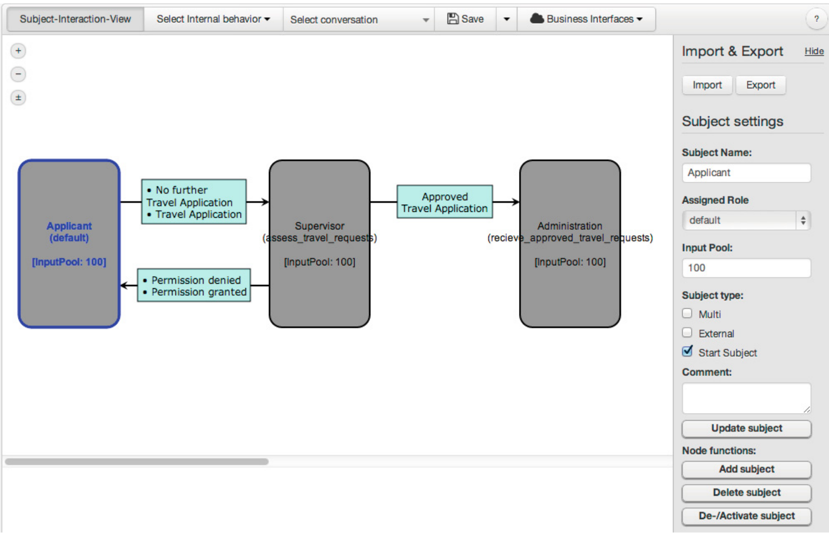


Fig. 1. The subject interaction view of the frontend

The subject interaction view is shown in figure 1. Subjects and their exchanging message types are to bee seen in the center. The view can be switched to the internal behavior view using the button bar at the top. Attributes like name of the subject, inputpool size, etc. can be specified on the right side. The attributes are *Subjectname*, *Assigned Role*, *Inputpool Size* and the subject type. The three supported types are *Multi-Subject*, *External-Subject* and *Start-Subject*. The view can be switched to the view of the internal behavior with the buttons bar at the top.

An example for an internal behavior is given in figure 2. The layout is drawn automatically and can be adapted manually. Beyond the basic actions *Send*, *Receive*, *Action* and *End*, the tool supports a couple of advanced actions like *Macros*, *Observer*, *Modal-Split* and *Modal-Join*. The Observer action is a certain kind of Receive action. The received messages are treated with a higher priority then usual messages. Therefore the control flow of the internal behavior can be moved to a different location. Modal-Split and Modal-Join enable parallel execution of control flow threads and additionally optional paths. The user can decide during runtime, which one is to be executed. The semantics of Modal-Split and Modal-Join is defined in [4].

Two different views are available for the process model execution. A list format view and a graphical view. Examples the views are given in figure 3. The list format view lists the current actions of all process model instances. The colors inform about the status of the states. Green indicates "Action can be processed." Orange indicates "Waiting for an input". The latter is shown every time a message must be received so that the work for this process model instance can be continued. In the graphical view, for each possible action a button is created.

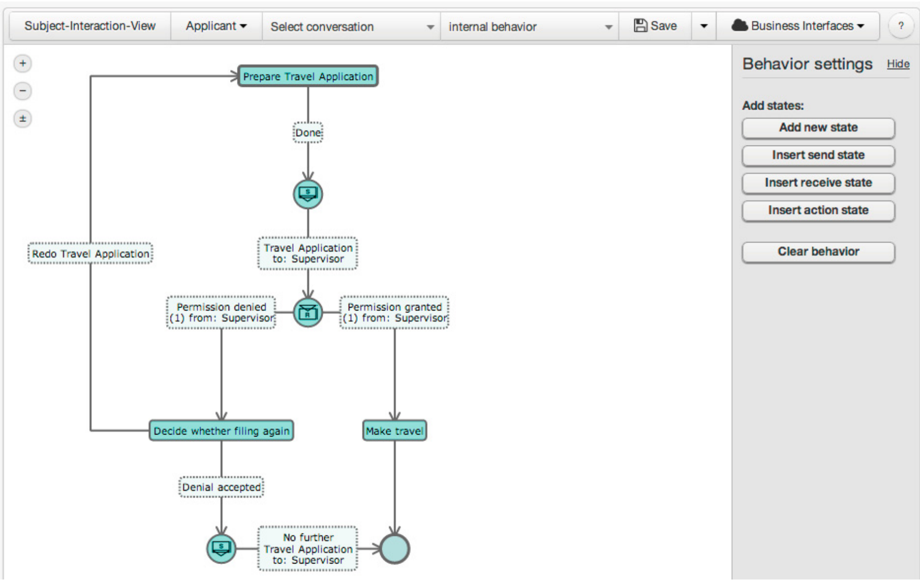


Fig. 2. The view for the internal behaviors of subjects

Behavioral interfaces can also be up- and down-loaded to or from a repository. Up to now, we only support the creation of behavioral interfaces for interface subjects with a 1 to 1 connection to other interfaces.

2.1 Knockout.js

Knockout.js is used to manage dynamic data to be displayed in the browser. To be exact Knockout.js allows a bidirectional link between the data and the visualization of the data in the browser. In addition, dependencies between different values are automatically managed. Besides the using of user input data, the change of programmatic manipulation of data is supported. This makes it easier to view complex data or duplicates in a web page and also ensures that all data always remain synchronous and no dependencies are forgotten in certain situations. We have decided to use Knockout.js because it has a very good performance and it is quiet a small library. It can be used very selectively in web projects but is easy to extend.

2.2 Model.js

The Model.js library developed by us provides a bridge between the program logic on the client side, the display of the data and the server. Model.js takes care of the interim administration and subsequent storage of local data that has been entered by a user or retrieved from the server. This includes for example form data, the current registration state and also the list of the currently

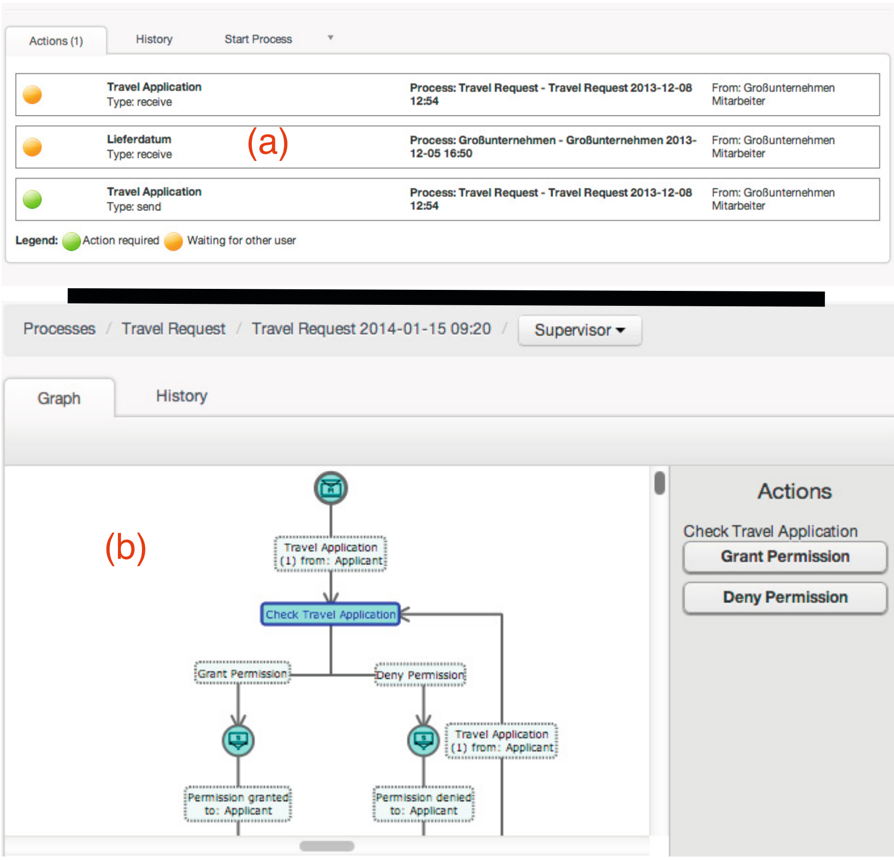


Fig. 3. Two different views are available for the process model execution. A list format view (a) and graphical view (b).

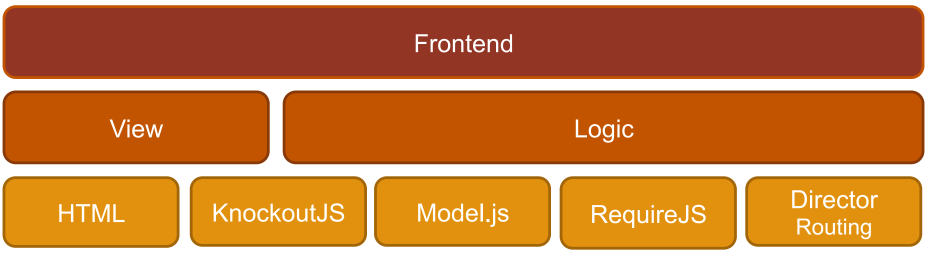


Fig. 4. The frontend architecture

business processes a user is involved and the process state changes concerned. The persistence of the data and states is done by a serialization to JSON and subsequently sending it to the server which then stores it. The behavior of the model library in terms of data storage is roughly based on the Active Record design pattern.

2.3 Require.js

This library is a module loader that attempts to provide a modern module system to Javascript, similar to languages like Java, Python, Scala etc.. The variables and functions of Javascript are usually defined globally. Exceptions are e.g. certain scopes in functions. Furthermore, no notation of public and private variables is available. This leads to name conflicts or misunderstandings about the private and public APIs in larger JS projects. In addition, all required libraries or files must be included in the correct order in an HTML file, which makes the maintenance of several HTML files with different subsets of libraries complicated and difficult.

Require.js solves these problems by grouping JS source code into so-called modules, which are located in separate files. Dependencies to other libraries must be specified explicitly in each module. Public functions and variables must be declared in the same manner. This establishes a clear separation of internal and external API implementations. Thus, only one JavaScript file must be included in the HTML files and all other dependencies of the program are automatically detected and loaded on demand. Require.js also enables the compilation of interdependent code parts into a single file with subsequent compression. Therefore, only a single, compressed file must be delivered to the browser resulting in shorter loading times.

2.4 Director Routing

Director Routing monitors changes of the address bar of the browser and can invoke a particular action based on the given path. This enables the creation of so-called "Single Page Applications". The server delivers only a static single page which can be updated completely by the browser. Different states of the page are stored and can be accessed via a virtual address. Reloading this virtual address page restores the state and does not lead to a start or main page.

3 Backend

3.1 Motivation

According to moore's law [1], “the number of transistors on integrated circuits doubles approximately every two years”. In the past the law holds for the number of transistors of single CPU systems. At the same time the clock rate of these CPUs increased until it reached its limit. Nowadays the law is still valid, only for multi-core CPU designs. These systems contain more and more CPUs or / and CPUs with more and more cores. In addition, systems are often combined to clusters e.g. in a cloud computing environment. Thus, concurrent and distributed programming techniques gain in importance. Traditional techniques use multithreading programming in combination with shared, mutable states in order to develop parallel programs. Making a shared space thread save is usually a cumbersome and error prone process and requires a high level of programming skills. Further more, the programs often do not exhibit a high grade of parallelism because often threads have to wait until the predecessor thread has finished and the states can be unlocked. For developing distributed programs, further technologies like web services or various message bus systems have to be used what can lead to a more complicated system setup and to a higher effort of maintaining.

For all of the above reasons, actor oriented developing frameworks are gaining in popularity. Two examples are ActorFx [7] and the Akka toolkit [3]. Akka is build on top of the JVM and can be used as a Java library as well as with the Java derivative Scala [2], which actor extension it is. The latter has already reached a mature developing state and allows building concurrent, scalable, distributed systems. It support location transparency concerning the execution and instantiation of actors. Remote actors can used in the same way as local ones. Because of the asynchronous message passing concept, applications can easily scale up on multicore servers. We have chosen Scala and Akka for our approach.

3.2 Architecture

The backend architecture is depicted on the right side of figure 5. The backend is developed with the programming language Scala and its actor extension Akka[3]. Akka provides an alternative to classical thread concepts used in languages like Java. An actor is a certain kind of class equipped with send and receive functions. By avoiding shared states and the problem of defining thread save classes, actors make it easier to design parallel and even distributed applications. Every rectangle symbol on the backend side of figure 5 represents an actor. Firstly the main actors and their tasks are introduced and secondly an example for a request and its processing in the backend is given to explain how the actors complement each other. The different actors fulfill the following tasks:

1. The **FrontendInterfaceActor** is an HTTP server and provides the REST interface to the frontend. For every request send from the frontend to the backend, a certain sub-interface actor is instantiated and the request is forwarded to it. Every sub-interface actor performs a certain task. Since all actors are executed in parallel, the FrontendInterfaceActor itself is able to process a huge number of requests.
2. The **InstanceInterfaceActor** e.g. performs all requests regarding the management of a process instance.
3. The **SubjectProviderManagerActor** instantiates the SubjectProvider-Actors and manages the mapping between UserID and ActorReference of

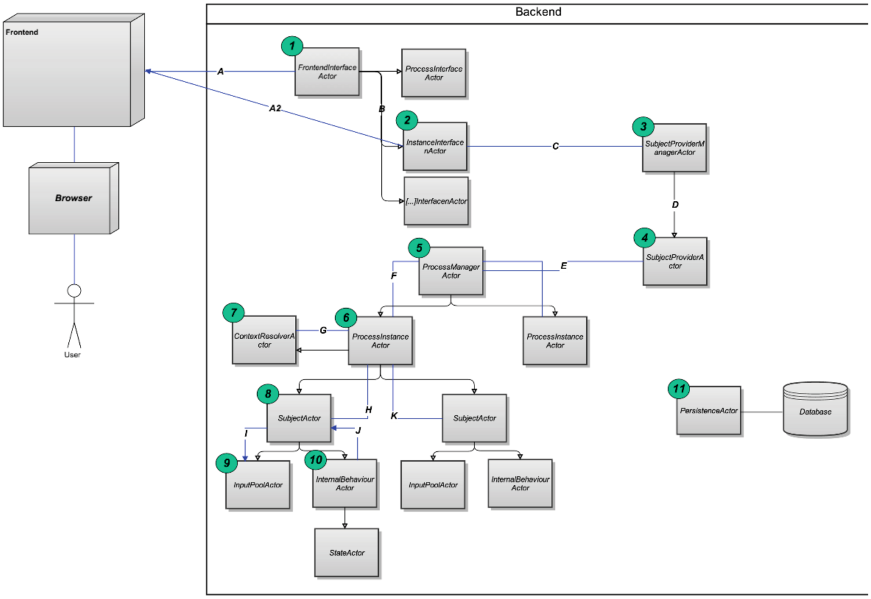


Fig. 5. Frontend and backend Architecture. The rectangles in the backend are the actors. Black arrows are used for instantiation, blow ones for an example message flow.

SubjectProviderActors. SubjectProvider is a synonym for agent. Since Akka has besides to the usual actors a certain kind of actor called agent, the term subjectprovider is used in the present paper. The UserID is the ID of the human agent who has triggered the request.

4. **SubjectProviderActor** is a proxy for the human agent in the frontend. It has stored the UserID and manages the interaction between the remaining part of the backend and the agent.
5. The **ProcessManagerActor** manages all processes and instantiates new process instances on demand. Additionally, it manages the mapping among SubjectProvider, Process Instance and Subject: Which SubjectProvider has an active subject in which subject instance ?
6. For every process instance a **ProcessInstanceActor** is instantiated. It instantiates the required SubjectActors and the message exchange among SubjectActors is passed through it. The mapping between Subject and UserID is requested from the ContextResolverActor.
7. The **ContextResolverActor** dissolves the mapping between the Subjects of the process model and the associated SubjectProvider. Example: SubjectProvider A is mapped to a Subject *Employee* of a process. When it send a message to the Subject *Manager*, the ContextResolverActor dissolves the associated SubjectProvider. If a dissolution is not possible, a list of all SubjectProvider's is given back as fallback solution.

8. The **SubjectActor** consists of two parts: The inputpool and an interpreter of internal subject behaviors. More details about the inputpool are given in section 3.3.
9. The **InternalBehaviourActor** performs the internal behavior of Subjects.

3.3 Inputpool

The inputpool is part of every subject, respectively every SubjectActor. It enables asynchronous communication between subjects. When a message is sent from one subject to another, it is stored in the inputpool until a receive action of the subject takes the message. Each SubjectActor has exactly one inputpool containing a set of first-in-first-out buffers (queues) for messages. There is one queue for each (subject name, message type) pair. The capacity of each queue is limited by a value given in the process model. If the limit of a queue capacity is reached, send actions from other SubjectActors are blocked until capacity becomes available.

3.4 Cross-Company Execution of Business Process Models

Akka supports the feature of location transparency. An external actor can be used like a local actor and the basic idea of our approach is to use external subjects like local subjects. The main problem is the matching of SubjectIDs and AgentIDs. The IDs are different on every machine.

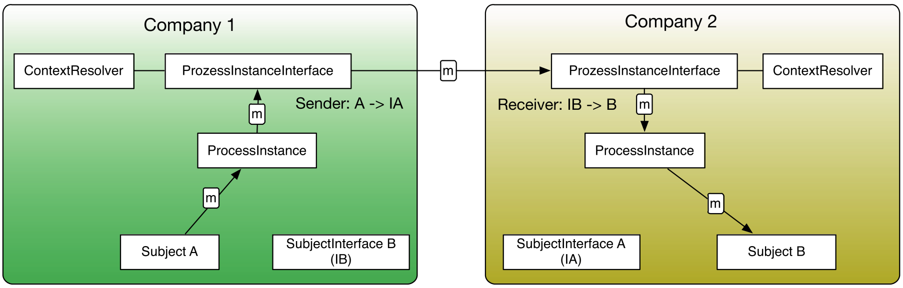


Fig. 6. Subject A of company 1 will send a message to subject B of company 2

In figure 6 an example for a messageflow between different companies is given. The Subject A of Company 1 will send a message m to the Subject B of Company 2. The message is sent from the SubjectActor of Subject A to the ProcessInstanceActor. This is the normal behavior since all message exchanges between subjects are done through this actor. The receiver in the process model is external and declared with the attribute *Interface Subject*. Therefor, the ProcessInstance-Actor forwards the message to the ProcessInstanceInterfaceActor which is asking the context resolver for a mapping. Afterwards, the ProcessInstanceInterfaceActor replaces the sender ID, which is stored in the header of the message, from A to IA and forwards it to the ProcessInstanceInterfaceActor of Company 2. This ProcessInstanceInterfaceActor of Company 2 uses its ContextResolverActor to replace the receiver ID from IB to B. The message is routed to the concerning ProcessInstance Actor of Company 2 which will forward it to the SubjectActor of Subject B.

3.5 Scalability

The same location transparency mechanisms could be used for distributed execution with the purpose of performance scaling. For a small number of users, an installation on one server node should be sufficient. A large number of users can be supported by outsourcing certain parts of the engine to other nodes. Good parts for outsourcing are the ProcessInstance Actors.

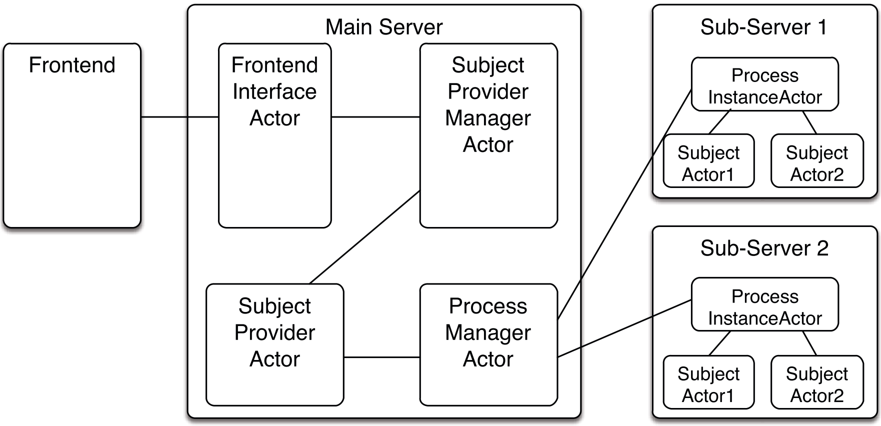


Fig. 7. Architecture for distributing the execution on several server nodes

An simplified architecture for distributing the ProcessInstanceActor's on several server nodes is given in figure 7. The ProcessManagerActor can instantiate ProcessInstanceActors on different sub-servers by using Akka's remote creation functionality. A simple solution would be a distribution by numbers. Every sub server gets the same number of instances. A more sophisticated solution can consider monitoring data to realize a load balancing distribution.

4 Summary and Future Work

This paper introduces a S-BPM managing suite with three important features. 1) The suite is a Software as a Service solution. 2) Cross company execution of business process models is supported. 3) The backend is based on an actor architecture and provides possibilities for performance scalability. The frontend user interface and components are explained and also the backend and its components.

The experiences made developing with Scala and Akka are very well. On the one hand, Scala's syntax is very different to Java's. On the other hand, the syntax is easy to learn and more intuitive. Scalas language concepts are well thought through which leads to less and better readable code in comparison to Java. Working with actors is straight forward. The messages can be easily defined and the message exchange is done by certain send and receive methods. The work with remote actors, i.e. actors which are instantiated and executed on external server, is uncomplicated. Nevertheless, developing complex distributed systems is still challenging. The more the system is distributed, the more actors have exchange messages among each other and the more complex the protocols have to be.

We plan to improve the creation of behavioral interfaces to support interfaces with 1 to n relations. Furthermore, we plan to support software agents in addition to the current supported human agents.

Acknowledgments. The work presented in this paper was performed in the context of the Software-Cluster project SINNODIUM (www.software-cluster.org). It was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. "01IC10S01" — 01—C10S05. The authors assume responsibility for the content.

References

1. Moore's law (2014), `http://en.wikipedia.org/wiki/Moore%27s_law` (last accessed on January 15, 2014)
2. Scala (2014), `http://www.scala-lang.org/` (last accessed on January 15, 2014)
3. Akka: A toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications on the JVM (2014), `http://akka.io/` (last accessed on January 15, 2014)
4. Borgert, S., Steinmetz, J., Mühlhäuser, M.: ePASS-IoS 1.1: Enabling Inter-enterprise Business Process Modeling by S-BPM and the Internet of Services Concept. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 190–211. Springer, Heidelberg (2011)
5. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management. Springer, Berlin (2012)
6. Meyer, N., Feiner, T., Radmayr, M., Blei, D., Fleischmann, A.: Dynamic Catenation and Execution of Cross Organisational Business Processes-The jCPEX! Approach. In: Fleischmann, A., Schmidt, W., Singer, R., Seese, D. (eds.) S-BPM ONE 2010. CCIS, vol. 138, pp. 84–105. Springer, Heidelberg (2011)
7. Microsoft Open Technologies: The Actor Framework for Windows Azure (2014), `http://actorfx.codeplex.com/` (last accessed on January 15, 2014)

When Processes Alienate Customers: Towards a Theory of Process Acceptance

Thomas Müllerleile and Volker Nissen

Wirtschaftsinformatik für Dienstleistungen,
Helmholtzplatz 3, 98684 Ilmenau, Germany
{thomas.muellerleile,volker.nissen}@tu-ilmenau.de
http://www.tu-ilmenau.de/wid

Abstract. Business processes and BPM can deliver great value to process clients. This is, however, only possible if the process is triggered in the first place. The option not to trigger a business process depends on the acceptance of the process by the involved subjects. In the present paper "Grounded Theory" is used to construct a theory of process acceptance from empirical data using qualitative content analysis. The analysis reveals that missing process acceptance can have a substantial economic impact. This indicates that business processes posses inherent social properties which should be measured and managed by process designers.

Keywords: Subjects, BPM, Interaction, Process, Acceptance, Failure.

1 Introduction

Business process management (BPM) has received much attention in the last 20 years by addressing important business challenges of the globalized postindustrial economy. BPM enables companies to strive towards effective and efficient processes. This endeavor should lead, among other things, to product and process innovation, cost reduction and higher customer satisfaction rates. These positive results, however, can only be achieved by triggering of the process by its users and through correct execution by the process operators. Subject oriented BPM, as an emerging area of BPM research, is highlighting the importance and behavior of these stakeholders in contrast to traditional BPM which focuses on functional process design.

Unfortunately, business scandals in various domains have shown that companies, and therefore the process operating subjects, sometimes do not execute their processes according to own standards or do not use existing processes at all. This failure in process execution can lead not only to suboptimal organizational business structures but also to life threatening disasters[1]. By circumvention of official channels, individuals within the company create shadow organizations.

[1] Bhopal or Chernobyl Disaster. Increased outbreak of MSRI in Hospitals due to lack of hand washing.

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 171–180, 2014.

Thus, unofficial processes and shadow IT systems emerge, which run alongside the official organization. The formation of these entities, which signal demand through their existence, can have different causes. Theses causes include that the established official processes are not well accepted and may even be unknown by their prospective users. Furthermore, users may perceive the value or the usefulness of existing processes, compared to their counterparts in unofficial channels, as low. As a result process execution is changed or not triggered at all. This behavior can lead to higher costs and could render governance, risk, and compliance efforts useless [4]. Therefore, in general, the value delivered by a business process depends on the social interaction patterns of its stake-holding subjects. Subject oriented business process management (S-BPM) focuses on subjects, which are responsible for any process variance and their collaboration via structured communication in business processes. The current paper investigates an airplane boarding process, in which subjects interact which each other and the flight operator in a process, with the goal, to find a suitable seat. To address the aforementioned issues, the present paper adopts an inductive methodology to construct a theory of process acceptance based on empirical company and customer experiences. The data set used draws on information provided by the various process subjects, their interaction patterns and goals. As a methodology "Grounded Theory" was selected.

2 Method

2.1 Grounded Theory

Qualitative methods based on "Grounded Theory" (GT) [1] enable researchers to inductively formulate new theories from available data. GT provides different comprehensive methods for data acquisition, data analysis, and construct conceptualization. In comparison to research with other methods, GT based research starts without formulating a hypothesis. GT also advocates to conduct a literature review upon formulating the theory, which results from the research process. GT generates substantive theories, which are closely coupled to an area of research. These "substantive theories" are used to formulate a more general theory at a later stage. The whole process has fluid characteristics, i.e., data acquisition and analysis interplay with theory building. Information from new sources is used to modify the theory being generated. The overall goal of GT is to formulate a theory based on empirical data. Several different data sources, e.g., case studies, events, raw data, are useful for GT based research. Contrary to quantitative empirical research, GT sets representative statistical sampling aside. In fact, GT tries to use data sources based on theoretic sampling, which means data is selected actively to show the occurrence or absence of the researched phenomena [2]. Three consecutive research steps are conducted with a given data set. First, concepts are created from analyzing the data. In a later step these concepts are used to formulate the theory. Second, categories are generated to aggregate the developed concepts at a higher level. Third, propositions from generalized relationships, regarding the concepts and categories, are derived. Therefore, GT and its associated methods strive to uncover the underlying behavior of the observed phenomena by formulating the aforementioned propositions.

In the context of epistemology, qualitative methods have drawn much critique regarding their rigorousness. GT, which has attracted much criticism itself [3], ameliorates these shortcomings by providing a comprehensive research framework. Qualitative methods are often used within a GT research project. This is, however, not mandatory as GT does not exclude any research method. Therefore a mixed methods approach is used in the present paper combining qualitative and quantitative analysis and modeling.

2.2 Research Design

In general, methods based on GT are used in the field of information systems, but they are uncommon in the area of BPM. The present paper tries to follow a GT approach for theory construction based on methodological fit [5] and to set out a future research agenda. This is done by applying a mixed methods approach to a publicly available data set which highlights a very peculiar social-economic phenomenon in the context of subject oriented BPM. In the present paper, GT is selected for three reasons. First, GT excels when no prior theory about the research object is available [12]. Second, the current research focuses on how individuals, influenced by business process design, interpret and construct reality [1]. Third, with the collected empirical data of an interesting phenomenon a proposition can be formulated. This results in a higher level of abstraction than the raw data itself.

Data Collection

The data set at hand acts as a first source of inquiry. As GT mandates, the information embedded in the data is conceptualized. This is accomplished by textual labeling and process modeling. Drawing on these entities, categories are formed to develop propositions. The data set, which can be considered suitable in the GT context, results from publicly available information about easyJet PLC, which styles itself as a low cost airline based in the UK, which offers mainly flights to European destinations. The phenomenon observed is a change in the customer seating policy, from open to preassigned seating, inside the airplane. As data sources for the content analysis, financial reports, corporate news, and a press interview with the CEO are used [9][10], as reported in table 1. Two processes are modeled, using the conveyed information in the data sets, to aid theory construction. One before the change in seating policy was implemented and one after. Many news outlets published the story online. Therefore, this study makes use of comments, which were added by users to the news article. These comments are used to gauge the overall reaction of prospective customers. From 201 comments which were posted online 147 could be technically retrieved. However, 23 were off-topic and therefore excluded from further analysis. Additionally, the best and worst rated 20 comments were retrieved and included into the data set.

Table 1. Types of data used

Type of raw data	No. of cases
News article (interview/video)	2
Company news	1
Company financial data	2
Comments	147
Top rated comments	20
Worst rated comments	20

Data Analysis
The data was analyzed within each type of raw data as well as across the data types to detect similarities and differences. In the initial coding phase, the data was annotated using open coding on a line-by-line level. Coding makes raw data accessible to further analysis. This is achieved by categorizing the data proposed by the data itself. Coding acts as a link between the collected data and theory construction [6]. About 100 codes were produced by the open coding process. A major concern is the short user comment length. As one comment is treated a single case, it is deemed not useful to employ a memo method, as suggested by Charmaz [6]. In the second stage of data analysis, axial codes were constructed. These axial codes describe the relationships of the open codes discovered in the initial coding phase. The identified codes are further analyzed in the theoretical coding phase. Goal of the theoretical coding is to select a code which conveys the key conceptual category. It should be noted that the concepts identified are not exhaustive. In the last phase, the theory construction stage, the identified codes are used to build a theory.

2.3 Content Description

Company View
Recently, easyJet PLC published financial reports, which are accompanied by an interview with the CEO, Mrs. McCall [8]. During the interview she stated:

> *"We've also reallocated seating, all the way across the airline, so you can now choose to pay a little bit extra to get an extra leg room seat or any seat you want or a window seat or an aisle seat or whatever. This has gone down very very well with our passengers. We done a lot of thing to the business passenger, flexi-fares are taking of. That means you can change your flight right up to the last minute, up to two hours before..."*

In a different interview [7] she added:

> *"There is no question there has been an increase in people who refused to even contemplate flying easyJet beforehand."*

In the financial statements easyJet reported an increase in revenues by almost 50%, which, according to Mrs. McCall, can be partly attributed to the change in seating policy. In fact easyJet could also increase the total number of passengers. Interestingly, the average age of all passengers increased from 38 to 42 years. According to easyJet, an older demographic is more likely to spend money on board, making this group very valuable. Also, easyJet reported that 25% of all passengers make use of this seat booking option.

Customer View

As online news articles allow readers to comment on the news, a short sentiment analysis of the available comments is conducted [7]. As seen in table 2, the majority of all comments are positive, considering the new seating policy. In the data source used, commentators could also rate the comments of the other users by marking them with green or red up and down arrows, which are also reported in table 2. These arrows can be interpreted as endorsement of the comment's embodied information. Overall, 650 negative and 2118 positive arrows were distributed by the users. Notably, positive comments about the new seating policy received 969 positive and 163 negative arrows, negative comments received 145 positive and 87 negative arrows. These ratings are useful for interpreting the general sentiment. However, users may have different reasons for rating a specific comment. Therefore, analysis is restrained to descriptive statistics.

Table 2. Ratings of User Comments

Type	Number	Positive rating	Negative rating
In favor of new seating policy	52	969	163
Not in favor of new seating policy	13	145	87
Unrelated	48	-	-

Model

Two process models may be deduced from the data sources. One model for the booking and boarding process before the changes took place and one after. As a modeling language the event-driven process chain (EPC) notation is selected. S-BPM is also evaluated but no adequate way of modeling interdependencies could be found. As subjects normally work in an interdependent setting, this aspect of subject orientation in modeling could be addressed in further research. Possibly, S-BPM could adopt suitable work-flow patterns or new elements to facilitate modeling of subject interdependence.

Figure 1 depicts the boarding process before the new seating policy, which is shown in figure 2, was implemented. Both models are deduced from the analyzed data sources. They implement the process as experienced by the passengers. It can be assumed, that both processes are not exhaustive and are likely to be more complex in the real world. It is obvious that the new process, to find the preassigned seat, implies less interaction with other passengers. The communication effort during the booking process is limited to desired seats, because normal seats are automatically preassigned. As a generalization, it can be deduced that

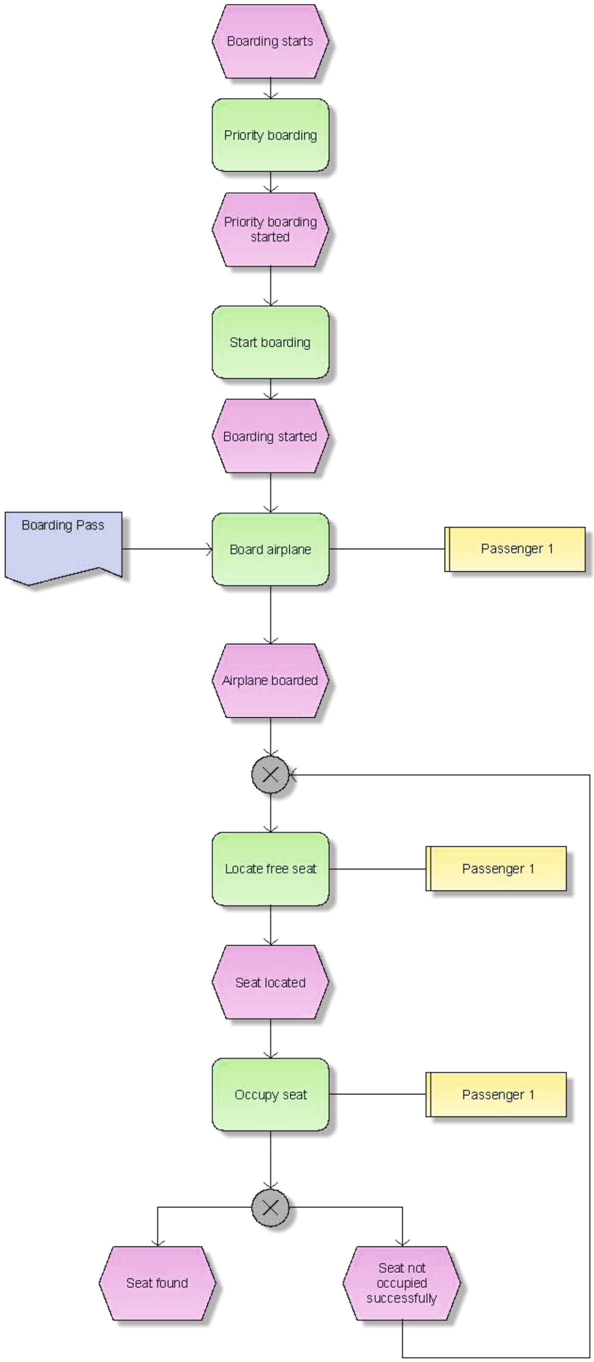


Fig. 1. Deduced boarding process before new seating policy was implemented

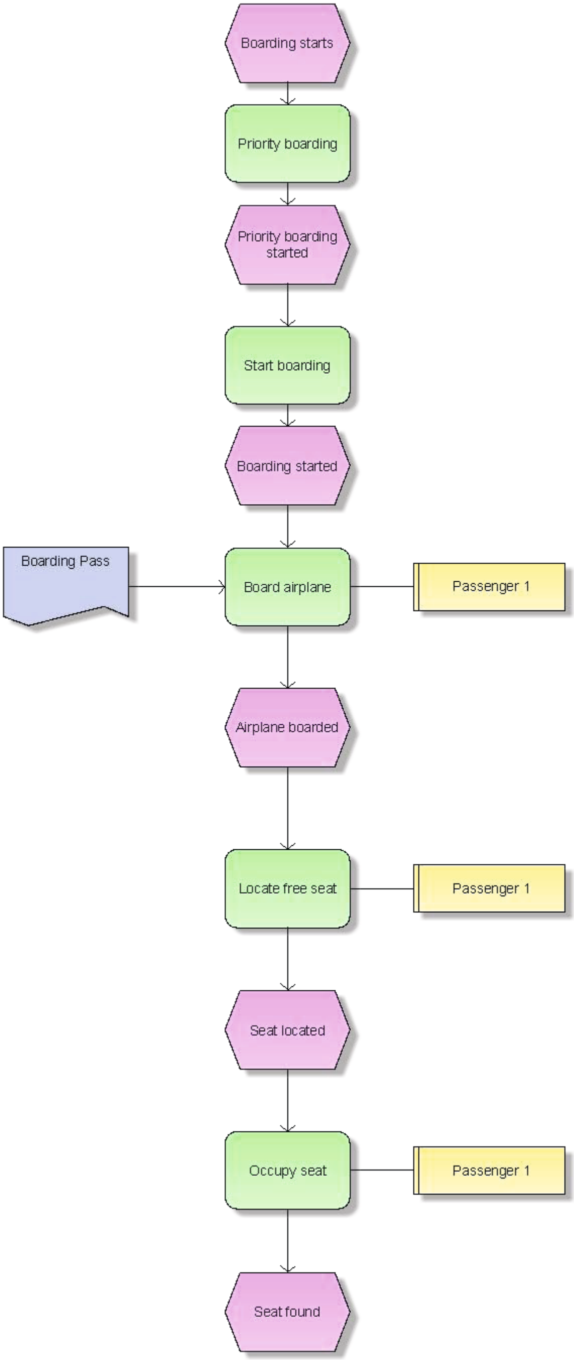


Fig. 2. Deduced boarding process after new seating policy (preassigned seats) was implemented

less interaction with subjects and resources are beneficial to the boarding experience. On the other hand, the old process required less seat management by easyJet. It is interesting to note that passengers attach more importance to the boarding process than the booking process.

3 Results

3.1 Company Perspective

The following concepts are found relevant in devising a new seating policy from a companies point of view.

Customers
easyJet customers include business and private travelers, which are subjected to customer retention efforts. These efforts focus on choice and affordable prices. Also, easyJet suggests that the implementation of a new seating policy is a reaction to passenger demand. As company research revealed, the new seating policy is very well accepted among its clients.

Corporate Strategy
As a public listed airline, easyJet is committed to increase profitability. This can be achieved through high customer retention and customer satisfaction. To this end, punctuality is considered as a key performance indicator. As a no-frills airline, easyJet is very cost sensitive. Therefore, the implementation of a new seating policy is not allowed to increase passenger fares.

Conditions for Adopting a New Seating Policy
As contemplated by management, the old seating policy alienated customers. Therefore, seating was identified as a problem. After receiving positive reaction from passengers, which took part in a test trial, the new seating policy was implemented for all flights. These findings are consistent with experiences reported in the posted user comments. After implementing the new policy, company objectives, e.g., punctuality and stable passenger fares, are still achieved.

3.2 Customer Reaction

As a reaction to the new seating policy the following concepts are found to be relevant from a passenger perspective.

Conditions before New Seating Policy Was Implemented
Clients were upset with the old seating policy. Many users describe that they actively contacted management to raise awareness of this issue. The old seating policy was connected with feelings of stress and disrespect for the customer. Especially, the high level of unpleasant interaction with other passengers to secure appropriate seating, e.g., for a family or a group, are regarded as a major drawback for flying with easyJet.

Attitude towards New Policy and Management
In general, the new seating policy is very well received. It is regarded as an improvement. Many passengers expressed that, with this new policy, they would fly with easyJet again or give the company a second chance. They stipulate that the competition would implement similar policy changes. Users expressed disregard for the management because the change is viewed as obvious and already requested by customers.

3.3 A Grounded Theory of Process Acceptance

This paper describes how a perceived small change in the design of a process can have an enormous impact on customer satisfaction, which results in higher reported profits. The change in seating policy had major implications for the internal IT systems. It is important to note that, before the new seating policy was implemented, the boarding process was perceived as stressful. As passengers did not accept the boarding process, they executed their option not to book a flight with this airline. In general, they would have the option not to fly at all or choose another airline with a less stressful boarding process. It can be concluded that business processes possess inherent properties. These properties, which result directly from process design, are attributed by subjects exposed to the process. The most important trait is process acceptance. If process acceptance is missing, clients are unlikely to trigger the process, e.g., by booking a flight, at all. In general, this missing process acceptance will result in lower demand and therefore lower profits.

4 Discussion and Conclusion

The goal of this paper is to contribute to the existing literature about subject oriented BPM by highlighting an interesting phenomenon, and to provide empirical insights of which factors are critical to process acceptance. These contributions are twofold. First, and most notably, the data analysis yielded a grounded theory of process acceptance by linking a company decision to client reactions. It is evident that subject-orientation in the design of business processes can help improve process acceptance. In the analyzed data an increase in acceptance of the boarding process results in higher profits for the company. Therefore, it can be concluded, that process acceptance should be measured and managedIt can be further stipulated, that one variable which influences process acceptance is user interaction. Future research should concentrate on identifying other influential variables and their impact on process acceptance. These findings could be integrated into a general process acceptance model. Second, shortcomings in the existing modeling languages, regarding the modeling of subject interdependent decisions, are identified. As subjects are working in a complex interdependent environment, future research should also focus on new modeling methods that map interdependent user interactions in business processes [11]. Especially S-BPM as a modeling language could benefit from the possibility to model interdependent subject interactions. However, some limitations are worth noting. In this study, only publicly available qualitative data sources regarding one process of one company are analyzed. Therefore, in future research the presented content analysis could be augmented with other quantitative and qualitative research methods. Especially interviews with process managers may be used to elicit information about which elements may influence process acceptance.

References

1. Glaser, B.G., Straus, A.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, Chicago (1967)
2. Corbin, J.M., Strauss, A.L.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, 3rd edn. Sage, Thousands Oaks (2008)
3. Thomas, G., James, D.: Reinventing Grounded Theory: Some Questions about Theory, Ground and Discovery. British Educational Research Journal 32(6), 767–795 (2006)
4. Behrens, S.: Shadow Systems: The Good, the Bad and the Ugly. Communications of the ACM 52(2), 124–129 (2009)
5. Edmondson, A.C., McManus, S.E.: Methodological Fit in Management Field Research. Academy of Management Review 32, 1155–1179 (2007)
6. Charmaz, K.: Constructing Grounded Theory: A Practical Guide through Qualitative Analysis. Sage, London (2006)
7. Campbell, P.: EasyJet Lures Older Fliers by Ending the Scramble for Seats: Profits Surge by 50% After Change in Policy. Mail Online (2013), http://www.dailymail.co.uk/news/article-2510268/easyJet-lures-older-fliers-ending-scramble-seats-Profits-surge-50-change-policy.html
8. BBC News: EasyJet Sees Profits Boost Through Seat Booking Option. BBC News (2013), http://www.bbc.co.uk/news/business-24997505
9. Moore, P.: EasyJet to Bring in Allocated Seating on All Flights. easyJet PLC (2012), http://corporate.easyJet.com/media/latest-news/news-year-2012/05-09-2012-en.aspx
10. EasyJet plc: Annual Report. easyJet (2013), http://2012annualreport.easyJet.com/downloads/PDFs/Full_Annual_Report_2012.pdf
11. Giachetti, R.E.: Understanding Interdependence in Enterprise Systems: A Model and Measurement Formalism. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 261–272. Springer, Heidelberg (2006)
12. Martin, P.Y., Turner, B.A.: Grounded theory and organizational research. Journal of Applied Behavioral Science 22, 141–157 (1986)

Author Index