5th SIKS/BENAIS Conference on

Enterprise Information Systems 2010

Eindhoven, The Netherlands, November 2010
Proceedings

Boudewijn F. van Dongen
Hajo A. Reijers (Eds.)

Volume Editors:

Boudewijn F. van Dongen
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: b.f.v.dongen@tue.nl

Hajo A. Reijers
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: h.a.reijers@tue.nl

The papers and extended abstracts in this book comprise the proceedings of the 5th Conference on Enterprise Information Systems (EIS 2010), held on November 16th 2010 at the Eindhoven University of Technology (TU/e).

EIS 2010 is organized by SIKS (School for Information and Knowledge Systems) in cooperation with LOIS (strategic initiative for Logistics, Operations and Information Systems), BENAIS (Benelux Chapter of the Association for Information Systems) and NIRICT (Netherlands Institute for Research on ICT).

The conference offers a unique opportunity for research groups from both the Computer Science-side and the Management-side to report on research, meet and interact. We also welcome practitioners with an interest in research and innovation, as well as doctoral students in the early stages of their careers.

These proceedings also appear online at http://ceur-ws.org/

SIKS	BENAIS	NIRICT	TU/e
http://www.cs.uu.nl/siks/	http://www.benais.nl/	http://nirict.3tu.nl/	http://www.tue.nl/lois/

Proceedings of the

5th SIKS/BENAIS Conference on

Enterprise Information Systems 2010

Eindhoven, The Netherlands,

16th November 2010

Boudewijn F. van Dongen
Hajo A. Reijers (Eds.)

Preface

In follow-up to the four previous conferences, which were held in Utrecht, Groningen, Tilburg and Ravenstein, the fifth edition of the SIKS/BENAIS Conference on Enterprise Information Systems (EIS 2010) is held in Eindhoven this year. The purpose of the conference series is to bring together Dutch and Belgian researchers interested in the advances in and the business applications of information systems. Against that backdrop, we are happy to have received contributions from almost all of the Dutch and Belgian research groups active in this field.

Overall, the program consists of 16 presentations of which six concern totally new work. Each of these was duly reviewed by at least three members of the program committee. The remaining 10 presentations relate to work already being published in a high-quality outlet and considered highly attractive to bring under the attention of the Dutch/Belgian EIS community.

On top of these presentations, we are very happy with the incorporation of two keynote presentations in the program. The opening keynote is to be given by Theodoor van Donge, CTO at Cordys and responsible for R&D. The closing keynote will be provided by prof. John Krogstie from the Norwegian University of Science and Technology.

At this occasion, we wish to express first and foremost our gratitude to the members of the EIS community who have fulfilled roles in the program committee for this conference. Their valuable feedback has helped the presenters to further improve their work. We also wish to thank SIKS, BENAIS, NIRICT, LOIS, and all involved local staff at Eindhoven University of Technology for their support in organizing this event.

It is our hope that the conference will stimulate discussions in our community, foster existing collaborations and lead to new ones. But most important of all, we hope that you will enjoy the conference day.

November 2010

Boudewijn van Dongen
Hajo Reijers

Conference Organization

Organization Chair:

Wil van der Aalst

PC Co-chairs:

Boudewijn van Dongen

Hajo Reijers

Publicity Chair:

Jan Martijn van der Werf

Demo Chair:

Pieter van Gorp

Program committee:

Marco Aiello (RUG)
Hans Akkermans (VU)
Mehmet Akşit (UT)
Farhad Arbab (CWI)
Egon Berghout(RUG)
Patrick van Bommel (RUN)
Harry Bouwman (TUD)
Sjaak Brinkkemper (UU)
Bert de Brock (RUG)
Walter Daelemans (UA)
Virginia Dignum (UU)
Remco Dijkman (TUE)
Pascal van Eck (UT)
Rik Eshuis (TUE)
George Fletcher (TUE)
Jaap Gordijn (VU)
Luuk Groenewegen (LEI)
Remko Helms (UU)
Willem-Jan van den Heuvel (UvT)
Jos van Hillegersberg (UT)
Stijn Hoppenbrouwers (RUN)
Geert-Jan Houben (TUD)
Patricia Lago (VU)
Aldo de Moor (CommunitySense)
Michaël Petit (UN)
Rinus Plasmeijer (RUN)
Stephan Poelmans (HUB)
Geert Poels (UGent)
Erik Proper (RUN)
Piet Ribbers (UvT)
Marten van Sinderen (UT)
Monique Snoeck (KUL)
Yao-Hua Tan (TUD)
Jan Vanthienen (KUL)
Roel Wieringa (UT)
Hans Weigand (UvT)
Niek Wijngaards (Thales)

Program

	Filmzaal	Zaal L4
09:30	**Reception with coffee & tea**	
09:50	**Opening**	
10:00	**Keynote: Theodoor van Donge** *Our Journey from Packaged Applications towards SOA, BPM2.0 and Composite Applications Deployed in Cloud and OnPremise*	
11:00	**Break**	
11:20	**Maturity Models**	**Requirements Analysis**
	Amy Van Looy, Manu De Backer and Geert Poels Which Maturity Is Being Measured? A Classification of Business Process Maturity Models	*Tom Huysegoms, Monique Snoeck and Guido Dedene* Building a Requirements Engineering Methodology for Software Product Lines
	Marlies Van Steenbergen, Rik Bos, Sjaak Brink-kemper, Inge van de Weerd and Willem Bekkers The Design of Focus Area Maturity Models	*Marco Aiello, Pavel Bulanov and Heerko Groefsema* A Survey of Variability Management Requirements
	Marijn G.A. Plomp and Ronald S. Batenburg Measuring Chain Digitisation Maturity: An Assessment of Dutch Retail branches	*Bert de Brock, Tommi Tervonen, Hans Hillege and Pieter de Graeff* Current Status and Future Perspectives of Drug Information Systems
12:20	**Lunch break**	
13:50	**Collaborative Modeling**	**Architecture, Ontology & Model Engineering**
	Sietse Overbeek, Virginia Dignum and Yao-Hua Tan Value-Sensitive Design for Cross-Enterprise Regulation	*Ralph Foorthuis, Marlies van Steenbergen, Nino Mushkudiani, Wiel Bruls, Sjaak Brinkkemper and Rik Bos* On Enterprise Architecture Conformance and Benefits
	Agnes Nakakawa, Patrick van Bommel and Erik Proper Challenges of Involving Stakeholders When Creating Enterprise Architecture	*Philip Huysmans, Kris Ven and Jan Verelst* Designing for Innovation: Using Enterprise Ontology Theory to Improve Business-IT Alignment
	Denis Ssebuggwawo, Stijn Hoppenbrouwers and Erik Proper Collaborative Modeling: Towards a Meta-model for Analysis and Evaluation	*Björn Kijl and Durk Boersma* Developing a Business Model Engineering & Experimentation Tool - The Quest for Scalable 'Lollapalooza Confluence Patterns'
14:50	**Break**	
15:10	**Supply Chains**	**Process Model Transformations**
	Romana Aziz, Jos van Hillegersberg and Kuldeep Kumar Inter Organizational Relationships Performance in Third Party Logistics: Conceptual Framework and Case Study	*Pieter Hens, Monique Snoeck, Manu De Backer and Geert Poels* Transforming Standard Process Models to Decentralized Autonomous Entities
	Ricardo Seguel, Rik Eshuis and Paul Grefen Architecture Support for Flexible Chain Management	*Willem De Roover and Jan Vanthienen* Unified Patterns to Transform Business Rules into an Event Coordination Mechanism
15:50	**Keynote: John Krogstie** *How Can Enterprise Information Systems Utilize the Future Internet?*	
16:50	**Closure and drinks**	

Table of Contents

Session 3: Collaborative Modelling

Session 4: Architecture, Ontology & Model Engineering

Session 5: Supply Chains

Session 6: Process Model Transformations

Papers of Interest

Keynote: Theodoor van Donge

"Our journey from Packaged Applications towards SOA, BPM2.0 and Composite Applications deployed in Cloud and OnPremise"

BPM plays an important role in the area of Business Applications. Previously, we were used to a three-layer model for Business Applications. However, we have grown to a five-layer model, consisting of Databases, the Application server and the User interface, together with the new layers: Integration and BPM.

In his keynote, Theodoor van Donge presents the experiences with this new model he had with Cordys. Among other subjects, he will present the deployment models such as OnPremise and Cloud, but also different types of workflows such as BPMN and Case.

About the author:

Theodoor van Donge has over 25 years of IT innovation and leadership in the software sector. Theodoor played a key role in developing the pioneering ERP technology that saw The Baan Company achieve global recognition and become a Fortune 500 company in the 1990s. As the key architect behind the Cordys platform, Theodoor takes responsibility for all software development undertaken at the company.

With Cordys, Theodoor had a unique opportunity to develop an enterprise IT solution within an entirely green field environment. From 2001, Theodoor led the team that built Business Operations Platform, now widely recognized as the purest composite BPMS available on the market, and one that proves it is possible to support business-centric process design using SOA.

Theodoor's fundamental approach to Cordys Business Operations Platforms' underlying architecture is the direct result of his cumulative learnings throughout his career, from his early involvement with the The Baan Company, to experience gained through Jan Baan's funding of Top Tier and WebEx Communications, as well as his commitment to always using the very latest Internet related technologies.

Keynote: John Krogstie

“How can Enterprise Information Systems utilize the Future Internet?”

The "Internet of Things" (IoT) has come to describe a new paradigm that enables the Internet to reach out into the real world of physical objects. Technologies like RFID, short-range wireless communication, real-time localization and sensor networks are becoming increasingly common and turning IoT into reality. It is expected to grow rapidly into a huge new market domain that may lead to disruptive changes in areas such as logistics, energy management and healthcare. Mobile and collaborative applications and services utilizing information processing and process support enabled by sensor data from a vast numbers of connected and cheap devices will change many markets when being made more easily available. New event-driven architectures (EDA) providing varied information to support collaborative decision-making enable more decisions to be made closer to the problem owner. The expected impacts of the combination of IoT and EDA on business and society are formidable. It also opens the possibility to take into account additional input from users to ensure shorter turnaround from ideas to new, personalized information systems support.

Future Enterprise Information Systems will need to take this situation into account, addressing both technological and conceptual challenges. This talk will focus on the latter, discussing in particular the potential role of model-based techniques and how to assess and improve the quality of models and modeling approaches in this setting.

About the author:

John Krogstie holds a PhD (1995) and a MSc (1991) in information systems from the Norwegian University of Science and Technology (NTNU), where he is currently a full professor in information systems. He is also the Vice Dean of the faculty, responsible for the thematic area ICT at NTNU coordinating multidisciplinary research involving ICT at the university. John Krogstie is the Norwegian representative for IFIP TC8 and chair of IFIP WG 8.1 on information system design and evaluations. His research interest are information systems modeling, quality of models and modeling languages, eGovernment and mobile information systems. He has published around 175 refereed papers in journals, books and archival proceedings since 1991.

Session 1:

Maturity Models

Which Maturity Is Being Measured? A Classification of Business Process Maturity Models

Amy Van Looy[1,2], Manu De Backer[1,2,3,4], Geert Poels[2]

[1] University College Ghent, Department of Management & Informatics, Voskenslaan 270, B-9000 Ghent, Belgium
{amy.vanlooy, manu.debacker}@hogent.be
[2] Ghent University, Department of Management Information Science & Operations Management, Tweekerkenstraat 2, B-9000 Ghent, Belgium
{Amy.VanLooy, Manu.DeBacker, Geert.Poels}@UGent.be
[3] University of Antwerp, Department of Management Information Systems, Prinsstraat 13, B-2000 Antwerp, Belgium
{Manu.DeBacker}@ua.ac.be
[4] K.U.Leuven, Department of Management Informatics, Naamsestraat 69, B-3000 Leuven, Belgium
{Manu.DeBacker}@econ.kuleuven.be

Abstract. Today's organizations face the challenge to excel due to demanding customers. Hence, they are relying on their business processes to outperform competitors. Maturity models have been proposed to gradually assess and improve business processes. However, the proliferation of business process maturity models has complicated the practitioner's choice. This article clarifies the foundation of business process maturity and presents a classification of maturity models. First, a literature study was conducted, based on the concepts of business process (BP), business process management (BPM), and business process orientation (BPO), to identify the different capabilities to be addressed by a business process maturity model: (1) modeling, (2) deployment, (3) optimization, (4) management, (5) culture, and (6) structure. Afterwards, these capabilities were used to compare and classify 61 business process maturity models. The main result is that we found six different types of maturity being measured by the currently proposed maturity models.

Keywords: business process maturity, business process management, business process orientation

1 Introduction

As the growing globalized market is characterized by demanding customers, organizations are striving to excel in order to gain competitive advantage or to outperform competitors in their societal obligations. Hence, organizations are increasingly focusing on their business processes [1]. Business process management is expected to contribute to both process excellence and business excellence by assuring a uniform way of working and by continuously looking for optimizations [2].

Nonetheless, the journey towards process excellence is challenging. As a result, various authors have proposed step by step road maps with best practices, from which organizations gradually benefit [3,4,5,6]. These road maps are called business process maturity models (BPMMs). They are evolutionary models for measuring (AS-IS) and improving (TO-BE) maturity, or '*the extent to which an organization consistently implements processes within a defined scope that contributes to the achievement of its business goals*' [7, p.2]. Maturity aims at systematically increasing the capabilities of a business process and the organization to deliver higher performance over time [6,8].

Given the importance of mature business processes, a proliferation of maturity models was realized during the recent decades [9]. It started with frameworks to deal with the software crisis during the 1970s-1980s, and which have been adapted to all types of business processes afterwards. At present, maturity models for specific business processes are integrated into single models [7,10,11], and new models have been designed for generic business processes [12]. Consequently, this proliferation of BPMMs prompts us to evaluate their content. For this purpose, the present study aims at providing a foundation for business process maturity, grounded in the business process literature, instead of rebuilding on existing BPMMs. We theoretically explore the capabilities to be addressed by a generic BPMM in the first research question:

(1) which capabilities, i.e., theoretical model components, must be assessed and improved to increase the maturity of a business process?

However, we do not assume that every BPMM actually has a model component for each capability found by the previous question. This leads us to the second research question:

(2) can the BPMMs be classified by the capabilities they actually address? If so, are there different types of maturity being measured?

Both research questions contribute to the BPMM literature, without presenting a new model. They clarify the BPMM fundamentals and a classification to support practitioners while choosing a model that best fits the organizational needs.

The subsequent section deals with the methodology. Next, the research results are presented (section 3) and discussed (section 4). Afterwards, section 5 explains the plans for future work. The last section concludes by summarizing the BPMM components and the resulting BPMM classification with possible maturity types.

2 Methodology

The research approach was twofold: (1) a literature study to identify the capabilities to be addressed, and (2) a comparative study to classify the existing BPMMs.

2.1 Identification of Theoretical BPMM Components: Literature Study

A BPMM assesses and improves a business process throughout its lifecycle by focusing on the necessary capabilities to perform. Hence, the model components of a BPMM must affect business process performance. In order to identify the theoretical model components, we relied on the extensive literature concerning business processes, which findings have been repeatedly corroborated by evidence.

It resulted in three comprehensive concepts, which are closely linked to the traditional business process lifecycles [13]: (1) business process (BP), (2) business process management (BPM), and (3) business process orientation (BPO). Their respective definitions clarified the differences between the concepts and indicated the theoretical BPMM components, i.e., the capabilities to be addressed. These components are also supported by theories on critical success factors for BP, e.g. [14].

2.2 BPMM Classification: Comparative Study

The theoretical components, previously found, were validated by collecting existing BPMMs. After mapping their content to the components, a classification was derived to determine the type of maturity being measured per model.

The research scope was set to generic business processes. It excludes BPMMs addressing specific process types, such as in the initial software engineering maturity models. However, models that integrate various specific BPMMs were withheld to represent those specific topics. Also supply chain maturity models were selected to study cross-organizational value chains.

Data was collected during the second quarter of 2010. First, we searched for articles in academic databases and search engines on the Internet by using the combined keywords of '*process*' and '*maturity*'. Secondly, we traced the references in the identified articles to get access to other relevant sources.

We acknowledge some restrictions regarding the accessibility of articles (in Ghent University engines), the language (English, Dutch, French or German), and the keywords. Notwithstanding these limitations, the technique turned out to be fruitful in terms of the number of maturity models identified.

3 Results

The research results are discussed by following the same structure as the methodology section. Each subsection deals with a distinct research question.

3.1 Identification of Theoretical BPMM Components

Most definitions of BP refer to a transformation taking place, also illustrated as a value chain. They frequently mention: (1) predictable and definable inputs, (2) a linear, logical sequence or flow, (3) a set of definable and interrelated activities, (4) predictable and desired outputs, (5) horizontal or cross-departmental, (6) performed by resources, (7) repeatable, and (8) adding value for customers [15,16]. For instance, Harrington's definition sounds: *'a process is a series of interconnected activities that takes input, adds value to it, and produces output. It's how organizations work their day-to-day routines. Your organization's processes define how it operates'* [1, p.xxii]. This transformational view originates from manufacturing, and is less clear in service delivery. Hence, other definitions exist which rather emphasize a coordination of activities, instead of value-adding transformations, e.g. in [17]. Despite these different emphases, all BP definitions focus on *business process modeling* and *deployment*. As a result, both aspects will be used as theoretical model components for BPMMs.

Secondly, BPM involves continuously managing and improving business processes, guided by process owners. Depending on their background, authors underline more the IT benefits [18], or the management aspects [19]. Gillot [17], Gulledge Jr. and Sommer [20] summarize four BPM components: (1) *modeling*, (2) *deployment*, with automation where possible, (3) *optimization*, or improving business processes based on real metrics to evaluate business process performance, and (4) the *management* of business processes, each with a process owner and a cross-departmental process team. Similarly to BP, these four components are selected as theoretical BPMM components. The difference with BP, is that BPM also addresses managerial aspects and optimization efforts with regard to one or more business processes.

Some authors go beyond these four BPM components by also referring to organization management, in particular by adopting a horizontal structure and a process-oriented culture with rewards linked to the performance of business processes instead of departments [21]. Even though the distinction between BPM and BPO is not always explicitly made, e.g. in [6], it allows us to separately examine the different nuances. It results in a funnel structure of BP, BPM and BPO, as shown in Figure 1.

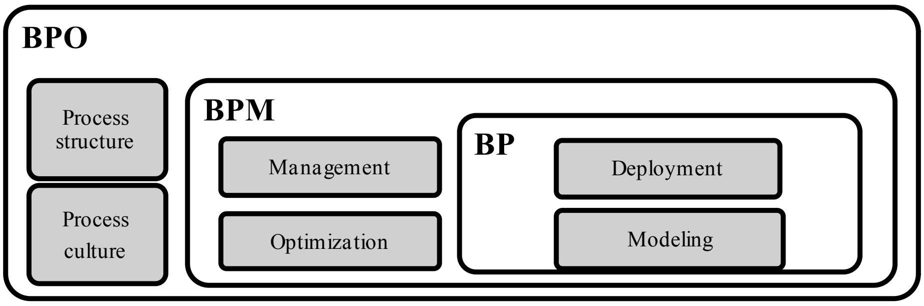


Fig. 1. The funnel structure of components in business process maturity models.

The six theoretical components specify whether BPMMs deal with BP, BPM or BPO.

3.2 BPMM Classification

61 BPMMs have been collected regarding business processes and supply chains:

(1) 37 business process models, of which:
- 13 academic [1,8,10,21,22,23,24,25,26,27,28,29,30];
- 24 non-academic [2,7,11,12,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50];

(2) 24 supply chain models, of which:
- 9 academic [51,52,53,54,55,56,57,58,59];
- 15 non-academic [60,61,62,63,64,65,66,67,68,69,70,71,72,73,74].

We have investigated their content in detail, including a mapping to the theoretical BPMM components. The result is a BPMM classification, as shown in Table 1.

Table 1. A classification of business process maturity models.

BPM for one BP (N=3)	BPM for more BPs (N=11)	BPM for all BPs (N=4)
• BP academic: [1,24] • BP non-academic: [40]	• BP academic: [22] • BP non-academic: [7] • SC academic: [55,58] • SC non-academic: [60,61,67,68,70,71, 72]	• BP academic: [21,27,29] • BP non-academic: [31]
BPO for one BP (N=3)	**BPO for more BPs (N=20)**	**BPO for all BPs (N=22)**
• BP academic: [8] • BP non-academic: [36,47]	• BP academic: [10,25] • BP non-academic: [11,12,38] • SC academic: [51,52,53,54,56,57, 59] • SC non-academic: [62,63,64,65,66,69, 73,74]	• BP academic: [8,23,26,28,30] • BP non-academic: [2,32,33,34,35,36,37, 39,41,42,43,44,45, 46,48,49,50]

In theory, all BP components are contained in BPM, and all BPM components in BPO. However, in practice, the lower components are not always present. BPMMs are classified as BPO if they address "process structure" or "process culture", and as BPM if they involve "management" or "optimization" without BPO components.

First, it turned out that no model merely addresses the BP components of "modeling" and "deployment". Instead, if present, they are supplemented by at least one BPM component. Secondly, the models strongly vary on the kind and number of business processes taken into account. As a result, a refinement in the classification was made to distinguish three BPMM foci: (1) a focus on one BP, (2) a focus on more than one, but not necessarily all BPs, and (3) a focus on all BPs in the involved organization(s) or supply chain (see Table 1). The result is a BPMM classification with six different types of maturity. It should be noted that some BPMMs offer multiple maturity types of which a practitioner can choose according to the organizational needs, for instance limited to a single BP or comprising all BPs [8,36].

4 Discussion

Six findings are drawn from the literature study and the comparative study. The first three concern the theoretical BPMM components (first research question), whereas the last three deal with the BPMM classification (second research question).

(1) *Component validation.* The six theoretical BPMM components, derived from the business process literature on BP, BPM and BPO, have been empirically validated

by comparing existing BPMMs. All actual model components were successfully mapped to a theoretical equivalent, without detecting new components.

(2) *Component coverage*. Most BPMMs do not cover all theoretical components, but three to five of them. All models address both “optimization” and “management”, except for four models, with [24,72] ignoring “management” and [37,51] underestimating “optimization”. The “structure” component is often neglected.

(3) *IT-enabled components*. Although IT is not a prerequisite, the majority prescribe IT to enable the three lowest components: “modeling” < “optimization” < “deployment”. The degree varies from general IT, such as mentionning hard- and software, to specific IT, e.g. EDI, ERP, SOA, SaaS, BPMS, and specific vendor tools.

(4) *No BP maturity type*. The collected BPMMs demonstrate that merely improving “modeling” and “deployment” are insufficient to achieve higher maturity regarding generic business processes, and that “optimization” and “management” are paramount. For instance, not all business processes need to be fully modeled in advance, e.g. semi-structured process flows in service delivery. Nonetheless, such a BPMM may theoretically exist, but restricted to specific business processes, e.g. by focussing on the workflows of manufacturing processes.

(5) *BPM and BPO maturity types*. The majority of collected BPMMs measure BPO maturity, mainly because of process-oriented values, e.g. a client focus, innovation, empowerment or trust, and the rewards to ensure their realization. Although an organization-wide perspective fosters higher maturity, it is not included in all models. Organizations can limit maturity to BPM by assigning a process owner to manage and statistically track a business process, possibly restricted to a department. Nonetheless, they won’t gain all benefits if the process owner has no cross-departmental authority nor if collaborating departments distrust each other.

(6) *Number of BPs*. BPMMs can be used to cope with one, more or all business processes. However, the models for a single business process are less numerous. More often, they are used in a single business domain with multiple business (sub)processes, such as software engineering or the supply chain. For instance, the latter has business processes for buying, producing, selling and planning products and services. This finding is conform to the idea of a large cross-departmental or cross-organizational business process, or horizontal value chain, with subprocesses in each department. Also frequent are BPMMs involving all business processes, which rather take a management perspective instead of focusing on particular business processes.

5 Future Work

All BPMMs will be further compared with regard to other elements in the assessment (AS-IS) and improvement (TO-BE) method, such as the lifecycle levels and the road map. Case studies will be conducted for the most comprehensive models. Above all, we will explore additional theories on the critical success factors for BP to obtain an operationalization of each component. Afterwards, we will be able to evaluate whether a new model design is appropriate for cross-organizational processes, and what the IT impact may be per component. Interestingly, different tracks may be identified depending on the organization size, type (products or services) and sector.

6 Conclusion

A business process maturity model (BPMM) addresses the capabilities of a business process and the entire organization, expressed as overall maturity, to deliver higher performance over time. These capabilities are represented by the BPMM components, which are systematically assessed and improved. The present study has elaborated on the theoretical model components to specify what is being measured by a BPMM. It has compared 61 BPMMs on six theoretical components, found in the business process literature. The components are linked to the traditional lifecycle of a business process, supplemented by organizational aspects: (1) modeling, (2) deployment, (3) optimization, (4) management, (5) culture, and (6) structure. In pairs, they form a funnel structure, starting from a business process (BP), which is a subset of business process management (BPM), and which is part of business process orientation (BPO).

However, in practice, BPMMs do not necessarily address all theoretical BPMM components. Above all, given the proliferation of BPMMs, practitioners may experience difficulties in choosing a model that best fits the organizational needs. In order to facilitate this choice, we present a BPMM classification based on two decisions: (1) which BPMM components are important for the organization (does a business process management perspective suffice or is an organizational perspective required?), and (2) which business processes to assess and improve (is there a focus on one, more or all BPs?). It results in six possible types of maturity: BPM maturity for one, more or all business processes, and BPO maturity for one, more or all business processes in the involved organization(s). Evidence has shown that a BP maturity type, centered around modeling and deployment, does not exist for generic business processes, as management and optimization are paramount.

In summary, the present study has reached its aim of providing a BPMM foundation in the BP literature. The six capabilities to be addressed in a generic BPMM have been identified and validated, as queried by the first research question. Regarding the second research question, the concept of maturity has been refined by specifying different maturity types. The resulting BPMM classification is relevant for both practitioners and academics, and contributes to the rather scarce BPMM literature. It allows clear communication, with scholars being able to clarify which dimension of maturity they investigate. New BPMMs may be designed based on the six theoretical BPMM components. Furthermore, the study challenges the maturity of maturity models by highlighting different designs, e.g. are BPO models for all BPs more complete and thus necessarily better than BPM models for one BP? Future research will focus on the operationalization by organization size, type and sector.

References

1. Harrington, H.J.: Process Management Excellence. The Art of Excelling in Process Management. Paton Press, California (2006)
2. Tolsma, J., de Wit, D.: Effectief Procesmanagement. Procesgericht Sturen met het BPM Model. Eburon, Delft (2009)
3. Hüffner, T.: The BPM Maturity Model - Towards a Framework for Assessing the Business Process Management Maturity of Organizations. GRIN, Munich (2004)

4. Lee, J., Lee, D., Kang, S.: An Overview of the Business Process Maturity Model (BPMM). In: Advances in Web and Network Technologies, and Information Management, June 16-18, Huang Shan, China, pp. 279--288. Springer: Heidelberg (2007)
5. Maier, A.M., Moultrie, J., Clarkson, P.J.: A Review of Maturity Grid based Approaches to Assessing Organizational Capabilities. In: Academy of Management Annual Meeting, 8-13 August. Academy of Management, California (2008)
6. Rosemann, M., de Bruin, T.: Application of a Holistic Model for Determining BPM Maturity. BPTrends (February 2005)
7. ISO/IEC: Information Technology - Process Assessment - Part 7: Assessment of Organizational Maturity - ISO/IEC TR 15504-7:2008(E). ISO/IEC, Geneva (2008)
8. Hammer, M.: The Process Audit. Harvard Business Review (April), 111--123 (2007)
9. Sheard, S.A.: Evolution of the Frameworks Quagmire. IEEEComputer 34(7), 96--98 (2001)
10. SEI: CMMI for Services, Version 1.2. Software Engineering Institute (February 2009), Online available, http://www.sei.cmu.edu/reports/09tr001.pdf
11. FAA: FAA-iCMM, Version 2.0, An Integrated Capability Maturity Model for Enterprise-wide Improvement (2001), Online available, http://www.faa.gov/about/office_org/headquarters_offices/aio/
12. OMG: Business Process Maturity Model (BPMM) - Version 1.0 (1 June 2008), Online available, http://www.omg.org/spec/BPMM/1.0/PDF
13. Houy, C., Fettke, P., Loos, P.: Empirical Research in BPM – Analysis of an Emerging Field of Research. Business Process Management Journal 16(4), 619--661 (2010)
14. Armistead, C., Machin, S.: Implications of BPM for Operations Management. International Journal of Operations & Production Management 17(9), 886--898 (1997)
15. Zairi, M.: Business Process Management: a Bounderyless Approach to Modern Competitiveness. Business Process Management Journal 3(1), 64--80 (1997)
16. Palmberg, K.: Exploring Process Management: Are There any Widespread Models and Definitions?. The TQM Journal 21(2), 203--215 (2009)
17. Gillot, J.-N.: The Complete Guide to Business Process Management. Booksurge Publishing, South Carolina (2008)
18. Smith, H., Fingar, P.: Business Process Management: the Third Wave. Meghan-Kiffer Press, Tampa (2002, 2006)
19. Lee, R.G., Dale, B.G.: Business Process Management: a Review and Evaluation. Business Process Management Journal 4(3), 214--225 (1998)
20. Gulledge Jr, T.R., Sommer, R.A.: Business Process Management: Public Sector Implications. Business Process Management Journal 8(4), 364--376 (2002)
21. McCormack, K., Johnson, W.C.: Business Process Orientation: Gaining the e-Business Competitive Advantage. St. Lucie Press, Florida (2001)
22. Aouad, G., et al.: Technology Management of IT in Construction: a Driver or an Enabler?. Logistics Information Management 12(1/2), 130--137 (1999)
23. Pritchard, J.-P., Armistead, C.: Business Process Management - Lessons from European Business. Business Process Management Journal 5 (1), pp. 10-32 (1999)
24. DeToro, I., McCabe, T.: How to Stay Flexible and Elude Fads. Quality Progress (March), 55--60 (1997)
25. Lee, J., Lee, D., Kang, S.: vPMM: a value based Process Maturity Model. In: Roger, L., Gonzu, H., Huaikou, M. (eds) Computer and Information Science 2009, pp. 193--213. Springer, Heidelberg (2009)
26. Maull, R.S., Tranfield, D.R., Maull, W.: Factors Characterizing the Maturity of BPR Programmes. International Journal of Operations & Production Management 23(6), 596--624 (2003)
27. Rohloff, M.: Case Study and Maturity Model for BPM Implementation. In: 7th International BPM Conference, September, pp. 128--142. Springer, Ulm (2009)

28. de Bruin, T., Rosemann, M.: Using the Delphi Technique to Identify BPM Capability Areas. In: 18th Australasian Conference on Information Systems, 5-7 December, pp. 642--653. Toowoomba (2007)
29. Skrinjar, R., Bosilj-Vuksic, V., Stemberger, M.I.: The Impact of Business Process Orientation on Financial and Non-financial Performance. Business Process Management Journal 14(5), 738--754 (2008)
30. Willaert, P., et al.: The Process-Oriented Organization: a Holistic View. Developing a Framework for Business Process Orientation Maturity. In: 5th International Conference on Business Process Management, 24-28 September, pp. 15. Springer: Brisbane (2007)
31. Bisnez Management: Business Process Management Onderzoek (2010), Online available, http://www.bisnez.org/
32. BPMInstitute: 2010 BPM Market Assessment Survey (2010), Online available, http://2010stateofbpm.surveyconsole.com/
33. BPT Group: Welcome to 8 Omega v2.0. BP Transformations Group (March 2008), Online available, http://bptg.seniordev.co.uk/8omega.aspx
34. Dowdle, P., et al.: The Process-based Management Loop. The Journal of Corporate Accounting & Finance (January), 55--60 (2005)
35. Dowdle, P., Stevens, J., Daly, D.: Process-based Management at Work in an Organization (2007), Online available, http://www.cam-i.org/docs/PBM_at_Work_in_an_organization.pdf
36. Champlin, B.: Dimensions of Business Process Change (25 June 2008), Online available, https://www.bpminstitute.org
37. Deloitte: Het Business Maturity Model (2010), Online available, http://www.deloitte.com/view/nl_NL/nl/diensten/consulting/
38. Ostolaza, E., Garcia, A.B.: EFQM/SPICE Integrated Model. In: International Conference on Product Focused Software Improvement, June 22-24, pp. 437--452. Oulu (1999)
39. Fisher, D.M.: The Business Process Maturity Model. A Practical Approach for Identifying Opportunities for Optimization. BPTrends (September 2004)
40. Gardner, R.A.: The Process-Focused Organization. A Transition Strategy for Success. ASQ, Quality Press, Milwaukee (2004)
41. Melenovsky, M.J., Sinur, J.: BPM Maturity Model Identifies Six Phases for Successful BPM Adoption. Gartner Research, Stamford (2006)
42. Harmon, P.: Evaluating an Organization's Business Process Maturity. BPTrends Newsletter 2(3), 1--11 (2004)
43. IDS Scheer: BPM Maturity Check (2010), Online available, http://www.bpmmaturity.com/
44. Oracle: BPM Lifecycle Assessment (2008), Online available, http://bpmready.nvishweb.com/
45. Remoreras, G.: Achieving the Highest Level of Process Culture Maturity (23 August 2009), Online available, http://mysimpleprocesses.com/
46. Rummler-Brache Group: Business Process Management in U.S. Firms Today (March 2004), Online available, http://rummler-brache.com/upload/files/PPI_Research_Results.pdf
47. Scavillo, M.: Business Process Transformation in the Software Industry (30 June 2008), Online available, http://www.sdn.sap.com/irj/scn/
48. Scheer, A.-W.: BPM =Business Process Management =Business Performance Management (2007), Online available, http://www.professor-scheer-bpm.com/
49. Smith, H., Fingar, P.: Process Management Maturity Models. BPTrends (July 2004)
50. Spanyi, A.: Beyond Process Maturity to Process Competence. BPTrends (June 2004)
51. Ayree, G., Naim, M.M., Lalwani, C.: Supply Chain Integration Using A Maturity Scale. Journal of Manufacturing Technology Management 19(5), 559--575 (2008)

52. Böhme, T.: Supply Chain Integration: A Case-based Investigation of Status, Barriers, and Paths to Enhancement. The University of Waikato, Waikato (2008)
53. Campbell, J., Sankaran, J.: An Inductive Framework for Enhancing Supply Chain Integration. International Journal of Production Research 43(16), 3321--3351 (2005)
54. Lockamy III, A., McCormack, K.: Supply Chain Maturity and Performance. In: McCormack, K. (ed.) Business Process Maturity. Theory and Application, pp. 105--135. Booksurge Publishing, South Carolina (2007)
55. McClaren, T.: A Measurement Model for Web-enabled Supply Chain Integration. In: Proceedings of the 19th Bled eConference eValues, pp. 1--13. Bled (2006)
56. Closs, D.J., Mollenkopf, D.A.: A Global Supply Chain Framework. Industrial Marketing Management 33, 37--44 (2004)
57. Netland, T.H., Alfnes, E., Fauske, H.: How Mature is your Supply Chain? Supply Chain Maturity Assessment Test. In: Proceedings of the 14th EurOMA Conference, pp. 1--10. EurOMA, Ankara (2007)
58. Riverola, J.: A General Approach to the Case Gathering Phase (2001), Online available, http://catalcatel.iese.edu/WEBSCM/Archivos/MaturityModel.pdf
59. Enkawa, T.: Logistics Scorecard (LSC) in Japan (May 2005), Online available, http://transportal.fi/Hankkeet/eglo/.383.pdf
60. AberdeenGroup: Global Supply Chain Benchmark Report (June, 2006), Online available, https://www-935.ibm.com/services/us/igs/pdf/aberdeen-benchmark-report.pdf
61. Anderson, D.L., Lee, H.L.: The Internet-Enabled Supply Chain: from the "First Click" to the "Last Mile". In: Anderson, D.L. (ed.) Achieving Supply Chain Excellence Through Technology, vol. 2, pp. 15--20. Montgomery Research (2000)
62. CGF: The Global Scorecard (2010), Online available, http://www.globalscorecard.net/
63. Ayers, J.B.: Supply Chain Project Management. A Structured Collaborative and Measurable Approach (2nd ed.). Taylor and Francis Group, Boca Raton (2010)
64. Chicago Consulting: Supply Chain Maturity. A Self-Assessment (2010), Online available, http://www.chicago-consulting.com/
65. Poirier, C.C., Quinn, F.J., Swink, M.L.: Diagnosing Greatness. Ten Traits of the Best Supply Chains. J. Ross Publishing, Fort Lauderdale (2010)
66. eKNOWtion: Supply Chain Maturity Monitor (2009), Online available, http://www.eknowtion.com/phpQ/fillsurvey.php?sid=8
67. IBM: The IBM Global Business Services 2007 Mainland China Value Chain Study (17 October 2007), Online available, http://www-935.ibm.com/services/us/gbs/thoughtleadership/#Next
68. van den Bergh, J.: Supply Chain Maturity Scan (2010), Online available, http://www.jvdbconsulting.com/supply-chain-maturity-scan.html
69. Boone, L.M., et al.: The GAIA Supply Chain Sustainability Maturity Model (2009), Online available, http://www.lmi.org/logistics/logistics.aspx
70. Holmes, J.: Building Supply Chain Communities (1997). Online available, http://criticalcomputing.com/
71. Cohen, S., Roussel, J.: Strategic Supply Chain Management. The 5 Disciplines for Top Performance. McGraw-Hill, New York (2005)
72. APQC: Supply-Chain Council SCOR-mark Survey (2007), Online available, http://www.apqc.org/scc
73. Schoenfeldt, T.I.: A Practical Application of Supply Chain Management Principles. ASQ, Milwaukee (2008)
74. Stevens, G.C. Integrating the Supply Chain. International Journal of Physical Distribution & Logistics Management 19(8), 3--8 (1989)

The Design of Focus Area Maturity Models

Marlies van Steenbergen[1], Rik Bos[2], Sjaak Brinkkemper[2], Inge van de Weerd[2], Willem Bekkers[2]

[1] Sogeti Netherlands, Wildenborch 3, 1112 XB Diemen, The Netherlands
Marlies.van.Steenbergen@sogeti.nl
[2] Department of Information and Computing Sciences,
Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands
{R.Bos, S.Brinkkemper, I.vandeWeerd, Bekkers}@cs.uu.nl

Abstract. Maturity models are a well-known instrument to support the improvement of functional domains in IS, like software development or testing. In this paper we present a generic method for developing focus area maturity models based on both extensive industrial experience and scientific investigation. Focus area maturity models are distinguished from fixed-level maturity models, like CMM, in that they are especially suited to the incremental improvement of functional domains.

Keywords: Design Research Methodology, Design Science, Enterprise Architecture, Software Product Management, Maturity Model.

This paper has been published as: Steenbergen, M. van, Bos, R, Brinkkemper, S, Weerd, I, van de, Bekkers, W.: The Design of Focus Area Maturity Models. In: W R. Winter, J.L. Zhao, and S. Aier (eds.): DESRIST 2010, LNCS 6105, pp. 317–332, Springer-Verlag Berlin Heidelberg (2010).

Within the field of IS new functional domains, like enterprise architecture and software product management, are still emerging. Developing such functional domains is a complex matter. Decisions have to be made about how and in what order to develop new processes, deliverables and competences. Maturity models are a well-known instrument to support incremental development of functional domains, as they distinguish different maturity levels that an organization progresses through.

In this paper we present a generic method to develop a particular kind of maturity model, the focus area maturity model. The focus area maturity model is particularly well-suited to support incremental development of functional domains as it departs from the concept of having a limited fixed number of generic maturity levels as used in CMM. Instead it defines maturity levels, called capabilities, per focus area within the functional domain. By juxtaposing all capabilities of all focus areas of a domain relative to each other, a balanced, incremental development path is defined. This juxtaposition of capabilities is done by positioning the capabilities in a matrix as shown in figure 1, which gives an example of a focus area maturity model in the functional domain of enterprise architecture. The capabilities are depicted by the letters A to D. Each capability is associated with a number of checkpoints. An architecture profile of a specific organization can be depicted by coloring the cells up to the capability that has not been implemented yet. The architecture profile provides insight into the strengths and weaknesses of the domain and where to focus improvement actions.

Focus Area / Maturity Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Development of architecture		A			B			C						
Use of architecture			A			B				C				
Alignment with business		A				B				C				
Alignment with the development process			A				B		C					
Alignment with operations					A			B			C			
Relationship to the as-is state					A				B					
Roles and responsibilities				A		B					C			
Coordination of developments							A			B				
Monitoring				A		B		C		D				
Quality management								A		B			C	
Maintenance of the architectural process							A		B		C			
Maintenance of architectural deliverables					A			B					C	
Commitment and motivation		A					B		C					
Architectural roles and training				A		B			C			D		
Use of an architectural method				A						B				C
Consultation			A		B				C					
Architectural tools							A				B			C
Budgeting and planning				A							B		C	

Fig. 1. A focus area maturity model for the functional domain of enterprise architecture.

Applying design science research methodology [1], we developed a generic method for developing focus area maturity models, depicted in figure 2.

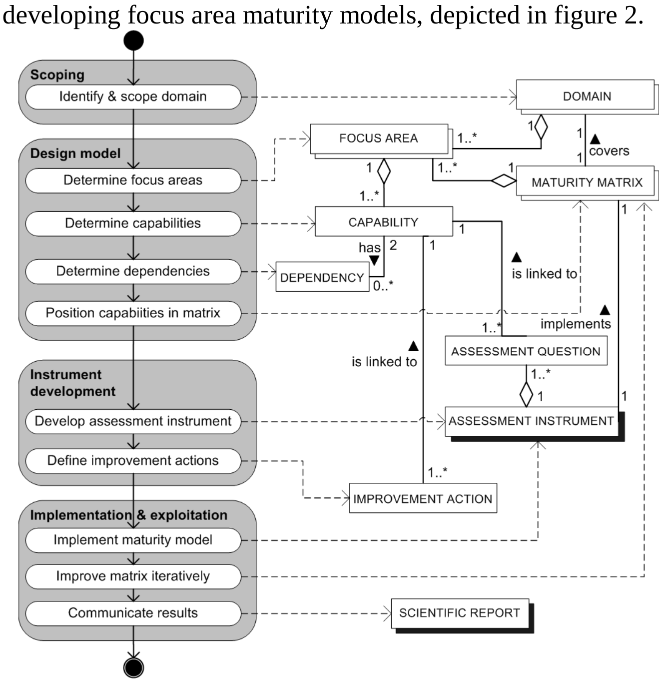


Fig. 2. The development method for focus area maturity models.

The development method is based on both literature review [2, 3, 4, 5] and extensive industrial experience in applying the focus area maturity model concept [6, 7, 8, 9] . The concept of the maturity matrix is refined by building a mathematical formalization of the matrix.

References

1. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, vol.24 (3), pp. 45-78 (2008)
2. Bruin, T. de, Freeze, R., Kulkarni, U., Rosemann, M.: Understanding the Main Phases of Developing a Maturity Assessment Model. In: Proceedings of the 16th Australasian Conference on Information Systems. Sydney (2005)
3. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: Proceedings of the 4th international Conference on Design Science Research in information Systems and Technology. Philadelphia (2009)
4. Maier, A.M., Moultrie, J., Clarkson, P.J.: Developing Maturity Grids for Assessing Organisational Capabilities: Practitioner Guidance. In: 4th International Conference on Management Consulting, Academy of Management (MCD'09). Vienna (2009)
5. Becker, J., Knackstedt, R., Pöppelbuß, P.: Developing Maturity Models for IT Management - A Procedure Model and its Application. Business & Information Systems Engineering 1(3), pp. 213-222 (2009)
6. Steenbergen, M. van, Berg, M. van den, Brinkkemper, S. A Balanced Approach to Developing the Enterprise Architecture Practice. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) Enterprise Information Systems. LNBIP 12, pp. 240–253 (2007)
7. Steenbergen, M. van, Schipper J., Bos, R, Brinkkemper, S: The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement. To appear in the proceedings of the 4th Workshop on Trends in Enterprise Architecture Research. Stockholm (2009)
8. Bekkers, W., Spruit, M., Weerd, I. v., Brinkkemper, S.: A Situational Assessment Method for Software Product Management. To appear in ECIS2010 proceedings (2010)
9. Weerd, I, van de, Bekkers, W., Brinkkemper, S.: Developing a Maturity Matrix for Software Product Management. To appear in Jansen, S., Tyrvainen, P. (eds.) Proceedings of the 1st International Conference on Software Business. LNBIP 51, Jyväskylä (2010)

Measuring Chain Digitisation Maturity: An Assessment of Dutch Retail branches

Marijn G.A. Plomp[1,2] and Ronald S. Batenburg[1],

[1] Utrecht University, Information and Computing Sciences, Padualaan 14, P.O. Box 80115, 3508 TC Utrecht, The Netherlands
[2] Dialogic Innovation & Interaction, Hooghiemstraplein 33-36, 3514 AX Utrecht, The Netherlands
{m.g.a.plomp, r.s.batenburg}@cs.uu.nl

Originally appeared in: Supply Chain Management: An International Journal, Vol. 15 No. 3 (2010), pp. 227–237

Keywords: Maturity model, Interorganisational information systems, Retail, SMEs, The Netherlands

Extended Abstract

The purpose of this article is to develop a validated measurement model and typology for chain digitisation maturity, defined as the degree of interorganisational collaboration through ICT.

The advantages of interorganisational information systems (IOIS) seem to meet the challenges currently facing the (Dutch) retail sector, as becomes clear from many large-scale examples. It can be seen as an omission that there are fewer examples for smaller organisations. There seems to be no clear insight into which factors drive small businesses to adopt and deploy IOIS. We depart from the notion that without considering the organisational dimension, the deployment of technology (i.e. ICT) will be less useful and/or effective. The common notion is that technological and organisational systems reinforce each other, as evidenced for example by theory on business–IT alignment. At every scale and level, technology and management (or ‘organisation’) should be related.

Literature discussing the level of chain digitisation often focuses on one single organisation. As chain digitisation exceeds the level of a single organisation, its maturity actually should be measured at the chain level as well. In this paper, we develop such a framework and validate the resulting measurement model at the level of interorganisational chains within a number of branches (i.e. sub-sectors of an industry).

We develop our integrated framework through a literature (meta) study, in which 22 existing maturity models are found and subsequently analysed. Our integrated framework (Fig. 1) incorporates the contents of many models as well as our specific findings with respect to model scope, domain focus, and the number of levels. We distinguish two dimensions: the level of technology and the level of organisation.

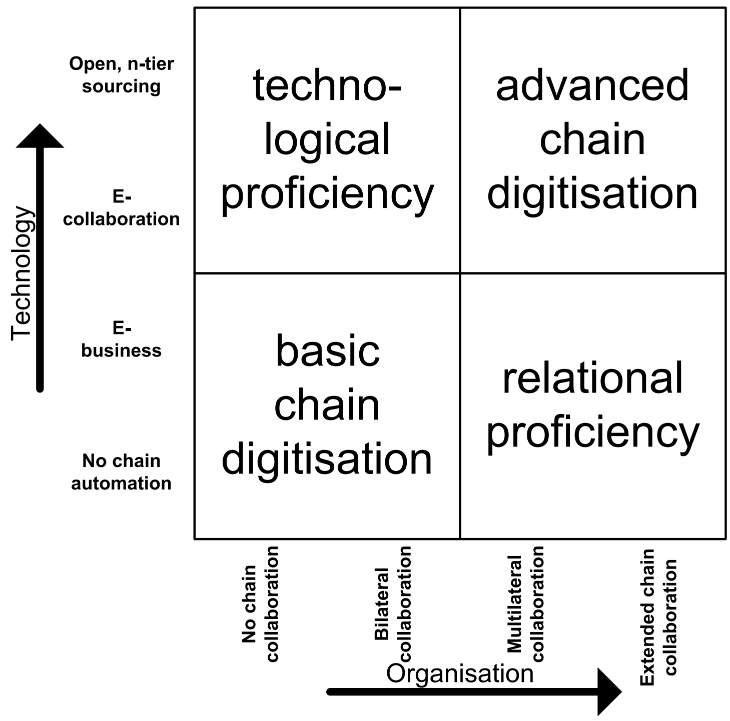


Fig. 1. A typology for chain digitisation.

We subsequently apply this typology to interorganisational collaboration within the Dutch retail sector (i.e. retailers and their wholesalers, manufacturers, customers, and trade organisations). The measurement model is tested by determining the chain digitisation level of 24 different retail sub-sectors (branches) through desk research, interviews, and surveys. Data are collected at the level of the branch, mainly through representatives of trade organisations.

As a result, the model appears to be applicable to describing the Dutch retail sector and comparing its branches, providing both expected and new insights. It is found that in general the level of chain digitisation of this sector – as of 2007 – is low: most branches are of the 'limited chain digitisation' type. Nevertheless, six branches are positioned within the 'relational proficiency' type.

The empirical application provides an extended view of the current situation of the (Dutch) retail sector with regard to chain digitisation. On this basis, a roadmap can be derived to support the adoption and deployment of chain digitisation among retail organisations.

Our framework for chain digitisation and the derived typology are of value to the SCM research community, as they are specifically developed and tested at the level of interorganisational chains. Here, the framework has been applied to the (Dutch) retail sector only. It seems suitable for application to other sectors as well.

Session 2:

Requirements Analysis

Building a Requirements Engineering Methodology for Software Product Lines

Tom Huysegoms, Monique Snoeck, Guido Dedene

Faculty of Business and Economics, Management Information Systems Group, K.U.Leuven
Naamsestraat 69, 3000 Leuven, Belgium
{Tom.Huysegoms, Monique.Snoeck, Guido.Dedene}@econ.kuleuven.be

Abstract. Software product lines are a great way to achieve reusability when they are correctly implemented. Theories about the product line paradigm already exist for multiple decades, but empirical research and reports of real life success stories are still scarce. Companies often still struggle to implement a software product line, because they don't possess the necessary knowledge and therefore do not sufficiently focus on the most basic aspect of a product line, namely the variability. Variability is the key to systematic and successful reuse, and should be considered as soon as possible in any software engineering project. The goal of the research is to develop a methodology for dealing explicitly with variability in software product lines during requirements engineering, because its impact will be maximal during this phase of software engineering. The methodology will be developed based on case-study research, in order to ensure practical relevance.

Keywords: Variability, requirements engineering, case-study research, harmonization.

1 Introduction

The financial crisis has intensified the strive for cost reduction. The software product line engineering (SPLE) paradigm offers a means to achieve this cost reduction inside the domain of software engineering through the systematic reuse of product line components. SPLE has already been studied for over two decades, software product lines are first mentioned in 1990 by Kang et al. as a part of their FODA specification [1]. A good definition of this software product line concept can be found in the work of Clements et al. [2]: "A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way". The SPLE paradigm requires a shift away from thinking about system development on a system by system basis as in traditional software engineering towards thinking in terms of development of product families. A product family is a somehow related group of software systems. SPLE focuses on these groups of related, highly alike software applications, and the concept of variability is used to situate the differences between these software applications.

Variability is one of the key concepts in SPLE. Without variability, the SPLE paradigm would not be capable to achieve a high level of cost reduction. The variability in the software product line allows the product line to be reused in similar, yet slightly different contexts. Although the exact same software system could be used in many different contexts, this would yield a suboptimal solution. Most of the time the software system would not fit its context perfectly, and additional efforts would be required to use it. In this way the cost benefits achieved by the reuse of a single application will be (partly) annihilated by the suboptimal results of the different implementations of the single system.

Despite the fact that variability is of utmost importance to the successful use of the SPLE paradigm, the absence of explicit reasoning about variability seems to be one of the reasons why a software product line project fails in practice. This observation was made in the exploratory study of some projects at an international bank and insurance company. The lack of a clear focus on variability during the early phases of SPLE doomed the projects even before they were fully implemented. The reason behind the fact that variability is overlooked in practice is not clear, but awareness about variability must be created in order to improve the results of any future SPLE projects. This is why we want to develop a methodology about the variability concept in SPLE, so that when a software product line is created, it will have a chance to succeed.

The paper consists of different parts. Section 2 defines the problem we wish to address and investigates related work. In section 3 we define the research plan and the type of research. In the sections 4 to 7 we then subsequently present the different aspects of the framework we wish to develop our methodology upon. Section 8 gives a summary and a conclusion, along with the direction of the future research.

2 Problem statement and related work

When we look at the state of the art literature of software engineering in general and requirements engineering more specifically, we establish the fact that there is a tendency to focus on the more theoretical aspects of research in favor of the empirical work [3]. This shortage in empirically based research (like e.g. [4]) can be one of the reasons why it is so difficult for companies to implement the theories developed in software engineering. It is not so that research results are unavailable or incomplete, but the fact that it is (perceived) difficult to use the developed theories in practice leads to reluctance to start up any new software engineering program or project, like for instance SPLE projects. This reluctance can be strengthened by the fact that most new software engineering projects require additional investment funds to be gathered. In the SPLE domain assets need to be developed and this generates extra costs at startup, while benefits are only achieved when the assets are reused multiple times (at least more than once). The benefits are uncertain and are located in the future, while the costs are certain at the starting point. In combination with the fact that the development and implementation of a software product line is perceived as difficult, implementing such a line is not an obvious choice.

Once the initial investment decision is taken and the company commits itself to the SPLE paradigm, the product line itself is created and implemented. Also then it is likely that problems will arise. Due to the lack of sufficient empirical guidelines like [5] and [6], many of the problems a company can possibly face during the creation or implementation are not well documented. Without any source of information on how to solve practical problems, the implementation of the product line is hampered at best. In the worst case, the problems that arise make the companies put a stop to their efforts in applying the SPLE paradigm and return back to the single system engineering approach. One could argue that because SPLE already exists for a long time, there ought to be solutions somewhere in the large amount of literature about it created over the years. The fact is however that many companies cannot rely on someone who is very proficient in SPLE literature, otherwise they would not be having problems with SPLE in the first place. For people who are less proficient in SPLE in particular or in software engineering in general, the complicated literature does not offer a suitable ready-to-use solution which is efficient but at the same time easy enough to grasp by anyone less than an expert.

The goal of our research is to provide the much needed bridge between the theory of SPLE and its implementation in practice [7] by offering a methodology that is usable for and encompasses all stakeholders who have a role in SPLE, from IT staff to business managers. At the center of this methodology there is the concept of variability [8], as this is the foremost driver of a successful SPLE [9]. Decisions about variability need to be taken with care, as they can impede the reuse of the software assets later on. In order to minimize the impact of implicit decisions, it is best to start thinking about variability as early as during the requirements engineering. Therefore the methodology will focus itself mainly on the requirements engineering phase of the software engineering process. How we are planning to create the methodology in concrete is explained more in detail in the next section.

3 Research method and plan

In order to ensure the practicality of our Variability enabling Software Product line and Requirements Engineering Methodology (VeSPREM) we create VeSPREM based on empirical research in the form of case studies. Starting from a theoretically grounded framework which provides the basic elements that are necessary to reason about variability, we fill in this framework based on multiple case studies conducted in two software intensive multinationals. These multinationals either develop their own information system in house (internal software provider model) or develop software in order to sell it to other companies (external software provider model). The research will focus first on the internal software provider model, because the context of the internal software provider model is likely to be more controllable than the context of the external software provider model. Most of the time the stakeholders in an internal software provider context are well known while in the external model future customers are unknown at design time. The size of the cases is also likely to be smaller in an internal context because the number of possible customers is limited to the number of internal departments who may request for software systems.

In current research [10] case study research is becoming more and more popular and is considered as a full-fledged research method. It is a qualitative research method, based on real life cases. The big advantage of case study research is that it is always based on the analysis and interpretation of empirical evidence, and therefore the link with practice is ensured. Other research methods, for instance ethnographic studies, have the same link with real life, but they are less usable in our research context because they require such long periods of time to be successfully conducted that conducting several case-studies (which we need) within a reasonable amount of time is too labor intensive for a single researcher. A drawback to case study research is claimed to be the lack of possibilities to quantitatively analyze the results of the research. The lack of a formal way to test the developed hypotheses can seem a drawback, but there are possibilities to ensure the validity of the research. Multiple case studies can be conducted, in order to create a form of replication. Yin [11] describes two forms of replication: literal replication and theoretical replication. Literal replications predict similar results in order to strengthen the conclusions from the case studies. Theoretical replications predict other results than the original cases but in such a way that the differences are anticipated. In our research literal replication will be done by conducting multiple case studies in an internal software provider context, while theoretical replications will be done by studying cases in an external software provider context (in contrast to the internal software provider context).

To start with, an initial theory needs to be developed in order to achieve the right focus for the case studies [11]. In our research, this starting theory takes the form of a framework that will then be systematically enriched with the results of the case studies in order to create a complete methodology. The initial framework that we will use as starting point of our research is presented in the following sections (section 4 to section 7). It combines the results of a survey of existing theoretical research and an exploratory observational case study in a multinational banking and insurance company. The framework represents the basic ideas and assumptions about variability in SPLE taken from literature, and extends them with findings from the observational case study. The planned case studies will provide substantial material to enrich this framework with the necessary details, so that the framework can grow organically into a full-fledged methodology. A feedback loop will then trigger a revision of any previous cases against the altered/extended framework. By forming a feedback loop the construct validity of the research will be guarded. The next four sections represent the pillars of the framework: the structure of the IT department providing the product line, the decision process concerning variability in requirements, the representation of variability in goals, features and architectural specifications and the engineering process in SPLE as a whole.

4 An multi-organizational context for IT

SPLE can be applied in many contexts. The methodology we will develop aims at developing management information systems for software intensive organizations. This setting is very different from a setting of e.g. developing a product line for embedded software for cars or cell-phones. The need for variability arises from the multinational, multi-divisional or multi-institutional property of the organization (a 'multi-' organization or 'multi-' company). Each business unit has very similar but nevertheless also very particular needs. As a result of this, the IT department needs both centralization and decentralization.

Central guidance is needed in order to obtain an overview of the total IT portfolio. This overview is needed if opportunities for reuse are to be spotted. These opportunities are marked by a significant overlap of functionality between different applications. This overlap is what is called the commonality of a product family.

Simultaneously, the decentralized view on the IT department is required to cope with the variability issues of the SPLE. Each of the geographically (or logically) divided regions has its own specific properties and domain assets need to be adjusted accordingly in order to obtain a usable implementation. The combination of both centralization and decentralization can be achieved if we assume that the IT department is structured according to the blueprint described below. The proposed blueprint is based on the Integrated Architecture Framework (IAF) of Capgemini and the Asset and Solution Framework (ASF) of KBC ICT Services studied in the exploratory case study. The proposed IT blueprint is described in Fig. 1 and represents the structure for the whole IT department of a software intensive 'multi-'company.

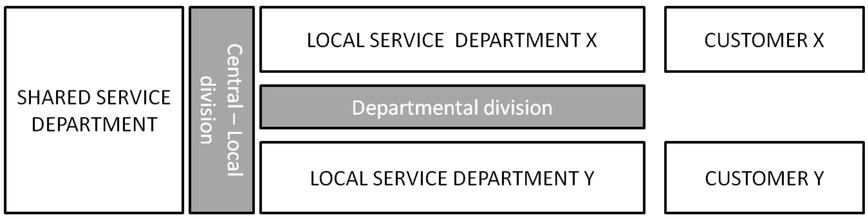


Fig. 1. The 'multi-' company IT department

On the left side of the figure we see the shared service department (or central IT department) which coordinates the whole IT structure and has an overview over all the IT projects in the different local service departments (or local IT departments). These local departments each provide software for their specific customer (or group of customers). Depending on the type of 'multi-' company, the customer can be a business line, a business unit, a market segment,… Each local department is thus coupled with a customer(group). The task of the local departments is to provide customized software systems to the customer they have been assigned to. The division between the central and the local departments on the one side, and the division between the local departments on the other side are both necessary in order to ensure the scalability of the department. If the different tasks for the different parts of the IT department are not clearly defined, it will become difficult to increase the size of the IT department.

Although other organizational blueprints are definitely possible, we opt for this construct because it is used by both existing companies and suggested by consulting firms. The most important aspects of the blueprint are the actors and their responsibility. In the next section we will elaborate our framework based on the organizational actors presented here.

5 Variability in software product line requirements

The second part of our framework concerns the variability of software product line requirements. As mentioned before, variability is a central concept in SPLE and therefore central to our starting framework. Reasoning *explicitly* about variability is something that really needs to be done if you want a SPLE project to have any chance on success. The earlier in the software engineering lifecycle decisions about variability are made explicit, the more effect they will have. Ignoring variability altogether makes it likely that implicitly some decisions concerning variability are already made. These decisions may not always be optimal, the risk of taking suboptimal decisions increases if variability is not explicitly dealt with. Besides the fact that wrong decisions will be made and extra costs will incur to correct these mistaken decisions, the total cost amount will also rise if the mistakes are only corrected later on in the process. Correcting the architecture of a domain asset before it is created is much cheaper than correcting all the implementations that are already built upon a faulty domain asset separately.

Our methodology will support making decisions about variability explicitly and as early as possible. During the requirements negotiation with the different stakeholders of a product line, it is easy to explicitly interact about variability. When multiple stakeholders have different demands and requirements for the information systems to be developed, these differences can be identified and reasoned upon. Reasoning about variability during requirements engineering is split up into two phases.

The first phase is the phase during which all stakeholders define their requirements and based on these requirements it is decided which requirements are common for all stakeholders (the commonality of the requirements) and which ones are different (the variability of the requirements). During this decision process differences between requirements can be discussed, in an attempt to slightly alter the requirements so they will possess more commonality and less variability. The drive for this harmonization step is the fact that the commonality part can be reused without making extra costs.

During the second phase it is decided which parts of the variable requirements will be implemented in the domain asset as variability (that is supported by the domain asset) and which variable requirements are left to the local service departments to implement themselves on a customer by customer basis. Supported variable requirements are those requirements that the central (shared service) department agrees to support. The shared service department implements these variable requirements as variation points [8] in the architecture of the domain assets. These variation points can be filled in by any of the variants that are offered by the central department. Therefore it can be said that the variable requirements that are supported in the domain asset are available for every implementation, and that that part of the variability is shared by all implementations. The other part of the variable requirements is specific to some stakeholders and therefore these are not supported centrally. The reason for supporting some variability centrally is once again for cost reduction purposes. The second phase is called variabilization, because this part of the decision process concerns defining up to what point variability is supported by the shared service department.

The following figure (Fig. 2) visualizes the variability decision process during requirements engineering for a product line. Each phase is visualized by its respective arrow next to the octagonal, which represents the total collection of requirements. On the left, with arrow number 1, we have the harmonization phase which decides on the amount of commonality and variability in the requirements. On the right we have the variabilization phase, labeled number 2. It is during this phase we decide which part of the variable requirements is shared in the form of variation points in the domain asset, and which part of the variable requirements is specific for certain stakeholders and left to be implemented by the local service departments.

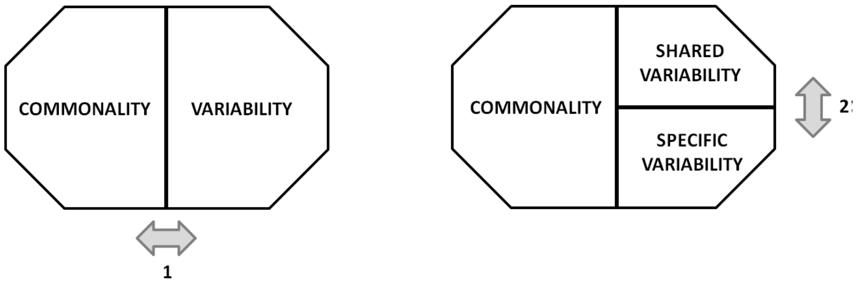


Fig. 2. Requirements harmonization (1) and variabilization (2)

6 Variability in model specifications

In the previous section we made explicit decisions about variability, but the results of these decisions need to be written down. Different modelling techniques available in literature describe the requirements of a software product line. These modelling techniques all represent variability in their own way. The problem however is linking these different kinds of representations correctly so that they represent the same variability. Common to the techniques is the fact that they represent some kind of decomposition possibility and that the variability is incorporated in it. We focus on feature and goal models, as they are the most popular requirements modelling techniques in early requirements engineering.

Probably the best known product line modelling technique is the feature model [12]. A feature model visualizes all possible feature combinations for one product line. A feature is a logical grouping of requirements which represent a certain aspect of a software system. The visualization of the features is done by a feature diagram [12]. This is an acyclic graph, and most of the times even a hierarchical tree. Features can be classified into groups in different ways. A first classification can be done based on the functionality which is represented. A feature can represent a group of requirements which are either functional or non-functional [13]. Functional requirements and features demand the development of some functionality of the information system. Non-functional requirements and features represent demands that link to certain performance or service levels, but do not demand any functionality.

The second manner of feature classification is more interesting in the context of variability and is based on the necessity of the feature in its context of the software product line. There are three different categories of features in a feature model based on their necessity: mandatory features, optional features and alternative features. These different kinds of features all have a distinct visualization in the feature diagrams so that they are easy to identify. The optional and the alternative features are the ones that represent the variability in a software product line, and the link with variability is exactly what we are looking for.

The second technique which is gaining more and more interest in current research on requirements engineering is goal modelling. According to van Lamsweerde [14], a goal is a prescriptive statement of intent that the system should satisfy through the cooperation of its agents. A goal is thus something that needs to be attained by certain system components in order to be fulfilled. A system component can be a human actor with a specific role, a software component, a measuring device (e.g. a sensor),… Variability in goal models is represented by alternative refinements.

The difference between a requirement and a goal is the following: a requirement is a prescriptive statement that needs to be enforced by the software system being developed, a goal is a prescriptive statement that can be enforced by any system component. Goals are therefore useful in a context where there are business stakeholders (or other non IT stakeholders) present, while requirements present the possibility for IT stakeholders to focus on the software system which needs to be developed. In a SPLE project both modelling techniques have their advantages, so both should be used. When using both modelling techniques, one should be careful that both techniques represent the same variability. Research about linking goal models and feature models is thus needed in order to ensure that when variability is explicitly dealt with, it is being represented correctly. Even during negotiations before decisions are taken, it is beneficial to link feature and goal models, to facilitate the negotiation process, hence the inclusion in our initial framework.

7 Software product line engineering with explicit variability

The last theoretical pillar of the initialframework concerns the total engineering process of a software product line. In the previous section we stressed the importance of requirements engineering, but all other phases in the engineering process need to cope with the identified variability. Typically SPLE is divided into two process cycles, namely domain engineering and application engineering [15]. The domain engineering cycle is concerned with the development of domain assets. Once the domain assets are developed, these assets can then be reused in the application engineering cycle each time an instance of the product family needs to be developed.

The VeSPREM framework slightly alters the traditional 2 cycle development by extending it with one extra cycle. The first cycle is the same as the domain engineering cycle, during which the assets are developed with special attention devoted to variability. The second cycle is the application instantiation cycle. This cycle roughly matches the traditional application engineering cycle. The variability points in the domain asset are bound with specific variants and the asset is then localized (or implemented specifically) for the customer in question. The third and last cycle is an

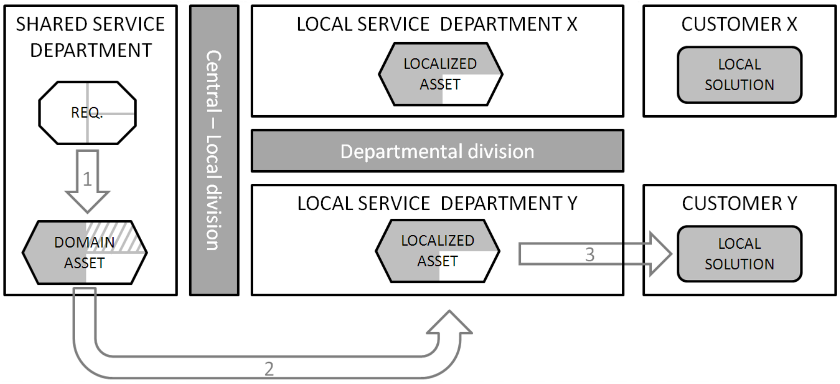


Fig. 3. Product line engineering cycles

additional cycle not included in the traditional bi-cycle process which deals with the customer specific extension of the now localized asset by the local departments. In traditional SPLE this last cycle is (implicitly) part of the application engineering cycle, but we suggest making this distinction because the extending activity clearly differs from the localizing activity. Localizing is done in the same way with the same set of possible variants for all product line members, while extensions are de facto member specific. Because different rules and forces can apply in both phases they should be split up instead of being considered as one application engineering step.
In Fig. 3 the three cycles are represented by numbered arrows. Number 1 is the domain asset cycle where the common part and the supported variability part are created, number 2 is the localizing cycle where the supported variability is filled in depending of the context and number 3 represents the extension cycle where local additions are made. The cycles are mapped against the IT department structure of section 4 in order to show the natural mapping of it on the cycles.

8 VeSPREM: a conclusion and look at the future

In the previous sections we have described each pillar of our framework. As presented in this paper, the framework only serves as a starting point for the case studies we plan to conduct and by no means yet fixed nor claimed to be complete. One possible extension of the framework which will be studied during the case studies is the possibility to create a formal method for transformations between goal models and requirement models. Another addition to the framework currently under study is the extension from traditional SPLE towards Service Oriented Product Line Engineering (SOPLE). Further extensions will also definitely arise when we are coupling the case study results back to the framework. The longer term objective of our research is to enhance and fill the framework to such a level we obtain a solid and complete methodology concerning variability inside software product lines.

As made clear by the explanation of the pillars of the framework, the theory for successful SPLE is available in research, but the link towards practice still lacks strength, and this is where VeSPREM will ultimately prove useful. Once the methodology is completed, the following step will be to introduce VeSPREM into companies and in that way create new opportunities for further explorations in empirical research.

Acknowledgements

We gratefully acknowledge KBC Global Services NV (member of the KBC Group) for funding this research through the KBC research chair "Developing and Managing Business Services as Shared Assets".

References

1. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-Oriented Domain Analysis (FODA): feasibility study. Software Engineering Institute, Carnegie Mellon University (1990)
2. Clements, P., Northrop, L.: Software product lines: practices and patterns. Addison-Wesley Longman Publishing, Boston (2001)
3. Alves, V., Niu, N., Alves, C., Valença, G.: Requirements engineering for software product lines: A systematic literature review. Information and Software Technology 52, pp. 806 -- 820 (2010)
4. Hubaux, A., Heymans, P., Benavides, D.: Variability Modelling Challenges from the Trenches of an Open Source Product Line Re-Engineering Project. In International Conference on Software Product Lines, pp. 55 -- 64, IEEE Computer Society, Limerick (2008)
5. Linden, F. J., Schmid K., Rommes, E.: Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer-Verlag, Berlin (2007)
6. Lee, K., Kang, K.C., Lee, J.: Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. In International Conference on Software Reuse, pp. 62 – 77, Springer-Verlag, Austin (2002)
7. Chen, L., Ali Babar, M., Ali, N.: Variability management in software product lines: a systematic review. In: Proceedings of the 13th international Software Product Line Conference, pp. 81 -- 90, Carnegie Mellon University, San Francisco (2009)
8. Kim, S.D., Her, J.S., Chang, S.H.: A Theoretical Foundation of Variability in Component-Based Development. Information and Software Technology. 47, pp. 663 -- 673 (2005)
9. Svahnberg, M., van Gurp, J., Bosch, J.: A Taxonomy of Variability Realization Techniques. Software Practice & Experience 35, pp. 705 – 754 (2005)
10. Travers, M.: Qualitative research through case studies. Sage Publications Inc., Los Angeles (2001)
11. Yin, R.K.: Case Study Research: Design and Methods. Sage Publications Inc., Los Angeles (2009)
12. Czarnecki, K., Eisenecker, U. W.: Generative Programming: Methods, Tools, and Applications. ACM Press/Addison-Wesley Publishing Co., New York (2000)
13. Robertson, S.,Robertson, J.: Mastering the Requirements Process. ACM Press/Addison-Wesley Publishing Co., New York (1999)
14. van Lamsweerde A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, Chichester (2009)
15. Pohl, K., Böckle, G., Linden, F. J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, New York (2005)

A Survey of Variability Management Requirements

Marco Aiello, Pavel Bulanov, and Heerko Groefsema

Johann Bernoulli Institute for Mathematics and Computer Science
University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands
{m.aiello, p.bulanov, h.groefsema}@rug.nl

Business process management (BPM) is impacting medium and large scale enterprises enormously these days. Designed to support rigid and repetitive units of work like production processes, business process models offer little in the area of flexibility and reuseability [1, 3]. By introducing variability to the world of BPM, many new possibilities are introduced.

An example comes from eGovernment. In the Netherlands there are 430 municipalities that have to implement the same national laws, though, they are different in size, business models, IT infrastructures and so on. Recently the WMO law (Wet maatschappelijke ondersteuning, Social Support Act, 2007) was approved that mandates, for instance, the rules for providing publicly subsidized wheel chairs to needing citizens. All municipalities have to implement this process, each with slight but clearly noticeable differences related to their organizational and IT structure [5]. The recurrence of the need to adapt processes to instances and changes become concrete with the notion of *variability* [4], which first emerged in software engineering. Variability in this case refers to the possibility of changes in software products and models.

In [2], of which this text is an abstract, we propose to use variability in order to take full advantage of the obvious reusability opportunities in such situations. Variability in this context, namely that of BPM, indicates that parts of a business process remain variable, or not fully defined, in order to support different versions of the same process depending on the intended use or execution context. Such variability is often included through the introduction of so-called *variation points*, that is, elements of a business process where change may occur. A process in which variability is included is called a *reference* or *generic process*. Processes where choices have been made deriving from the reference process are called *variants*.

We introduce *variability management* as an extension of the typical activities involved in business process management. We give a general depiction in Figure 1. On the left, we notice how requirements drive the definition of the design processes. Variability management complements these general BPM phases by introducing a set of parallel stages, on the right in the figure. In this context, two main stages are introduced: *design-time* and *run-time* variability.

In [2], after defining variability in business process management, we consider the requirements for explicit variation handling for (service based) business process systems. eGovernment serves as an illustrative example of reuse. Finally, an

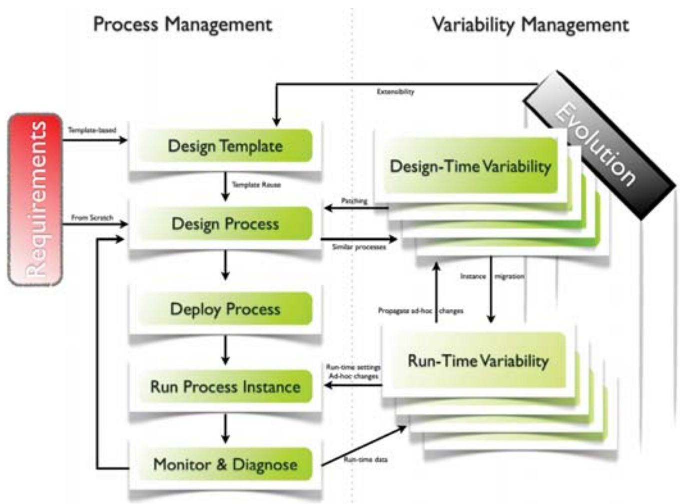


Fig. 1. Process lifecycle and variability management.

evaluation of existing tools for explicit variability management is provided with respect to the requirements identified. A video illustrating a first prototype to manage service-based business processes with explicit support for design time variability is available at `http://www.sas-leg.net/web/index.php?n=Main.Demo2010`.

Acknowledgements

The research is supported by the NWO Jacquard program via the **SaS-LeG** project, `http://www.sas-leg.net` contract 638.000.000.07N07. We thank P. Avgeriou, N. van Beest, F. van Blommestein, A. Lazovik, D. Tofan, and H. Wortmann for fruitful discussion.

References

1. van der Aalst, W., Jablonski, S.: Dealing with workflow change: Identification of issues and solutions. International Journal of Computer Systems, Science, and Engineering 15(5), 267–276 (2000)
2. Aiello, M., Bulanov, P., Groefsema, H.: Requirements and tools for variability management. In: Requirement Engineering for Services (REFS 2010) at COMPSAC. IEEE (2010)
3. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems: A survey. Data and Knowledge Engineering 50(1), 9–34 (2004)
4. Sinnema, M., Deelstra, S., Hoekstra, P.: The covamof derivation process. In: ICSR. pp. 101–114 (2006)
5. Sun, C., Rossing, R., Sinnema, M., Bulanov, P., Aiello, M.: Modelling and managing the variability of web service-based systems. Journal of Systems and Software, Elsevier 83, 502–516 (2010)

Current Status and Future Perspectives of Drug Information Systems

Tommi Tervonen[1], Bert de Brock[2], Pieter A de Graeff[3],. Hans L Hillege[4]

[1] Econometric Institute, Erasmus University Rotterdam, The Netherlands, tervonen@ese.eur.nl

[2] Faculty of Economics and Business, University of Groningen, PO Box 800, 9700 AV Groningen, The Netherlands, e.o.de.brock@rug.nl

[3] Department of Internal Medicine/Clinical Pharmacology, University Medical Center Groningen, PO Box 30.001, 9700 RB Groningen, The Netherlands, p.a.de.graeff@int.umcg.nl

[4] Department of Epidemiology/Cardiology, University Medical Center Groningen, PO Box 30.001, 9700 RB Groningen, The Netherlands, j.l.hillege@tcc.umcg.nl

Extended Abstract

We consider the current status and future perspectives of Drug Information Systems (DISs) from an IS point of view. The information lifecycle of a drug starts already during the drug discovery phase and continues far into the future in prescription- and adverse-effect databases. The drug lifecycle consists of the disjunct phases of discovery, development, approval, and marketing.

Each phase is supported by various types of DISs. We distinguish the following types of DISs: *compound databases* (containing the physico-chemical structural data for the computational drug discovery methods), *(pre-)clinical trial databases* (containing the - raw or aggregate - data of (pre-) clinical trials), *SmPC databases* (containing Summary of Product Characteristics, the source of information visible to non-professionals through drug labelling and package inserts), *ADR databases* (containing data on Adverse Drug Reactions), and *CPOE systems* (Computerized Physician Order Entry systems, automating the human error-prone parts of the process of the prescribing physician, especially with regard to drug prescription).

In this paper, we describe the past literature and existing technology of Drug Information Systems. We develop a mapping of DISs to the phased drug lifecycle, taking into account the system information contents. The mapping shows that currently there is a lack of DISs providing efficacy- and safety-data in a suitable format. This lack severely hinders the possibility of physicians, researchers, as well as regulatory authorities and the pharmaceutical industry, to make quantitative analyses of efficacy and safety over a wide range of drugs.

Drug development, testing and administration are information-intensive areas with varying computing needs. These range from storing a single drug's labeling information to complex algorithms for analyzing quantitative structure-activity relationships in the drug discovery process. We use the term Drug Information System (DIS) for any system that stores data related to *some* phase(s) of the drug

lifecycle, and that processes it into user relevant information. DISs have various uses in, for example, recording clinical trial results, disseminating findings of adverse drug reactions, and operational support in a hospital environment. Although the amount of clinical health research is growing rapidly, the supporting operational systems have only recently received comparable effort in research.

The past few years have seen a rise in the amount of research in DISs, but the majority of health information systems literature seems to concentrate on DISs as a tool to improve drug prescription processes and focus on health care safety in a clinical environment. However, in order to support clinical pharmacological decision making and to access and aggregate information from the complete drug lifecycle, it is crucial to have an overview of existing DISs.

A clear overview of existing DISs enables processes dealing with information gaps between discovery, development, regulatory approval, and pharmacovigilance stages. The main motivation to fill these gaps is the need for a reform of the regulatory process recently brought into discussion by regulatory bodies, academia, and industry. In order to improve management of drug information, we need an overview of DISs nowadays available and contributing to the drug lifecycle. More structured information will lead to improved transparency in the decision making process of regulatory authorities. There exists evidence that even published clinical trial results have had statistical evidence interpreted incorrectly in order to appear positive. Transparency of the process could help to find such incorrect analyses as the original studies would be linked with the aggregated results, and finally, with the marketing authorization decisions that (in principle) take into account all relevant clinical data.

This survey considers the existing literature and DISs from a perspective that is, up to our best knowledge, new in the area. We review the existing technology and DISs from a functional point of view, providing an overview of the current state of the technology. Although we briefly present drug discovery systems, we concentrate on clinical data from drug development. The review is followed by a mapping of the existing systems to the various phases of the drug lifecycle. This mapping should be taken as a starting point for information integration across DISs. The mapping allows finding possible integration points between DISs of different phases, which can eventually lead to information re-use, to improved communication, and following this, to shortened drug development cycles.

The meta-analytical approach as applied in the Cochrane Library seems to be the most appropriate starting point for building the next generation DISs for regulatory uses. The future systems should store all required measurements in a numerical format with strict semantics. For aggregate clinical trial results, we are currently working on building such a system (see http://www.drugis.org).

- This study was performed in the context of the Escher project of the Dutch Top Institute Pharma.

- The full paper appeared in the proceedings of ECIS 2010 (Pretoria), see http://web.up.ac.za/ecis/ECIS2010PR/ECIS2010/Content/Papers/0004.R1.pdf.

Session 3:

Collaborative Modelling

Value-Sensitive Design for Cross-Enterprise Regulation*

Sietse Overbeek, Virginia Dignum, and Yao-Hua Tan

Faculty of Technology, Policy and Management, Delft University of Technology,
Jaffalaan 5, 2600 GA Delft, The Netherlands, EU
{S.J.Overbeek,M.V.Dignum,Y.Tan}@tudelft.nl

Over the past several years, the business community has devoted considerable attention to corporate responsibility, in order to address significant social and environmental questions with value for business and society. Organizational activities are expected to be transparent to governments, investors, and other stakeholders. Enterprises, governmental institutions, and the public in general benefit from well-defined and well-enforced laws and legal guidelines, in order to protect companies from manipulations of financial reporting data. Traditionally, control and enforcement were government tasks, however, the advent of sound IT support and the increasing cost and complexity of regulation are leading towards collaborative regulation between enterprises and governments.

Regulation of organizational processes is based on the norms that organizations have to comply with. A *norm* can be defined as standard behavior that is acceptable for the regulating institutions, indicating desirable behaviors that should be carried out as well as undesirable behaviors that should be avoided [4]. *Norm enforcement* mechanisms are used to determine if organizations have complied to the *norms* that they should satisfy [1]. If norms are to be enforced, then the institution should specify and handle sanctions for every possible violation of the norms. This means that enforcement mechanisms often require the introduction of special 'regulator actors' that actively monitor the behavior of the other agents [1]. Such agents are assigned to monitor the behavior of organizations and sanction them in case of norm violations. Implementing self-regulation as a control mechanism results in a redistribution of control tasks among the actors.

Which enforcement mechanisms are effective and how sanctions are likely to be followed is directly related to the values of an organization. Moral values are the standards of good and evil that guide an individual's behavior and choices [5]. Individuals, groups, and societies develop own value systems used for the purpose of ethical integrity. The value notion and the two mentioned different types of norm enforcement mechanisms can be combined to design a *value-sensitive system* that supports agents in *norm fulfillment* and *norm enforcement*. Value Sensitive Design (VSD) is a methodological design approach that aims at making moral values part of technological design, research, and development [2]. Values are typically high-level abstract concepts that are difficult

*Published as: Overbeek, S., Dignum, M., Tan, Y.H.: Value-sensitive design for cross-enterprise regulation. In: Proceedings of the 1st International Workshop on Cross Enterprise Collaboration, People, and Work (CEC-PAW10) held in conjunction with the BPM 2010 Conference, Hoboken, New Jersey, USA, September 13, 2010.

to incorporate in software design. In order to design systems that are able to deal with moral values, norms must be operationalized while maintaining traceability of its originating values. This change calls for architectures that satisfy the following principles: (1) coordination policies need to be described at a high level of abstraction; (2) the enforcement needs to be negotiated between governments and enterprises; (3) coordination policies need to be formulated explicitly rather than being implicit in the interactions; and (4) it should be possible to deploy and enforce a policy incrementally.

An increasingly important value in organizations is that of *ethical and transparent business practices*. The development of codes and standards for ethical and transparent business practices can help limit corruption, ensure fair and open competition, and encourage a better business environment. A formalism for values must be able to describe and reason about social structures and interactions, facilitating analysis and verification through logical reasoning. Moreover, in open systems where agents are assumed to be autonomous and rational, agents can, involuntarily or by deliberate choice, violate social norms and regulations and therefore one must be able to deal with and reason about such violations. In normative systems, interactions between actors are regulated by normative templates that describe desired behavior in terms of deontic concepts such as obligations, prohibitions, permissions, deadlines, violations and sanctions [3]. Deontic logic provides mechanisms to reason about violability of norms, that is, about how to proceed when norms are violated.

The results of our research provide the basis for a value-sensitive system to support actor agents in norm fulfillment and regulating agents in norm enforcement. This foundation has been laid by applying a value-sensitive system development process and by incorporating the principles of the norm enforcement mechanisms of direct control and self regulation in the system design. By following this specific system development process, the value that is created for the agents that apply the norm enforcement mechanisms of direct control and self regulation is explicitly incorporated in the development of the system.

References

1. Burgemeestre, B., Hulsteijn, J., Tan, Y.H.: Towards an architecture for self-regulating agents: a case study in international trade. In: Proc. 2nd Multi-Agent Logics, Languages, and Organisations Federated Workshops, Italy. CEUR Workshop Proceedings, vol. 494 (2009)
2. van den Hoven, M.: Design for values and values for design. Information Age +, Journal of the Australian Computer Society 7(2), 4–7 (2005)
3. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Journal of the IGPL 4(3), 429–445 (1996)
4. Meneguzzi, F., Luck, M.: Norm-based behaviour modification in bdi agents. In: Sierra, C.e.a. (ed.) Proc. 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Hungary. vol. 1, pp. 177–184 (2009)
5. Scott, E.: Organizational moral values. Business Ethics Quarterly 12(1), 33–55 (2002)

Challenges of Involving Stakeholders When Creating Enterprise Architecture

A. (Agnes) Nakakawa[1], P. (Patrick) van Bommel[1], and H.A. (Erik) Proper[1,2]

[1]ICIS, Radboud University Nijmegen
P. O. BOX 9010 6500, GL Nijmegen, The Netherlands
[2]CITI, CRP Henri Tudor Luxembourg, Luxembourg
A.Nakakawa@science.ru.nl, pvb@cs.ru.nl, e.proper@acm.org

Abstract. Although researchers report challenges that occur during enterprise architecture development (in general), there is lack of an elaborate description of those that occur during enterprise architecture creation – particularly if organizational stakeholders are to be deeply involved. Yet understanding challenges of involving organizational stakeholders when creating enterprise architecture is a prerequisite for devising a relevant solution to enterprise architects. An exploratory survey was therefore conducted with the aim of investigating challenges that enterprise architects face when they involve organizational stakeholders during enterprise architecture creation. This paper presents and discusses findings from the survey. The survey results generally indicate why 90% of enterprise architects face challenges when delivering products of enterprise architecture creation, although 96% of architects closely collaborate with organizational stakeholders during enterprise architecture creation.
Keywords: *Enterprise Architecture Creation, Stakeholder Involvement.*

1 Introduction

Since architecture "*is the normative restriction of design freedom*" [2], enterprise architecture can be conceived as a normative instrument (in the form of principles, views, and models) that directs and informs a given transformation in an organization [11]. Enterprise architecture can be used for: decision making regarding an intended business transformation; formulating business strategy impact; specifying (business) requirements; and informing and contracting service providers [13]. The main threats in enterprise architecture development include: choosing an ineffective leader as the lead enterprise architect; and not involving business (or organizational) stakeholders in the architecture program [3]. However, involving stakeholders in the architecture development process tends to result in several challenges. In [15] it is was reported that collaboration between stakeholders and enterprise architects is often challenging. However, there is lack of an elaborate description of the problematic nature of this collaboration, or of the challenges that arise if organizational stakeholders are to be involved in enterprise architecture development. Yet such a description is a prerequisite for developing a relevant solution to practitioners (in this case enterprise architects).

Therefore, there was need to investigate the challenges architects face when they involve organizational stakeholders in enterprise architecture development. Since developing enterprise architecture involves creating (designing/specifying); applying (implementing); and maintaining the architecture to support an organization's business goals [13], investigations of these challenges were limited to creating architecture. To investigate challenges that enterprise architects face when they involve stakeholders in creating an enterprise architecture, an exploratory survey was conducted. The survey specifically investigated: factors that hinder effective collaboration among organizational stakeholders and enterprise architects; challenges architects face when evaluating enterprise architecture design alternatives; methods architects use to manage collaboration with stakeholders; strengths and weaknesses of the methods used to support collaboration between stakeholders and architects; challenges architects face when delivering architecture products; and key determinants for successful enterprise architecture creation. A sample of 70 enterprise architects participated in this survey.

This paper discusses findings from this survey. The findings generally indicate the practical relevance of devising an artifact that will improve enterprise architecture creation by offering support for effective and efficient stakeholder involvement in the enterprise architecture creation process. The survey was conducted as part of an ongoing research (reported in [10,11,12]) that generally aims at developing a standard (but flexible) approach that can support effective and efficient collaborative problem solving and decision making during enterprise architecture creation. The survey findings serve as a motivation for this research. Section 2 discusses the rationale for undertaking an exploratory survey, section 3 explains the survey design, section 4 discusses the survey results, and section 5 gives the conclusion and ongoing work.

2 Rationale for an Exploratory Survey

According to [8,9], activity theory articulates that: artifacts (i.e. tools, symbols) mediate between the subject (i.e. the important actor(s) in an activity) and the object (i.e. the objective of an activity); there are rules that influence or govern the execution of an activity; the execution of an activity involves a community of actors; and the execution of an activity requires division of labour. The process of creating an enterprise architecture can be conceived as an activity (or collection of sub activities that contribute to completion of the main activity). Thus, table 1 shows how the notions in the activity theory have been interpreted and adapted in this research.

All aspects of the activity theory (see column 2 of table 1) are explicitly interpreted in the context of enterprise architecture creation (see column 3 of table 1), except item 6 (i.e. the division of labour). According to [19], during enterprise architecture development, an enterprise architect must identify all stakeholders' concerns, and then develop architecture views that reflect how all concerns will be addressed in the architecture and the intended tradeoffs. This implies (as shown in step 6 of table 1) that in enterprise architecture creation, there are: (1)

Table 1. Adaptation of Activity Theory in Enterprise Architecture Creation

Step #	Aspects to Identify (Based on Mwanza & Engestrom 2003)	Perspective in this Research
1	Activity of interest (i.e. type of activity of interest)	Creating an enterprise architecture for a given organization
2	Objective (i.e. why the activity is taking place)	According to Op't Land *et al.,* 2008, enterprise architecture can be created for any of the following purposes, i.e.: 1. Decision making regarding an intended business transformation; 2. Formulating business strategy impact; 3. Specifying (business) requirements; and 4. Informing and contracting service providers
3	Subjects (i.e. those involved in carrying out the activity)	Enterprise architects and organizational stakeholders
4	Tools (i.e. the means used by the subjects to perform the activity)	Enterprise architecture approaches, architecture modeling languages (for designing the enterprise architecture models), & other (simulation or visualization) tools that may be relevant depending on the situation in a given organization
5	Rules and regulations (i.e. cultural norms, rules, or regulations governing the performance of the activity)	1. The organization policies, principles, culture, strategic business drivers, business goals, and business requirements; 2. The external laws of business from regulatory bodies to which the organization is accountable; and 3. The guidelines defined by enterprise architecture approaches.
6	Division of labour (i.e. determining who is responsible for what when carrying out the activity, and organizing the roles)	In enterprise architecture creation, there are mainly 2 types of roles, i.e.: 1. Roles that are supposed to be accomplished by the enterprise architects, 2. Roles that are supposed to be accomplished by both enterprise architects and organizational stakeholders through effective collaboration.
7	Community (i.e. the environment in which the activity is carried out)	The environment comprises of enterprise architects and organizational stakeholders. According to TOGAF, stakeholders can be categorized into: corporate, end-user organization, project organization, system operations, and external key stakeholders.
8	Outcome (i.e. the desired outcome from carrying out the activity)	Feasible enterprise architecture products that address all stakeholders' concerns. According to Op't Land *et al.*, 2008, architecture products are: tangible & intangible products including: principles, models, views, intermediate results used to develop the enterprise architecture models, the evaluation of alternative solutions, shared understanding, shared agreement, & commitment amongst stakeholders.

some tasks must be accomplished by the enterprise architects; and (2) tasks that are supposed to be accomplished through effective collaboration between enterprise architects and organizational stakeholders. Moreover, several researchers and practitioners (e.g. [3,6,13,19,18,15,16]) have advocated for the need for collaboration between enterprise architects and organizational stakeholders during architecture creation. Yet in [15], it has been reported that it is often difficult for enterprise architects and stakeholders to effectively collaborate during enterprise architecture creation. This implies that during the division of labour, there is complexity on how roles in category 2 (see row 6 in table 1) can be fulfilled. For this complexity to be resolved, there is need for an in-depth understanding of challenges enterprise architects face when they collaborate with stakeholders to accomplish the roles in category 2. However, literature on enterprise architecture is silent on the details of the problematic nature of collaboration between stakeholders and enterprise architects during architecture creation. Therefore, there was need to undertake an exploratory survey so as to investigate the collaborative aspects in enterprise architecture creation. Findings from the survey would then give insight on how collaboration between stakeholders and architects can be improved during architecture creation (see section 4).

3 Design of the Exploratory Survey

This section presents the design of the exploratory survey that was conducted to find out problems that enterprise architects face when they involve organizational stakeholders in the enterprise architecture creation process. The target respondents in this survey were enterprise architects. Self administered questionnaires were used (see Fig. 1 in appendix), and were pretested using 10 enterprise architects. Prior to the survey, the suitable sample size and sampling method had to be determined. According to [7,17], the appropriate sample size to use in a survey depends on: (1) the acceptable sampling error, i.e. the error that occurs in survey results due to studying a sample instead of the whole population; (2) the size of the population and its heterogeneity with respect to the features of interest; (3) the desired level of accuracy (or confidence); (4) the research budget; and (5) the desired statistical value, i.e. population mean or population proportion – the percentage of individuals who fall into a given category. For this survey the desired level of accuracy was at least a 95% confidence interval, and acceptable sampling error was ± 10%. The statistical values required from the survey were percentages of the target population (i.e. enterprise architects) who experience or do not experience the aspects this research was investigating. Therefore, the following formula, as defined in [5,7,17], was used to calculate the required sample size in this survey:

$$s = \frac{z^2(p(1-p))}{e^2}$$

Here s represents the required sample size, z represents the number equivalent to the desired level of confidence, p represents the estimate of the proportion of people (i.e. enterprise architects) falling into the whole population of the target respondents, and e represents the acceptable sampling error. The self administered questionnaires were to be posted on international, national, and corporate mailing lists of enterprise architects. Thus, it was assumed that at least 90% of the population or subscribers to these mailing lists are enterprise architects. Hence the value of p is 90%. Moreover, since the desired level of confidence was 95%, then from the z statistical tables $z = 1.96$. Since the acceptable sampling error was ± 10%, then $e = 0.1$. Inserting these values in the equation above, $s = 35$. Therefore, the required sample size was at least 35 enterprise architects. In other words, assuming 90% of subscribers on mailing lists of enterprise architects are real architects, then for us to be at least 95% confident that the conclusions we draw from the survey results have a sampling error of ± 10%, at least 35 enterprise architects were required to participate in this survey.

The next step was to determine the appropriate method for selecting the required sample of architects. Sampling methods are divided into two categories i.e. probability sampling methods (which are used when the list of the whole population of study is available and it is possible to determine the likelihood of selecting any of the population units) and non probability sampling methods (which are used when the list of the population of study is not available and is difficult to obtain) [5,14,17]. In this survey the list of the target population (i.e. all enterprise architects) was not available and was difficult to obtain. Therefore, a non probability sampling method was used, i.e. purposive (or purposeful) sampling. Purposeful sampling is used when there is need to study and understand something about, or features of, a specific group of people [14]. The survey was conducted online (i.e. via http://www.thesistools.com/), where the target respondents received the questionnaires through the mailing lists of enterprise architects. A maximum of 70 enterprise architects participated in this online survey. This response doubled the earlier required sample size of 35 participants and consequently lowered the sampling error to ± 7%. This implies that we are 95% confident that the findings, or conclusions drawn, from the survey (see section 4) have a sampling error of ± 7%.

4 Survey Results and Discussion

In this section results from the exploratory survey are presented. These include: the problems architects face when collaborating and evaluating architecture design alternatives with stakeholders during architecture creation; challenges encountered when delivering enterprise architecture products; insights on how problems encountered during enterprise architecture creation can be overcome; and methods enterprise architects use to manage collaborative tasks during architecture creation. Aspects presented in sections 4.1 − 4.6 can be perceived as (research) requirements that must be fulfilled in order to improve enterprise architecture creation through effective and efficient stakeholder involvement.

4.1 Factors Hindering Effective Collaboration

Since collaboration (between stakeholders and enterprise architects) is a core thread in enterprise architecture creation [12], it was vital to investigate the hindrances of its effectiveness and efficiency during architecture creation. The following were reported as the factors affecting it. The percentage of enterprise architects that experience each hindrance is given in brackets.

1. Time constraints i.e. unavailability of key stakeholders because they have no time or priority to collaborate, and yet project time schedules are taut (77%).
2. Organization politics, hidden agendas of stakeholders (which cause them to block a long term vision due to their short term needs), prima donna behaviors (i.e. self-centeredness) of some stakeholders, and cases where people in the organization do not want clear decision making due to selfish reasons (56%).
3. Difficulty in truly understanding and communicating with stakeholders because architects mainly talk about abstract concepts and are unable to explain the true value of architecture in a language that the key decision makers understand, while stakeholders use words that do not have the same meaning for everyone (50%).
4. Lack of a well founded and shared vision on the business itself, its future development, its enterprise architecture, and the consequences of the architecture on the organization's sub levels. This is because some people find it difficult to imagine a new situation (47%).

5. Lack of architecture governance and a strong decision making process which leads to stakeholders not taking responsibility for their decisions (47%).
6. Limited awareness of (infrastructure) architecture or the need for architecture, stakeholders' perception about architecture (e.g. architecture is perceived to be about only technology), and the gap between (business) operations and enterprise architecture (44%).
7. Lack of long term planning e.g. long term effects may not be considered as part of the business case or project goal, members of the architecture project (i.e. business and IT staff) may be unknown, project managers may be assigned late when projects are already on critical path (42%).
8. Social complexity of an organization, conflicting agendas or interests of stakeholders, differences in stakeholders' perception about ambition levels, and the ladder of inference - i.e. stakeholders overreacting or quickly drawing conclusions based on personal beliefs, insecurities (40%).
9. Lack of documentation of knowledge in the organization (31%); the old fashioned distinction between business and IT (30%); and the "not invented here" syndrome of stakeholders (27%).
10. Lack of methods, tools, and techniques for supporting collaboration (17%).
11. Constrained project budgets (24%) and the "100% syndrome" of the architect (16%). Other factors include cases where stakeholders are unqualified for tasks assigned to them, or have an attitude of "the outsider is the expert but the outsider does not understand our situation" (3%).

Hindrances 3 – 9 and 11 above arise due to lack of a shared understanding (among stakeholders) of the aspects pertaining to the problem or challenge the organization is facing, and aspects pertaining to the (possible) solutions to address the problem. Where enterprise architecture development is the appropriate way of synergizing the formulation and implementation of the possible solutions. A shared understanding among stakeholders can be created through effective and efficient collaboration between stakeholders and architects, and as the level of shared understanding increases, stakeholders are motivated to effectively collaborate so as to achieve a successful architecture process [12]. Consequently, issues highlighted in hindrances 1, 3 – 9, and 11 will be (hopefully) overcome, since the value of architecture will have become explicit to the stakeholders. Therefore, the key requirement for addressing hindrances 1, 3 – 9, and 11 above, is to deploy into the architecture creation process, techniques (or approaches) that enhance a shared understanding of critical aspects among actors. Hindrance 2 however is beyond the scope of this research, and discussions on how to overcome organization politics are given in [18]. Hindrance 10 is what this research aims to address through devising an approach that will minimize occurrences of the issues reported above.

4.2 Methods Used in Practice to Support Collaborative Tasks

To address collaborative issues in enterprise architecture creation, there was need to first find out the methods currently used in practice, so as to identify their strengths and weaknesses. The following are the methods enterprise architects currently use when executing tasks that require involvement of organizational stakeholders during architecture creation. The percentage of enterprise architects that use a given method is given in brackets.

- Interviews (90%), traditional facilitated workshops (83%), desk research and modeling (53%)
- Rapid design workshops (24%), Accelerated Solutions Environment (ASE), and Innovate (13%). Accelerated Solutions Environment (ASE) is a generic approach used to create commitment, agreement, and approval by aligning a large group of critical stakeholders at the start of a business transformation strategy [1].
- Group Support Systems (13%), gaming (9%)
- Other methods (16%): These other methods include massive emailing, General Enterprise Architecturing (GEA), thematic work groups, peer reviews, elaborate-review sessions, crowd sourcing or co-creation methodologies. GEA is a tool that helps directors to underpin strategic decisions, connect several solutions within the organization, increase the cohesion between the organization's units and entities, and effectively achieve business objectives [20].

The use of interviews in architecture creation is discussed in [18], while the successful use of workshops during architecture creation remains ad hoc and in the hands of professional facilitators. Since workshops are widely used (as indicated above) to support collaboration between stakeholders and enterprise architects, there is need for a standard approach of effectively using them during architecture creation (see weaknesses of workshops in section 4.3). Standard in this context implies successful use of the approach without overdependence or reliance on the presence of professional facilitators, yet with minimal occurrences of the problematic issues reported in sections 4.1, 4.3, 4.4, and 4.5.

4.3 Strengths and Weaknesses of Methods Currently Used

The strengths and weaknesses of the methods currently used in practice gives insights into what needs to be done in order to improve collaboration between stakeholders and architects. For example, knowing weaknesses that need to be addressed in a particular method helps to improve the method through refining it or supplementing it with other methods. Architects revealed the following as the strengths and weaknesses of methods given in section 4.2.

Strengths and Weaknesses of Using Workshops. On the one hand, the strengths of using workshops are: (1) Workshops are suitable for ensuring engagement of stakeholders, enforcing group decision making, achieving common agreement on future states, and increasing acceptance and ownership of results. (2) They yield multiple stakeholder views, and make it possible to identify potential conflicts and then develop a common (or shared and supported) view. (3) They enable stakeholders to undergo the collaboration experience, where there is intensive communication and mutual understanding, and this builds stakeholders' commitment and support. Architects also reported that if workshops are prepared and conducted properly, they are the efficient way of managing collaborative tasks during enterprise architecture creation.

On the other hand, the following are the weaknesses of using workshops: (1) Results of workshops are of an informal character. (2) The quality of output from a workshop depends on the skills of the facilitator(s) and also on the skills and knowledge of workshop participants (stakeholders). (3) Information from workshops is not sufficiently detailed. (4) Workshops lack anonymity. (5) It is time consuming to prepare and conduct workshops, and to process their results. (6) In workshops it is difficult to stay focused on the agenda. (7) When not all key stakeholders are available, workshops slow down decision making, and this negatively affects the momentum of the architecture development process. (8) Workshops are often not very structured and allow interpretation freedom to a large degree. Architects also reported that if GSSs are used within a workshop, they help in sharing and storing content during the workshop, although using GSS requires a lot of preparation time.

Strengths and Weaknesses of Using Only Interviews. On the one hand, the strengths of using interviews are: (1) Interviews are necessary if awareness is low, since they provide detailed information in a little time, and help the architect to get a good understanding of the interviewee's situation. (2) Interviews are more useful for investigating individual needs and obstacles since they are private, focused, and flexible to get ideas. They enable the architect to ask very specific questions and stakeholders to give true and less socially wanted answers. They prompt introvert persons to disclose their opinions. (3) They are easy to prepare and schedule, and are suitable for stakeholders who have limited time for collaborative activities or sessions. (4) Interviews have, by nature, an active party (the party answering questions) and a passive party (the party asking the questions) – hence they do not involve non participating parties or stakeholders. (5) They help to manage stakeholders' expectations, and to get buy-in and commitment from stakeholders.

Weaknesses of using only interviews include the following: (1) Conducting interviews with all key stakeholders and processing results from the interviews is time consuming. Hence a limited number of people can be reached. It is also often difficult to get the right person or time or right mindset of a stakeholder. (2) Interviews give single stakeholder view(s), hence several different opinions or views and there is lack of agreement (on matters) between stakeholders. It is therefore very difficult to create a shared, well documented, understandable, prioritizable, referenceable, discussable, and evolutionary architecture. (3) The lack of interaction between stakeholders leads to insufficient understanding of each others concerns and issues. (4) Interviews offer limited opportunities for creativity among stakeholders.

Strengths and Weaknesses of Desk Research and Modeling. Desk research is required in all cases to get a deeper understanding of the issues and objectives involved in a given organization situation. Weaknesses of desk research and modeling are: (1) division of work between architects is often difficult; and (2) preparation and processing of results from desk research is time consuming.

Strengths and Weaknesses of ASE and Rapid Design Workshops. The speed at which things are done when using ASE and rapid design workshops is good, and they enable thorough discussion with all stakeholders. Weaknesses of ASE and rapid design workshops are: (1) ASE is sometimes too fixed on achieving a specific task; and (2) the depth of problem solving and detailing of an ASE (as well as a rapid design workshop) is also limited.

4.4 Evaluation of Enterprise Architecture Design Alternatives

In enterprise architecture creation there is need to evaluate enterprise architecture design alternatives i.e. alternative ways in which stakeholders' concerns can be addressed in the architecture [10]. Survey findings show that 96% of enterprise architects involve organizational stakeholders when evaluating architecture design alternatives, and the problems these architects face are given below. The percentage of architects who face each problem is given in brackets.

1. Lack of a truly shared vision and strategy by all stakeholders (53%).
2. Organization politics (40%).
3. Lack of shared agreement, i.e. it is hard to reach a compromise or to get everyone to agree with the same result due to conflicting agendas (36%). Biased scores due to personal preferences, agendas, and visions; or not invented here syndrome (34%).
4. Lack of a clear decision making unit in the organization (36%). It is difficult to make a very good presentation that leads to decision making; and is very clear, only containing the essentials and alternatives, and prevents discussions of too much detail (39%).
5. Stakeholders have limited knowledge of the content, the goals of the architecture, or how to read an architecture document or view (32%).
6. Bridging the gap between the abstract long term consequences and the more concrete examples that stakeholders can understand (31%).
7. Time or budget constraints rarely allow sufficient interactions with stakeholders, so as to break the complexity in evaluating alternatives (24%).
8. It is hard to quantify advantages and disadvantages of alternatives (23%).

Issues 1 and 3 – 7 above can be addressed through creating a shared understanding (among stakeholders) of the problem and solution aspects of the organization (as discussed in section 4.1).

4.5 Acceptance–Related Challenges Faced by Architects

Successful architecture creation is perceived as designing an architecture, gaining stakeholders' acceptance of the architecture, and being able to implement it with their support and commitment [13]. It is reported in [12] that although 96% of enterprise architects closely collaborate with organizational stakeholders during enterprise architecture creation, 90% of them face challenges when delivering products of enterprise architecture creation. From the survey, architects reported the following as the challenges they face when delivering the architectural products. In brackets, the percentage of architects who face each challenge is indicated.

1. Some organizations lack a clear decision making unit, leading to a loud applause but no action (44%). In other cases architecture may be too complex for the decision making unit or organization maturity level (29%).
2. Since architecture is often perceived to be about only technology, some organizations lack a governance process for ensuring architecture compliancy (44%).
3. Architecture conclusions may sometimes conflict with personal ambitions or agendas (37%).
4. The client organization may change its business plans (37%).
5. Using the right language such that every stakeholder understands the architecture (34%), and making a short and clear description of the architecture to all stakeholders within a short time (13%).
6. Lack of commitment from people who were not earlier involved in the architecture process (24%). In other cases concerns arise from other stakeholders who were not seen as stakeholders before (21%).
7. Difficulty in translating enterprise architecture products to program start architectures (17%).
8. Architecture products do not often deliver what has been promised or what was required (11%).
9. Other issues include cases where stakeholders do not want to (or are not able to) follow the advised architecture, or where the created architecture shows that the impact of the business strategy is higher than anticipated (3%).

Challenge 4 can be addressed using guidelines in the architecture requirements management phase of TOGAF ADM [19]. In our view, other challenges listed above are byproducts of the quality of, and the preparations for, stakeholders' involvement (i.e. collaboration between enterprise architects and organizational stakeholders) during architecture creation. For example, delivery of models that are too complex often indicates that the architecture function is not properly integrated in the organization and this is due to, among other factors, the problematic nature of collaboration between architects and stakeholders [15].

4.6 Success Factors For Enterprise Architecture Creation

Since successful enterprise architecture creation is a key motivation for this research, it was vital to find out practitioners' views on its determinants. At least

72% of architects recommend that it is vital to first get the business goals clear i.e. to know the reasons for creating the architecture or which organization problems should be solved by creating the enterprise architecture. In addition, 51% of architects agree that there should be an effective (i.e. understood by, and visible to, all stakeholders) translation of these business goals and concerns into the actual architecture because enterprise architecture is purely a means by which an organization can achieve its goals. Moreover, according to [4,13], translation of strategy into architecture or desired business operations is perceived as architecture creation.

According to 71% of architects, it is vital to select the right stakeholders and get involved with them early in the process. Moreover, 66% of the architects agree that architects need to have good collaboration with these stakeholders (e.g. owners or subject matter experts) through regular communication in order to keep everyone on track and create a strong sense of cooperation and shared objectives. At least 24% of architects further advise that it is important to create a situation that enables all stakeholders to experience the development process through, e.g., scheduling short group sessions that fit in the schedules of key stakeholders early in the architecture process. At least 48% of architects agree that it is vital for an organization to have a clear and strong decision making unit or architecture board which can make decisions and give a clear mandate for architects to make decisions within agreed boundaries. This is because architecture concerns assessment of governorship, guidance, and growth. Lastly, 34% of architects recommend that architects, project manager(s), and business executive(s) need to respect each others' roles during the architecture process.

5 Conclusions and Ongoing Work

Findings from the exploratory survey on enterprise architects generally indicate that although involving organizational stakeholders in enterprise architecture creation is vital, it results in several issues. These issues indicate the problematic nature of involving stakeholders in enterprise architecture creation. They also justify the need for supplementing existing enterprise architecture approaches with support for collaborative problem solving techniques, so as to create a shared understanding of the organizational problem and solution aspects among stakeholders. Findings from the survey give insight on how collaboration between stakeholders and architects can be improved during architecture creation. Therefore, survey findings are currently being used as a basis for developing a theory and a method that will (hopefully to a large extent) address the challenges in collaborative architecture creation.

References

1. Holland ASE Team. The power of group dialog: Accelerated Solutions Environment. Utrecht, The Netherlands. (2005)

2. Dietz, J.: *Architecture Building Strategy into Design* (Netherlands Architecture Forum, Academic Service SDU, The Hague, The Netherlands, 2008). ISBN-13: 9789012580861, http://www.naf.nl.
3. Gartner, Inc.: Gartner Identifies Ten Enterprise Architecture Pitfalls. Gartner Enterprise Architecture Summit 2009 `http://www.gartner.com/it/page.jsp?id=1159617`. Accessed on August 15th 2010.
4. Jonkers, H., Lankhorst, M. M., Doest, H.W. ter, Arbab, F., Bosma, H., Wieringa, R. J.: Enterprise architecture: Management tool and blueprint for the organisation. Information Systems Frontiers 8(2) 63–66, (2006)
5. Kish, L.: Survey Sampling. John Wiley and Sons Inc. New York (1965).
6. Lankhorst, M. et al.: Enterprise Architecture at Work: Modelling, Communication, and Analysis. Springer Verlag Berlin, Heidelberg (2005)
7. Lomax, R. G.: An Introduction to Statistical Concepts for Education and Behavioral Sciences. Lawrence erlbaum associates. Mahwah, New Jersey. (2001).
8. Mwanza, D., Engestrom, Y.: Managing content in e-learning environments. Britisch Journal of Educational Technology, 36(3), 453 – 463.
9. Nardi, B.: Context and Consciousness: Activity theory and Human Computer Interaction. MIT Press Cambridge, Massachusetts (1996).
10. Nakakawa, A., Bommel, van P., Proper, H. A.: Quality Enhancement in Creating Enterprise Architecture: Relevance of Academic Models in Practice. In E. Proper, F. Harmsen, and J.L.G. Dietz (Eds.): PRET 2009, LNBIP 28, 109 - 133, 2009.
11. Nakakawa, A., Bommel, van P., Proper, H. A.: Requirements for Collaborative Decision Making in Enterprise Architecture. Proceedings of the 4th SIKS/BENAIS Conference on Enterprise Information Systems. Nijmegen: The Netherlands. (2009)
12. Nakakawa, A., Bommel, van P., Proper, H. A.: Towards a Theory on Collaborative Decision Making in Enterprise Architecture. Global Perspectives on Design Science Research. LNCS 6105, 538 – 541, Springer (2010).
13. Op ‘t Land, M., Proper, H.A., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture - Creating Value by Informed Governance. Berlin, Springer. (2008).
14. Patton, M. Q.: Qualitative evaluation and research methods. SAGE Publications. Beverly Hills. (1980)
15. Raadt, B. van der, Schouten, S., Vliet, H. van: Stakeholder Perspective of Enterprise Architecture. In: Morrison, R., Balasubramaniam, D., and Falkner, K. (eds.) ECSA 2008. LNCS, vol. 5292, 19–34. Springer, Heidelberg (2008)
16. T. W. Rehkopf, N. Wybolt, Top 10 Architecture Land Mines. IT Professional 5 (6) pp. 36 - 43 (2003). doi:10.1109/MITP.2003.1254967
17. Sallant, P., Dillman d. A.: How to Conduct your Own Survey. John Wiley & Sons, Inc. New York. ISBN 0471012734 (1994).
18. Spewak, S. H.: Enterprise Architecture Planning: Developing a Blue Print for Data, Applications, and Technology. New York, John Wiley & Sons Inc (1992)
19. The Open Group Architecture Forum. (2009). TOGAF Version 9. Zaltbommel, The Netherlands: Van Haren Publishing. ISBN: 978-90-8753-230-7
20. Wagter, R.: Focus on consistency, excel with GEA. `http://www.ordina.nl/Visie%20en%20Ervaring/Themas/Organisatiebesturing/GEA.aspx`. Accessed on August 20th 2010

Appendix

Radboud University Nijmegen, The Netherlands
An Exploratory Survey on Collaborative Aspects in Enterprise Architecture Creation

Introduction: The aim of this survey is to investigate aspects concerning collaboration between stakeholders and enterprise architects during enterprise architecture development. Results from this survey will be used to determine the practical relevance of developing an approach that enterprise architects can use to effectively and efficiently execute enterprise architecting guidelines that are "*collaboration dependent"*. Collaboration dependent architecting guidelines are architecture creation tasks, whose successful completion requires enterprise architects to successfully collaborate with organizational stakeholders.

1. Which architecture method(s) are you currently using? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
2. Do you consider the architecture development process to be collaborative in nature?
   1. YES 2. NO
3. If YES to question (2) above, which method do you use to manage collaborative tasks during architecture creation? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
4. Please give a strength and/or weakness of the method(s) you use to manage collaboration during architecture creation.
   ………
   ………………………………………………………………......…………………………………………………………………….
5. Which factors hinder effective collaboration between architects and key stakeholders during architecture creation? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
6. Do you also engage organisation stakeholders during the evaluation of architecture design alternatives?
   1. YES 2. NO
7. If YES to question (6) above, which type of organisation stakeholders do you engage in the evaluation of architecture design alternatives? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
8. Which method do you use to evaluate architecture design alternatives with stakeholders? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
9. Which challenges do you face during the evaluation of architecture design alternatives? (please mark all options that apply, and specify on option "other")
   [due to space limitations the bullet options have been removed]
10. Do you face any challenges related to acceptance of the products you deliver after architecture creation?
    1. YES 2. NO
11. If YES to question (10) above, which of the following are examples of such challenges? (please mark all options that apply, and specify on option "other")
    [due to space limitations the bullet options have been removed]
12. From your experience, which of the following do you consider as success factors for architecture creation? (please mark all options that apply, and specify on option "other")
    [due to space limitations the bullet options have been removed]
13. We are developing a method to manage collaborative tasks in enterprise architecture. We will be conducting another questionnaire survey with the aim of validating the design of the method. Would you be interested in participating in that survey?
    a. NO b. YES (please give your contact) ………………………………….
14. We will also carry out an experiment on the designed method, would you be interested in participating in the validation experiment of such a method?
    a. NO b. YES (please give your contact) …………………………………

Fig. 1. Exploratory Questionnaire Survey

Collaborative Modeling: Towards a Meta-model for Analysis and Evaluation *

D. (Denis) Ssebuggwawo[1], S.J.B.A. (Stijn) Hoppenbrouwers[1], and H.A. (Erik) Proper[1,2]

[1] Institute of Computing and Information Sciences, Radboud University Nijmegen
Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands, EU.
D.Ssebuggwawo@science.ru.nl, stijnh@cs.ru.nl
[2] Public Research Centre – Henri Tudor, Luxembourg, EU.
erik.proper@tudor.lu

Abstract. In this paper we discuss a meta-model for the analysis and evaluation of collaborative modeling sessions. In the first part of the meta-model, we use an analysis framework which reveals a triad of rules, interactions and models. This framework, which is central in driving the modeling process, helps us look inside the modeling process with the aim of understanding it better. The second part of the meta-model is based on an evaluation framework using a multi-criteria decision analysis (MCDA) method. Central to this framework, is how modelers' quality priorities and preferences can, through a group decision-making and negotiation process, be traced back to the interactions and rules in the analysis framework.

Key words: Collaborative Modeling, Modeling Process Quality, Modeling Process Analysis, Modeling Process Evaluation, Group Support Tools

1 Introduction

A number of studies have, over the years, looked at collaborative modeling [1,2,3]. There have also been attempts to understand the modeling process [4,5]. Such modeling is driven by participants' communication. Human communication [6], in collaborative modeling, involves argumentation, negotiation and decision making. Often, participants need to agree, through negotiation and decision making, on what constitutes, for example, "quality" for the different modeling artifacts and how such quality should be assessed. However, how to assess the quality of the collaborative modeling process, especially with respect to the modeling artifacts, remains a largely unexplored area.

The current paper tries to develop a meta-model which can be used for both the analysis and evaluation of a collaborative modeling process and the relation

---

* This paper first appeared as a Working Paper on Information Systems in Sprouts. http://sprouts.aisnet.org/10-36/

between events in the process and the resulting artifacts. The meta-model links the modeling artifacts and the evaluation framework to the rules, interactions and models (RIM) framework [7] through the interactions which are governed by rules. The interactions, rules and models are a result of the communicative process, mainly through modelers'negotiation. Negotiation plays a key role in collaborative modeling. It is through negotiation that modelers reach agreement and possibly consensus. In this paper we limit our discussion to negotiation dialogues from argumentation theory.

Negotiation dialogue has been widely studied, see for example [8,9,10]. Its practical applications include multi-agent systems (MAS) [11,12,13,14] with wide applications in electronic commerce [15,16,17]. Negotiation dialogues start from a position of conflict and the goal is to establish some consensus or compromise for all the parties involved. Usually, participants have conflicting objectives, interests, preference and priorities. Through the process of negotiation, they get a compromise position that everyone is comfortable with. This is what happens in a multi-actor (collaborative and interactive) modeling process. Modelers have conflicting views, priorities and preferences and they engage in an argumentation process, that involves, propositions, (dis)agreements, acceptances and rejections, supports and withdraws, etc, to reach a compromise.

It should be noted that, although there are a number of factors that one may be interested in looking at in the analysis and evaluation of the modeling process, which in fact may influence the quality of the modeling process, e.g., power struggle, leadership and the unspoken message or body language, etc., (see for example, [18]), our interest at the moment is in what we call "drivers" of the modeling process. Rules and/or goals, interactions, and models are hypothesized to be drivers of the modeling process. In this paper we concentrate on only these.

2 Modeling Process Analysis: The RIM Framework

Stakeholders, in a collaborative modeling process, interact and communicate their ideas and opinions to other members through the communication process. Three key items concerning this communication are the rules, the interactions and the models. The rules, interactions and models (RIM) framework is based on these items and helps us look into the collaborative modeling process. This framework is depicted in Fig. 1. Details of the RIM framework are found in [7]. The RIM framework is a three-tier framework that examines the communicative acts (interactions) in a modeling session, the rules/goals set, and the models produced as a result of the interaction and collaboration. The different collaborative modeling *players* work under a set of *rules and goals*. The rules/goals, interactions and models are all time-stamped to help us track and identify he interplay between any pair. The interplay of rules, interactions and models is explained in Table 1.

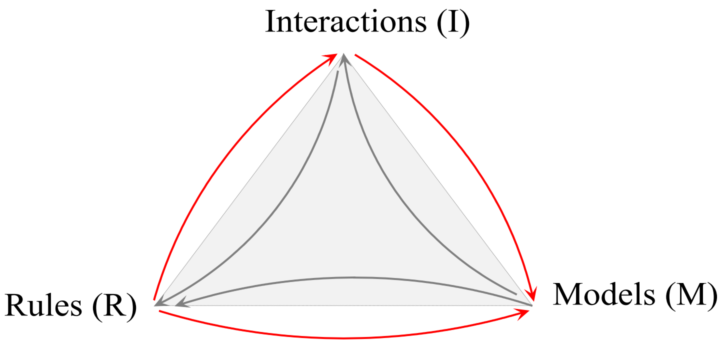


Fig. 1. A framework for analyzing interactions, rules and models.

Table 1. RIM framework features

Path	Interplay
IM-MI	The interactions lead to the generation of models and generated (intermediate) models drive further interaction.
RM-MR	Some rules/goals of modeling apply to (intermediate) models and these models may lead to the setting of new rules/goals.
RI-IR	Rules guide and restrict interactions and some interactions may change the rules of play.

2.1 Interaction Analysis: The Structure

In order to analyze the interactive conversations and determine the structure of the speech-acts that result thereof, we need to apply a discourse analysis or conversation analysis technique. There are a number of methods which can be used, notably, speech-act theory by Searle [19]. Searle's aim in his "Theory of Speech Acts" [19] was to show that: "speaking a language is performing acts $(\cdots)$ in accordance with certain rules for the use of the linguistic elements", and to formulate these rules. He argues that the minimal unit of an utterance is not a word or sentence but a "speech act". Two types of speech acts were identified in his theory: *propositional act* - which is the act of uttering words and *illocutionary act* - which is a complete speech act. An illocutionary act has two components: *propositional content* which describes what an utterance is about and *illocutionary force* describes the way it (utterance) is uttered. In addition, each illocutionary act has an *illocutionary point* which characterizes that particular type of speech act. Searle classifies utterances according to the illocutionary point and proposes five classes of speech acts shown in Table 2.

However, as argued in [20], speech-acts are individual statements in the whole conversation and cannot be analyzed outside the whole conversation in which they occur. The language-action perspective (LAP) [21] is, therefore, a candidate in analysing the whole conversation in which the speech-acts are just components. We base our analysis of the communicative process on LAP to identify the conversational interactions that occur in a collaborative modeling process.

Table 2. Illocutionary speech act types .

Speech-Act Type	Explanation
Assertive	represent facts of the world of utterance or common experiences, e.g., reports or statements
Directives	represent the speaker's attempt to get the hearer perform the action indicated in the propositional content, e.g., requests
Commissives	represent the speaker's intention to perform the action indicated in the propositional content, e.g., promises
Expressives	say something about the speaker's feeling or psychological attitudes regarding the state of affairs represented by the propositional content, e.g., apologies
Declaratives	change the world through the utterance of a speech act

Fig. 2 shows the structure of the interactions. We use *Object Role Modeling* (ORM) method [22] to represent analysis and evaluation concepts in this paper. Table 3 shows the elements of the interaction component.

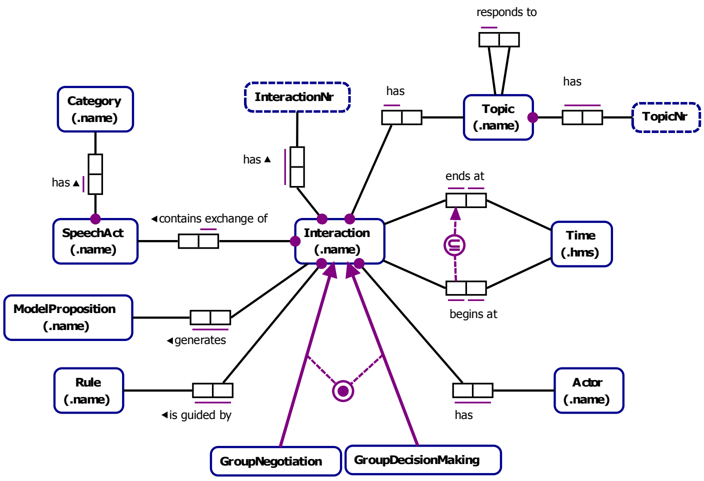


Fig. 2. Elements of an interaction

2.2 Rule Analysis: The Structure

Rules govern the interactions and production of the models. They guide collaborative modelers during the modeling process and can be set for (before) or in (during) the modeling process. They link the product of the conversations - the model to the conversations and they are intended to guarantee both process

Table 3. Explanation for elements of an interaction

Element	Explanation
InteractionNr	Unique number that refers to an interaction.
Time	Time at which an interaction is (de-)activated.
Topic	Subject under discussion in an interaction with a topic number.
Actor	A participant in an interaction.
Speech-act	An illocutionary act from the interaction and has a category.
ModelProposition	Model formation proposition (implicitly/explicitly agreed to).
Rule	Guideline(s) or convention(s) that direct the interactions.

quality and model quality. Rules are either explicitly stated or implicitly stated. The elements of a rule are given in Fig. 3 while Table 4 explains these elements.

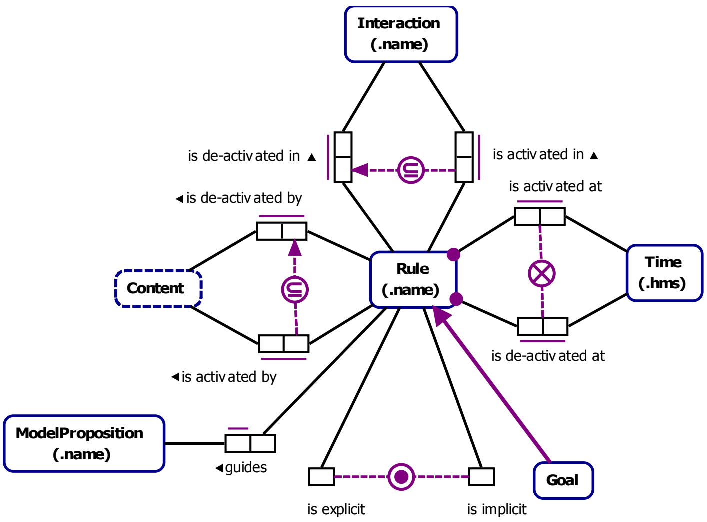


Fig. 3. Elements of a rule

Table 4. Explanation for elements of a rule

Element	Explanation
Content	Conversational content in which a rule is (de-)activated.
Time	Time at which a rule is (de-)activated.
Interaction	Conversations from which propositions are generated.
ModelProposition	Model formation proposition (implicitly/explicitly agreed to).
Goal	A rule that sets the state to strive for.

2.3 Model Analysis: The structure

Models (intermediate or final) are lists of propositions up to time t, i.e. conversational statements commonly agreed upon and shared by all the modelers. These model propositions are subject to selection criteria in order to determine which one makes it to the group (shared) model. In collaborative modeling a model proposition is either explicitly agreed with or implicitly not disagreed with. The structure of a model proposition component is shown in Fig. 4 while its elements are explained in Table 5.

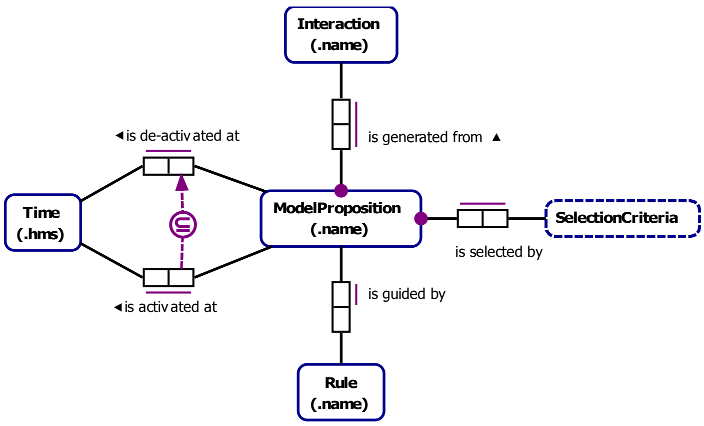


Fig. 4. Elements of a model proposition

Table 5. Explanation for elements of a model proposition

Element	Explanation
Rule	Guidelines that direct the selection of a model-proposition.
Time	Time at which a model-proposition is (de-)activated.
SelectionCriteria	A set of evaluation criteria used to select a model-proposition.
Interaction	Interaction from which a model-proposition is generated.

3 Modeling Process Evaluation: An MCDA Framework

In collaborative modeling a number of artifacts are used in, and produced during, the modeling process. These include the modeling language, the methods or approaches used to solve the problem, the intermediate and end-products produced and the medium or support tool that may be used to aid the collaboration, see for example [23]. The priorities of the individual decision makers need to be aggregated, so as to reach agreement and consensus on what should be the group's position as far as modeling process quality is concerned. Reaching agreement requires group decision making and negotiation. Group decision making and negotiation are special types of interactions during the modeling process. This is what provides a link between the analysis (RIM) framework and the evaluation (MCDA) approach. In Section 4, it will be shown how this link is exploited to get a unified framework for analysis and evaluation. In the evaluation, we use a Multi-criteria Decision Analysis (MCDA) method to evaluate the modeling artifacts. We specifically use the single synthesizing (weighting) criterion preference approach - with Analytic Hierarchy Process (AHP) [24].

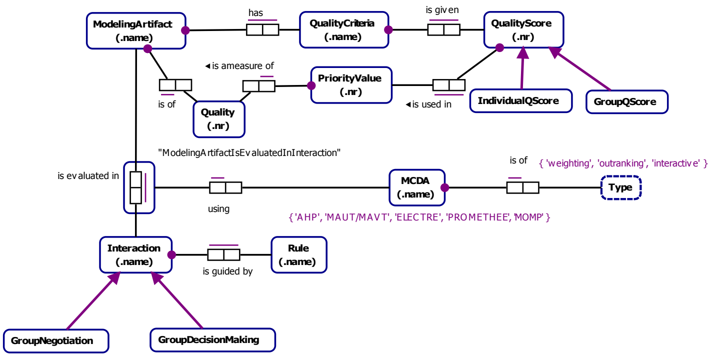


Fig. 5. Elements of a modeling artifact

Table 6. Explanation for elements of a modeling artifact

Element	**Explanation**
Quality	Degree of excellence or deficiency-free state.
QualityCriteria	A modeling artifact feature to measure quality.
QualityScore	A value given to a criterion as a measure of its quality. It may be an individual or group score.
PriorityValue	Aggregated quality scores to determine priority values.
Interaction	Group negotiation/decision-making to agree on quality scores.
Rule	A set of guidelines that direct the interactions.
MCDA	A multi-criteria decision analysis approach used for the evaluation. It is of a certain type

The structure of the evaluated modeling artifact component, within the MCDA evaluation framework, is shown in Fig. 5. The different concepts are explained in Table 6. One important observation about the modeling artifact and the evaluation framework is the link provided by the evaluated modeling artifact to the RIM framework through the interactions which are governed by rules. This is an important observation since it helps us to unify the two frameworks.

4 The Analysis and Evaluation Meta-model

In this section we combine the components to form a unified model for the integrated analysis and evaluation (of process and results) of collaborative modeling. The aim of having a unified framework is twofold: 1) to trace the flaws in the modeling process using the evaluation framework back to the analysis framework, 2) to automate the analysis and evaluation by a having support tool which can be used to both analyze and evaluate the modeling process. Although the analysis and evaluation frameworks can stand on their own, having a tool-support that can help modelers to analyze and evaluate the process and trace flaws in the entire modeling process is more attractive than the individual frameworks. The components of the integrated frameworks are linked together in a meta-model shown in Fig. 6. The novelty of the meta-model is that it combines the analysis and evaluation frameworks, i.e., the RIM framework and the MCDA framework. This is easily visible in the meta-model where the triage of the rules (R), interactions (I) and models (M) in Fig. 1 is depicted through the rules, interactions and model proposition entities.

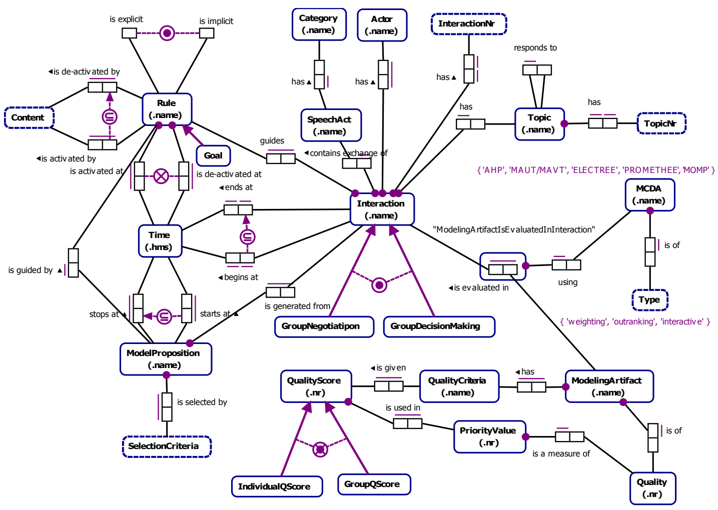


Fig. 6. An integrated meta-model for collaborative modeling analysis and evaluation

5 Meta-Model in Use: Illustrative Examples

To demonstrate the theoretical importance and practical significance of the model we provide below some illustrative examples. The examples are drawn from recorded communication/conversations that took place during a modeling session.

5.1 Application of the Meta-Model: The Analysis

Example 1. Interaction analysis in Fig. 2 is based on the following excerpt. Table 7 shows the elements of an interaction.

Time	**Actor**	**Speech Act**
02:00	M1	*So, where does Ordering start?*
02:03	M2	*First we have to decide who takes part in it. So we can set that on top of the diagram?*
02:10	M1	*There are numbers, so that's easy, so probably the purchasing officer is involved?*
02:18	M2	*Eh ... I guess so.*
02:21	M1	*So he needs ordering one second ... "draws 2".*

Table 7. Extracted elements of interaction from the coded meta-data

Int. #	Int. Name	Top. #	Top. Name	Speech Act Type/Category	Rsp. to	Time	Actor
1	INFORMATION SEEKING	1	SET CONTENT	QUESTION [Where does ordering start?]		02:00	M1
2	DECISION MAKING	2a	SET CONTENT	PROPOSITION [First we have to decide who takes part in Ordering]		02:03	M2
		2b	SET GRAMMAR GOAL	QUESTION [Can we set who takes part in Ordering on top of the diagram?]			
3	INQUIRY	3a	SET GRAMMAR GOAL	PROPOSITION-QUESTION [There are numbers, so that's easy, so probably the purchasing officer is involved?]	2b	02:10	M1
		3b	SET CONTENT	PROPOSITION [Purchasing Officer is involved in Ordering]	2a		
4	NEGOTIATION	4	SET CONTENT	AGEEMENT WITH [Eh… I guess so]	3b	02:18	M2
5	DELIBERATION	5	SET CONTENT	DRAWING [So he needs ordering … one second … "draws **2",i.e.,** number 2 (purchasing officer) on top of first swim lane		02:21	M1

KEY: Int.: Interaction Top.: Topic Rsp.: Response.

Example 2. Rule analysis for Fig. 3 is based on the following excerpt of modeling session conversations. Extracted elements of a rule from the coded meta-data are given in Table 8.

Time	**Actor**	**Speech Act**
01:25	M1	*Let's create 5 swim lane diagrams.*
01:30	M2	*Yes, isn't that what I just proposed?*
08:43	M1	*Sequences are started with the START symbol ...*
08:45	M2	*Yes ...*
08:48	M2	*Use blocks to indicate activities.*
15:18	M1	*So no decision diamonds in UML activity diagrams?*
15:19	M2	*No; well; maybe.*

Table 8. Extracted elements of a rule from the coded meta-data

Rule	Int. Name$_{[A]}$	Content$_{[A]}$	Time$_{[A]}$	Int. Name$_{[D]}$	Content$_{[D]}$	Time$_{[D]}$	M.P
VALIDATION GOAL	DELIBERATION	All participants should agree on the model. [Proposed and activated in the Assignment.]	All t	DELIBERATION	De-activated when all or the majority have agreed on the model, i.e. reached consensus.	End t	
CREATION GOAL	PERSUASION	Let's create 5 swim lane diagrams - [14] **PROPOSITION**	01:25	PERSUASION	Yes, isn't that what I just proposed?-[15] **ARGUMENT FOR** 14	01:30	A.C [14]
GRAMMAR RULE	INFORMATION SEEKING	Sequences are started with the START symbol …- [148] **CLARIFICATION**	08:43	INFORMATION SEEKING	Yes…[149] **AGREEMENT WITH** 148	08:45	A.C [148]
GRAMMAR GOAL	NEGOTIATION	Use blocks to indicate activities - [151] **PROPOSITION**	08:48	-	-	-	A.C [151]
GRAMMAR GOAL	INQUIRY	So no decision diamonds in UML activity diagrams?[248] **QUESTION**	15:18	INQUIRY	No; well; maybe-[249] **ANSWER** 248	15:19	

KEY: Int.: Interaction A.C.: Activation Content M.P.: Model Proposition [A/D]: Activated/De-activated

Some explanation is in order for some of the concepts shown in Tables 7 and 8. The categories for coding the modeling conversations, i.e., the interaction names in both tables correspond to the dialogue types of Walton and Krable [25] whereas the topic names and rule categories, in Table 8, are explained in [7]. The validation goal is an example of an explicitly stated rule. This is activated at the start of the modeling session and remains so until de-activated at the end of the modeling session. The others are all implicitly stated and are (de-)activated during the interactions as shown by the (de-)activation content. It should be be noted that we use the terms **"activation"** and **"de-activation"** in the sense that modeler M1 starts the argument and modeler M2 concludes it in the sense of reaching a final agreement. For each we identify, respectively, the interaction, content and time in (by, at) which the argument was started and concluded.

Example 3. Model proposition analysis in Fig. 4 is based on the following excerpt. Extracted elements of a model proposition from the coded meta-data are given in Table 9.

Time	**Actor**	**Speech Act**
14:41	M1	*If there is no place, he can't order or there is no availability.*
14:45	M2	*Yeah, true...*
14:50	M2	*You cannot do decision diamonds in UML activity diagrams.*
14:57	M2	*You can only have splits and joins of some sort, not the decisions as such.*
16:46	M1	*We can also say that if the form isn't filled in well then it is rejected but...*
16:55	M2	*Yeah ...*
17:07	M1	*No-route and terminal point from "accept" in swim lane 7, with "no order" ...*
17:14	M2	*OK..., Yes*

Table 9. Extracted elements of a model proposition from the coded meta-data

Model Proposition	Time		Rule Name	Int. Name	Selection Criterion
	Act.	De-act.			
If there is no place, he cannot order or there is no availability. Yeah, true...	14:41	 14:45	CREATION	NEGOTIATION	Explicitly agreed with
You cannot do decision diamonds in UML activity diagrams. You can only have splits and joins of some sort, not the decisions as such.	14:50 14:57	- -	GRAMMAR	PERSUASION	Not explicitly disagreed with.
We can also say that if the form isn't filled in well then it is rejected but... Yeah ...	16:46	 16:55	CREATION	NEGOTIATION	Explicitly agreed with.
No-route and terminal point from "accept" in swim lane 7, with "no order" ... OK..., Yes	17:07	 17:14	GRAMMAR	NEGOTIATION	Explicitly agreed with.

KEY: Act.: Activated De-act.: De-activated Int.: Interaction

5.2 Application of the Meta-Model: The Evaluation

Example 4. Evaluation analysis in Fig. 5 is based on an evaluation instrument part of which is shown in Fig. 7. This instrument is used, first by individual modelers, and then second by a team of modelers, to evaluate the modeling artifact (modeling language, modeling procedure, modeling products-the models and the support tool). The instrument shows, for example, how a modeling procedure is evaluated (using its selected quality criteria). These are assigned scores using the fundamental scale [24]. The quality criteria (quality dimensions of the modeling artifacts) are defined in [23] and the process of assigning these quality criteria scores is explained therein. Upon reaching consensus through negotiation and decision making processes, modelers use these scores in the computation of priorities and the overall quality for the modeling artifacts as shown in Table. 10.

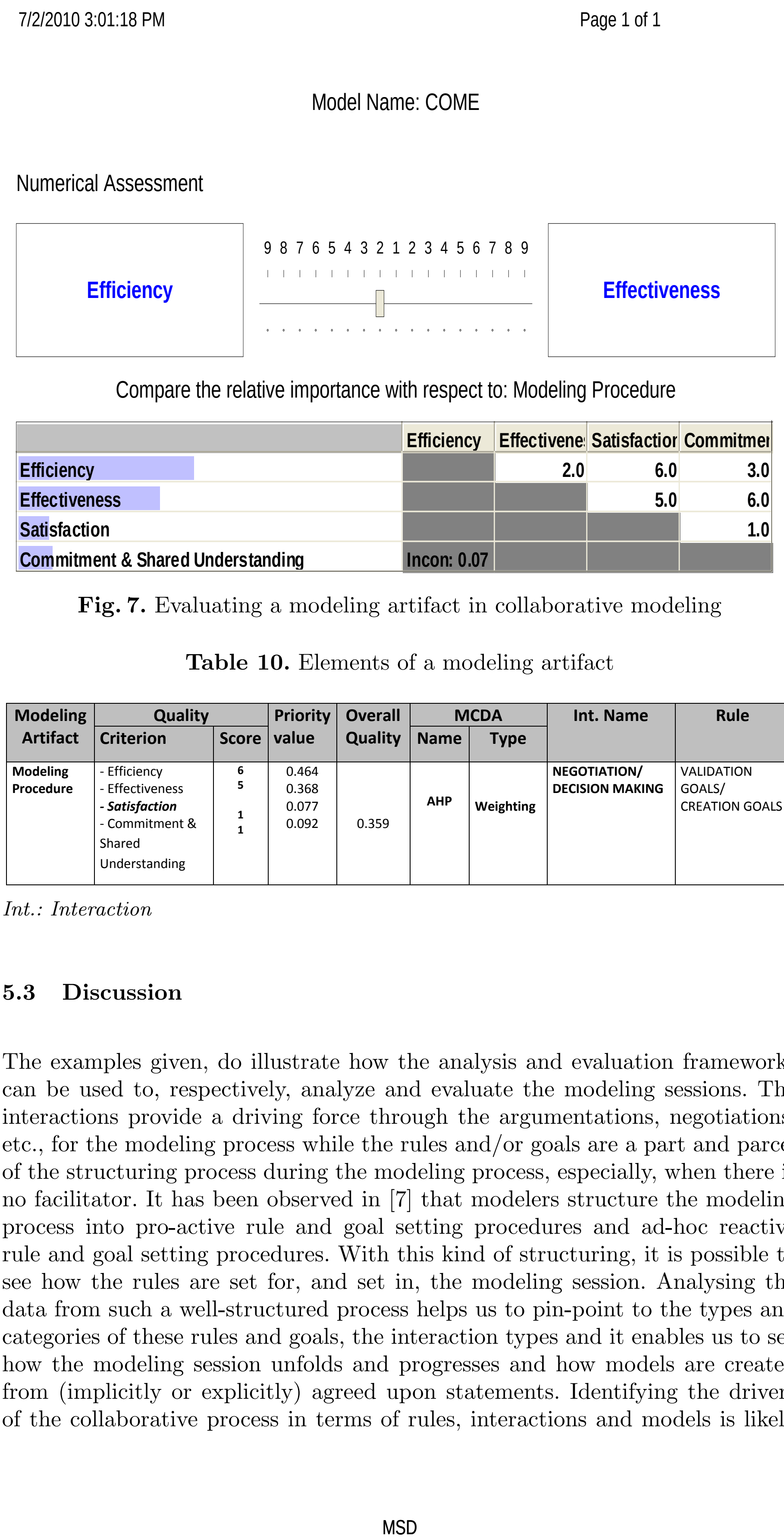


Fig. 7. Evaluating a modeling artifact in collaborative modeling

Table 10. Elements of a modeling artifact

Modeling Artifact	Quality Criterion	Quality Score	Priority value	Overall Quality	MCDA Name	MCDA Type	Int. Name	Rule
Modeling Procedure	- Efficiency - Effectiveness - ***Satisfaction*** - Commitment & Shared Understanding	6 5 1 1	0.464 0.368 0.077 0.092	0.359	**AHP**	**Weighting**	**NEGOTIATION/ DECISION MAKING**	VALIDATION GOALS/ CREATION GOALS

Int.: Interaction

5.3 Discussion

The examples given, do illustrate how the analysis and evaluation frameworks can be used to, respectively, analyze and evaluate the modeling sessions. The interactions provide a driving force through the argumentations, negotiations, etc., for the modeling process while the rules and/or goals are a part and parcel of the structuring process during the modeling process, especially, when there is no facilitator. It has been observed in [7] that modelers structure the modeling process into pro-active rule and goal setting procedures and ad-hoc reactive rule and goal setting procedures. With this kind of structuring, it is possible to see how the rules are set for, and set in, the modeling session. Analysing the data from such a well-structured process helps us to pin-point to the types and categories of these rules and goals, the interaction types and it enables us to see how the modeling session unfolds and progresses and how models are created from (implicitly or explicitly) agreed upon statements. Identifying the drivers of the collaborative process in terms of rules, interactions and models is likely to enable development of guidelines that can be used in the development of an automated support tool for the analysis.

Figure 7 and Table 10 show, respectively, how the evaluation of the modeling process and the associated artifacts can be done and how the modelers' priorities can be aggregated. There are a number of modeling artifacts that are used in and developed during a collaborative modeling session. These include the modeling language, the modeling procedure, the models, and the support tool or medium. Analyzing what takes place during the modeling process, and what drives the modeling process won't be complete unless we assess and evaluate the quality of all these modeling artifacts. Evaluation is quite important since it gives assurance about the quality of these artifacts and through the meta-model we can trace the flaws in the modeling process back to the analysis. One key observation is that the modeling artifacts' quality dimensions can be assigned quality scores during a negotiation and decision making (interactive) process using a multi-criteria decision analysis technique, e.g., AHP [24], where the modelers' different priorities, preferences are reconciled and aggregated, and the overall quality is finally obtained by synthesizing the priorities. Rules and/or goals play a role since they direct and guide the modeling process.

6 Conclusion and Future Research

The contribution of the paper is twofold. First, it shows how the collaborative modeling process can be analyzed through the RIM framework and how it can be evaluated through the MCDA evaluation framework. Second, it develops a meta-model which unifies the analysis framework and the evaluation framework. To test the soundness of the meta-model, we provided illustrative examples from real modeling sessions. Though simple in description, these examples bring out well the concepts discussed for the meta-model. One key observation is that the types or names of the identified interactions are similar to those identified by Walton and Krabbe [25][26] in *"Argumentation Theory"*, with the exception of the "eristic" dialogue.

Future Research Direction. For future research, we intend to apply the meta-model to modeling sessions, especially empirical tests with experts in industry to further test the theoretical significance and practical relevance and importance of the meta-model. More specifically, we intend to further study and analyze the modeling process using a number of other factors other than those concentrated on in this paper, e.g., dialogue games and argumentation process through negotiation from a number of perspective, e.g., multi-agents, (see for example, [27,28]). We further intend to test our a priori hypothesis about the interdependencies of the modeling artifacts and how the quality of one affects the quality of the other. We hypothesize that the the modeling language and the support tool are independent whereas the modeling products (models) and the modeling procedure are dependent variables in a multi-actor multi-criteria modeling session. Our intention is to empirically study this interdependency. Establishing this relationship is key in helping develop guidelines for a support tool that automates the analysis and evaluation of the modeling process.

References

1. Gjersvik, R., Krogstie, J., Følstad, A.: Participatory Development of Enterprise Process Models. In: Krogstie, J., Halpin, T., Siau, K.(eds.), Information Modeling Methods and Methodologies, pp. 195–215. IGI Global (2005)
2. Stirna, J., Persson, A.: Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice. In: Proper, H.A., Halpin, T.A., Krogstie, J. (eds.), Proceedings of the EMMSAD 2007, held in conjunctiun with CAiSE'07, pp. 97–106. Tapir Academic Press, Trondheim, Norway (2007)
3. Barjis, J., Kolfschoten, G.L., Verbraeck, A.: Collaborative Enterprise Modeling. In E. Proper, F. Harmsen, and J.L.G Dietz (eds.): PRET 2009, LNBIP 28, pp. 50–62. Springer Heidelberg (2009)
4. Veldhuijzen van Zanten, G., Hoppenbrouwers, S.J.B.A., Proper, H.A.: System Development as a Rational Communicative Process. J. of Systemics, Cybernetics and Informatics 2(4), 47-51 (2004)
5. Rittgen, P.: Negotiating Models. In Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS 4495, pp. 561-573. Springer Heidelberg (2007)
6. Clark, H.H., Brennan, S.E.: Grounding in Communication. In: Resnick, L.B., Levin, J.M., Teasley, S.D. (eds.) Perspectives on Socially Shared Cognition, pp. 127–149. American Psychology Association, Washington (1991)
7. Ssebuggwawo, D., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Interactions, Goals and Rules in a Collaborative Modeling Session. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP 39, pp. 54-68. Springer Berlin Heidelberg (2009)
8. Pruit, D.G.: Negotiation Behaviour. Academic Press NewYork (1981)
9. Raiffa, H.: The Art and Science of Negotiation. Havard University Press, Cambridge, MA (1982)
10. Rosenschien, J., Zlotkin, G.: Designing Conventions for Automated Negotiations among Computers. MIT Press, (1994)
11. Amgoud, L., Prade, H.: Generation of Different Types of Arguments in Negotiation. In: Delgrande, J.P.,Schaub, T. (eds.), Proceedings of the 10th International Workshop on Non-Montonic Reasoning, Whistler, Canada, June 2-5, 2004, pp. 10–15 (2004)
12. McBurney, P., van Eijk, R.M., Parsons, S, Amgoud, L.: A Dialogue-game Protocol for Agent Purchase Negotiations. J. of Auto. Agents and Multi-agents Sys 7(3), 235–273 (2003)
13. Parsons, S, Jennings, N.R.: Negotiation Through Argumentation: A Preliminary Report. Proceedings of the 2nd International Conference on Multiagent Systems, Kyoto Japan, pp. 267–274 (1996)
14. Parsons, S., Sierra, C., Jennings, N.: Agents that Reason and Negotaiate by Arguing. J. Logic Comput. 8(3), 261–292 (1998)
15. Lâasri, B., Lâasri, H., Lander, S., Lesser, V.: A Generic Model for Intelligent Negotiating Agents. Int. J. of Cooperative Information Systems 1, 291–317 (1992)
16. Sandholm, T., Lesser, V.: Issue in Automated Negotiation and E-Commerce. Extending the Contract Net Protocol. In Proceedings of the 1st International Conference on Muti-agent Systems, AAAIPress: Menlo Park, CA, USA, pp. 328–335 (1995)
17. Sierra, C., Dignum, F.: Agent Mediated Electronic Commerce: Scientific and Technological Road-map. In Dignum, F. and sierra, C. (eds.), Agent Mediated Electronic Commerce 1991. LNAI 1991, pp. 1–18. Springer-Verlag-Berlin (2001)

18. Sackler, M.L.: The Unspoken Message. Modern Psychoanalysis 23, 53–62 (1998)
19. Searle, J.R.: Speech Acts. an Essay in the Philosophy of Language. London, Cambridge University Press (1969)
20. Winograd, T., Flores, F.: Understanding Computers and Cognition. A New Foundation for Design. Norwood, Ablex (1986)
21. Goldkuhl, G.: Conversational Analysis as a Theoretical Foundation for Language Action Approaches? In: Weigand, H., Goldkuhl, G., de Moor, A. (eds.). In: 8th International Working Conference on the Language Action Perspective on Communication Modelling (LAP 2003), pp. 51–69. Tilburg, The Netherlands (2003)
22. Halpin, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufmann Pub (2001)
23. Ssebuggwawo, D., Stijn Hoppenbrouwers, S.J.B.A., Proper, H.A. (2009). Evaluating Modeling Sessions Using the Analytic Hierarchy Process. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP 39, pp. 69-83. Springer Berlin Heidelberg (2009)
24. Saaty, T.L.: The Analytic hierarchy Process. McGraw-Hill, New York (1980)
25. Walton, D., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, N.Y. (1995)
26. Reed, C., Norman, T.J.: Argumentation Machines: New Frontiers in Argument and Computation.Kluwer Academic Publishers, Dordrecht, The Netherlands (2004)
27. Hindricks, K., Jonker, C.M., Tykhnov, D.: Negotiation Dynamics: Analysis, Concessions Tactics and Outcome.In: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Multi Agent Technology, pp. 427–433 (2007)
28. Markus, M.G., Gerhard, W.: A Generic Framework for Argumentation Based Negotiation. In: Cooperative Multi Agents. LNCS 4676, pp. 209–233 (2007)

Session 4:

Architecture, Ontology & Model Engineering

On Enterprise Architecture Conformance and Benefits

Ralph Foorthuis[1], Marlies van Steenbergen[2], Nino Mushkudiani[3], Wiel Bruls[4], Sjaak Brinkkemper[5] and Rik Bos[5]

[1] UWV Business Services, La Guardiaweg 116, 1040 HG Amsterdam, The Netherlands, ralph.foorthuis@uwv.nl
[2] Sogeti Nederland BV, Hofsemolenweg 5, 8171 PM Vaassen, The Netherlands, marlies.van.steenbergen@sogeti.nl
[3] Statistics Netherlands, Henri Faasdreef 312, 2492 JP The Hague, The Netherlands, n.mushkudiani@cbs.nl
[4] IBM Netherlands NV, David Ricardostraat 2-4, 1066 JS Amsterdam, The Netherlands, wiel_bruls@nl.ibm.com
[5] Utrecht University, Institute of Information and Computing Sciences, Padualaan 14, 3584 CH Utrecht, The Netherlands
{S.Brinkkemper, R.Bos}@cs.uu.nl

Abstract. This three-page document is an abstract of the paper "On Course, But Not There Yet: Enterprise Architecture Conformance and Benefits in Systems Development", which is accepted for presentation at the Thirty First International Conference on Information Systems (ICIS 2010), St. Louis, Missouri, USA.

Keywords: IS Projects, Enterprise Architecture, Determinants of Conformance, Benefits.

Introduction

By providing holistic overviews and high-level constraints, guidelines and logic, Enterprise Architecture (EA) aims to achieve coherent and goal-oriented organizational processes, structures, information provision and technology. Various claims have been made regarding the application and effectiveness of EA, by academics and practitioners alike. At the level of the *entire organization*, for example, benefits in the reduction of complexity and realization of business/IT alignment have been claimed. At the level of *individual projects*, costs and risks are said to be reduced when complying with (i.e. conforming to) EA. Since EA conformance is crucial for gaining the aforementioned benefits, various techniques for stimulating project compliance with EA are suggested in the literature. This study aims to critically and empirically verify various claims regarding EA benefits and conformance. The research question therefore is: *What benefits can be gained by conforming to EA, and what are the most effective techniques for achieving conformance?* The research method for testing the hypotheses is an online survey (n=293) and subsequent statistical analysis (binomial tests, chi-square tests, multivariate ordinal regression).

Results

In general, we can conclude that most of the identified *techniques* for encouraging compliance with EA are used regularly, such as compliance assessments of projects, management propagation of EA, Project Start Architectures (i.e. architecture contracts), architects providing assistance to or participating in projects, organized knowledge exchanges, and the use of document templates. Quite notable is the fact that financial incentives and disincentives are used only very rarely.

In terms of the *organization as a whole*, several benefits can be attributed to EA. However, it seems that most positive perceptions are held regarding the sub-goals, whereas the ultimate goals are not being judged as positively. EA, for example, proves to be a good instrument to provide insight into the complexity of the organization, whereas it is less suitable for actually controlling it. In addition, EA is an effective means to depict a clear image of the desired future situation and a good instrument to standardize, integrate or eliminate redundant processes and systems. However, EA does not seem to be a good instrument to control costs. EA is also not a highly effective means for achieving business/IT alignment.

Regarding the benefits for *individual projects*, EA simply does not seem to offer projects very much in the way of time and cost savings, but it does clearly increase delivered quality. In addition, EA offers better management of project risks and complexity. Furthermore, projects conforming to EA do not get initialized faster than projects that do not have to conform. In fact, a majority of the respondents indicate that projects actually start up (much) slower than projects not conforming to EA. This is probably due to the additional commitment that EA brings to bear on projects (such as getting acquainted with architectural standards and balancing possible conflicts between local and enterprise-wide interests).

Another interesting finding is that EA creators (e.g. enterprise architects) generally tend to have more positive perceptions regarding EA than EA users (e.g. project members). This can probably be attributed to their involvement and commitment. In their turn, EA users can not view the overall picture due to their local focus. Moreover, they have to deal with an additional effort when conforming to EA, which may temper their enthusiasm regarding EA. Therefore, in order to have a balanced view, it is of paramount importance that EA research takes both perspectives into account.

Using multivariate ordinal regression, we studied the effects of the techniques for stimulating compliance on project conformance, as well as the effects of project conformance on achieving the benefits for the organization and individual projects. Figure 1 presents some of the most interesting findings (excluding the statistics).

Three techniques prove to be important determinants of project conformance: compliance assessments of projects, management propagation of EA, and assistance to projects. In general we observe that the more a technique is used, the higher the level of conformance achieved. Being assessed on compliance has the largest effect on whether projects will actually conform. The fact that a project will be explicitly confronted with its nonconformance apparently stimulates them to comply with the norms. This could be due to the fact that carrying out compliance assessments is an indication of the importance of conforming or, alternatively, simply to the project's

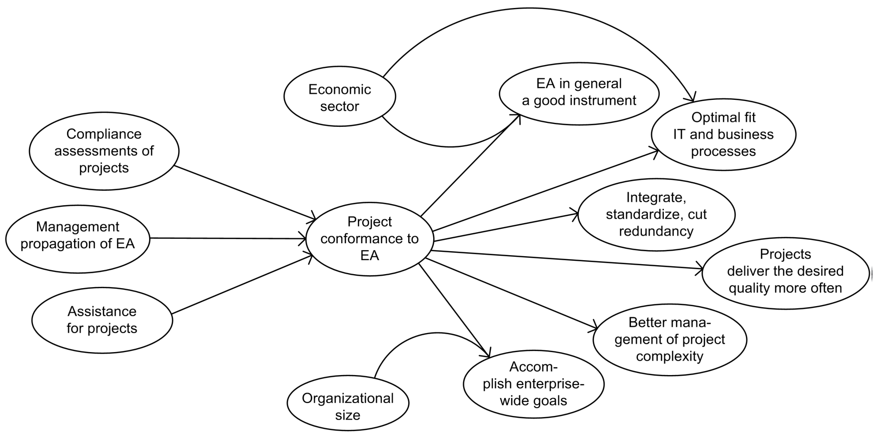


Fig. 1. The empirical model for EA conformance and benefits.

desire to avoid confrontation. Management propagation of the importance of EA has the second largest influence. Third in rank is providing assistance to the projects when applying the EA's rules and guidelines.

In its turn, conformance has significant influence on several EA-related benefits. Four *project benefits* could be attributed to conformance, namely delivering more of the desired functionality and quality, and better management of complexity and risks. If there is a high level of conformance, then projects conforming to EA – compared to non-conforming projects – are likely to achieve higher levels of these benefits.

Project conformance is also positively associated with *enterprise-wide benefits*. The strongest relationships were found for accomplishing enterprise-wide goals, achieving an optimal fit between IT and business processes, and also integrating, standardizing or eliminating redundancy from related processes and systems. These represent some of the key aims of Enterprise Architecture, the achievement of which in the respondents' experience is dependent on project compliance with EA. Especially for the last two benefits the important role of projects is not difficult to see: business/IT alignment in processes and integrating several individual systems are typically EA-related issues, but the organization is highly dependent on projects for actual implementation. It is therefore not entirely unexpected to find strong relationships with project conformance here. Weaker, but still statistically significant associations were found for other important goals, such as achieving organizational agility and providing insight into and controlling the complexity of the organization. See the full paper for an overview of all significant effects.

Particular interesting is the fact that project conformance seems to have more impact on organization-wide benefits than on project-level benefits. This demonstrates that (project conformance to) EA is indeed an important factor in achieving enterprise-wide benefits and goals. Contextual factors, such as the economic sector and organizational size also have a (moderate) influence. To conclude, this study has shown that EA offers different kinds of value, but that additional effort is required from the IS community to fulfill more of its promised potential.

Designing for Innovation: Using Enterprise Ontology Theory to Improve Business-IT Alignment

Philip Huysmans, Kris Ven, and Jan Verelst

University of Antwerp, Belgium
{philip.huysmans,kris.ven,jan.verelst}@ua.ac.be

In today's economy, innovation plays an increasingly important role in the strategy of organizations. Managers therefore need to understand and be able to manage the innovation process. It has been noted that "*[a]t a time when so much attention is given to innovation and entrepreneurship, it is rather pathetic that a deep understanding of the process is lacking. It is no wonder that firms and governments have difficulty trying to stimulate (and manage) innovation when its fundamental processes are so poorly understood.*" [2, p. 3].

In our research, we aim to contribute to a more effective innovation process by focusing on the structure of organizational artefacts which allows the realization of innovations. This research direction is framed in the emerging scientific area called Enterprise Engineering. Enterprise Engineering builds on existing approaches including organizational theory and information systems sciences in order to purposefully design organizations. The ability to adopt innovations depends largely on the ability to realize changes to the organizational artefacts. There is consensus in literature that information technology (IT) is an important enabler for the implementation of innovations. Most enterprise architecture frameworks acknowledge the importance of aligning the information technology (IT) infrastructure with the enterprise architecture. Therefore, the recent research efforts in the enterprise architecture domain are very relevant for innovation research in enterprise engineering.

This paper, which was originally presented at the ICITIE conference [1], contributes to insights regarding three important issues in the enterprise architecture domain. First, organizations are competing in increasingly volatile environments. In such environments, no long-term competitive advantages can be obtained, and organizations need to strive towards realizing a succession of short-term competitive advantages. Therefore, the enterprise and its supporting IT architecture should be well-aligned, in order to be able to quickly adapt to changing environments.

Second, many enterprise architecture frameworks have a descriptive, rather than prescriptive nature. Although such frameworks are able to describe the original and the revised structure of the organization, it remains unclear why the applied changes resulted in a desirable outcome for the organization. This insight is essential to be able to repeat the architectural process in the future. Consequently, using these frameworks does not contribute to the understanding of the innovation process.

Third, it has been observed that enterprise architecture frameworks constitute a heterogeneous collection without clear integration, and that their theoretical foundations are currently still limited. A common theoretical foundation at the organizational and information systems level could improve alignment, and could provide prescriptive rules for the architecture. In our research, we propose modularity as a theoretical foundation. Modularity has already been used at both the organizational and information systems level.

As argumented by Enterprise Engineering, a systematic and scientific approach for constructing enterprise architectures should contribute to a solution for these three issues. Therefore, we analyze in this paper a case study of a successful enterprise architecture project with these issues in mind. In the case study, we focus on a European organization that is able to realize substantial improvements in implementing innovations by aligning its IT and enterprise architecture. In both architectures, the same modular structure is used to create loosely coupled entities. At the IT level, the application portfolio consists of loosely coupled applications which are based on high-level, stable business activities from the enterprise architecture. This approach allows business and IT staff to implement changes more quickly, whereas IT used to be considered as a bottleneck in innovation projects. More specifically, we identify advantages in the areas of alignment, change assessment, reuse, and improvements to the development process. Notwithstanding the successful outcome of this enterprise architecture project, the approach taken by the organization strongly relies on the heuristic knowledge of employees: no guidelines or principles are used to identify the appropriate granularity of the modules. As a result, the repeatability and reproducibility of their approach is limited, and no explicit knowledge about the alignment is gained. In order to develop a better understanding of the process elements which can lead to the identified results, a more systematic method is proposed.

Based on the insights from practice, we therefore attempt to contribute to a more systematic method to construct enterprise architectures. We take a design science approach by repeating the enterprise architecture project using the Enterprise Ontology theory. Our results show that the model created by following the Enterprise Ontology theory was very similar to the model created by the organization, which is a desirable result. The main advantage of Enterprise Ontology is that it provides a more repeatable and reproducible result and that the resulting models are more evolvable. This result shows that scientific research in Enterprise Engineering can be relevant and applicable.

References

1. Huysmans, P., Ven, K., Verelst, J.: Designing for innovation: Using enterprise ontology theory to improve business-it alignment. In: Shishkov, B., Tsihrintzis, G.A., Virvou, M. (eds.) Proceedings of the 1st International Conference on IT-enabled Innovation in Enterprise (ICITIE 2010). pp. 177–186. SciTePress (July 2010)
2. Teece, D.J. (ed.): The Competitive Challenge: Strategies for Industrial Innovation and Renewal. Ballinger Publishing Company, Cambridge, MA (1987)

EXTENDED ABSTRACT
Developing a business model engineering & experimentation tool – the quest for scalable 'lollapalooza confluence patterns'[1]

Björn Kijl[*], Durk Boersma[*]

[*] University of Twente, School of Management & Governance, PO Box 217, 7500 AE Enschede, The Netherlands
{b.kijl, d.boersma}@utwente.nl

Keywords: Business models, action design research, business model engineering, business model dynamics, business model experimentation, business model management, business model innovation, growth & deployment strategies, lollapalooza tendencies, Internet services, service innovation, entrepreneurship, startups.

1 The growing importance of business model innovation

In a globalizing world where consumer electronics (CE), information technology (IT), telecom, and media are converging, opportunities for new Internet services are emerging [1, 2]. With increasing marketplace dynamics, shorter time-to-market cycles and rapid technological developments, the ability to imagine and combine different, formerly separated, technological capabilities in order to facilitate new and useful value propositions for users and customers will be critical [3]. To be able to offer these value propositions with new Internet services in a sustainable manner, new viable business models need to be developed [4-6] – in the end, every service needs a viable business model.

2 Struggling with business models

Essentially, a business model can be seen as a definition of the way by which an organization or group of organizations delivers value to customers, entices them to

[1] Recommended Citation:
Kijl, Björn and Boersma, Durk, "Developing a business model engineering & experimentation tool – the quest for scalable 'lollapalooza confluence patterns'" (2010). *AMCIS 2010 Proceedings*. Paper 567. http://aisel.aisnet.org/amcis2010/567

pay for value and converts those payments to profit [6-8]. Business model research is a relatively new field of research – the concept of a business model has no established theoretical grounding yet in economics or business studies [7]. A business model supports simulating, analyzing and understanding current or new business concepts and exploiting these concepts [9, 10]. Business model design can be seen as a key decision for new firm entrepreneurs [11] and with increasing technological possibilities, more and more companies are struggling on the level of business model innovation instead of on the level of technological innovation [12].

A lot of high tech start-ups still fail by focusing too much on technology instead of on business model aspects like value network structures, revenue models and value propositions – they just ‘forget’ to consciously think about the quality and logic of their underlying business model and directly start writing a business plan [13]. Having a great technology is no guarantee for success; a viable business model may even be more critical [12-15]. About 90% of the start-ups have to change their business model before going to market – for them it is critical to know as early as possible whether a business model design is viable or not [3, 8, 16].

3 An action design study of business model experimentation

Currently, most business model research is focused on business model design, whereas there is almost no attention for validation and implementation of business models [6, 8]. The goal of the research as described in this paper is to test a business model engineering method supporting business model experimentation as a continuous design, validation and implementation cycle. The method is applied to an online investment research start-up in the form of an in depth action design study [17, 18] – the author of this abstract co-founded and also still manages the company and its business model – of the companies’ business model evolution as a continuous design, validation and implementation cycle as well as the related experimentation and effectuation processes [15, 19] – from the establishment of the company in 2007 till the beginning of 2010. The start-up started with an easily scalable freemium business model [10] by offering a free weekly investment column to a small mailing list of about 100 Dutch and Belgian investors and a related paid monthly stock analysis service based on value investing principles [20]. Directly from the beginning, as many processes as possible were automated by making use of e.g. online payment systems, mailing systems, content protection systems and membership management systems. In about two years, related business model experimentation and effectuation actions – focusing on creating nonlinear, so called ‘lollapalooza’ growth patterns [21] – led to a strong underlying growth: the list size multiplied more than 300-fold, it led to a Dutch best seller on investing principles as well as the introduction of a new premium subscription service.

Based on the results of the action design study as described above, the business model engineering method as developed and tested in earlier research [see e.g. (Haaker and Kijl 2008) and (Kijl et al. 2010)] will be further discussed, improved and extended in the paper.

References

1. Standage, T., Your television is ringing. The Economist: A Special Report on the Future of Telecoms, 2006: p. 14-20.
2. Klemettinen, M., Enabling technologies for mobile services: The MobiLife book. 2007: Wiley.
3. Kijl, B., et al., Developing a dynamic business model framework for emerging mobile services, in ITS 16th European Regional Conference. 2005: Porto, Portugal.
4. Haaker, T., et al. Challenges in designing viable business models for context-aware mobile services. in 3rd International CICT Conference (Mobile and Wireless Content, Services and Networks). 2006. Technical University of Denmark, Center for Information and Communication Technologies, Kongens Lyngby, Denmark.
5. Haaker, T., et al. Challenges in designing viable business models for context-aware mobile services. in CollECTeR Iberoamérica. 2007. Córdoba, Argentina.
6. Bouwman, H., et al., Conceptualizing the STOF Model, in Mobile Service Innovation and Business Models, H. Bouwman, T. Haaker, and H. De Vos, Editors. 2008, Springer Verlag: Berlin. p. 31-70.
7. Teece, D.J., Business Models, Business Strategy and Innovation. Long Range Planning. In Press, Corrected Proof.
8. Kijl, B. and L.J.M. Nieuwenhuis, Deploying a Telerehabilitation Service Innovation: An Early Stage Business Model Engineering Approach, in HICSS-43. 2010: Hawaii.
9. Osterwalder, A., Y. Pigneur, and C. Tucci, Clarifying business models: Origins, present, and future of the concept. Communications of the Association for Information Systems, 2005. 16(1): p. 1-25.
10. Osterwalder, A. and Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. 1st Edition ed. 2009: OSF.
11. Zott, C. and R. Amit, Business Model Design: An Activity System Perspective. Long Range Planning. In Press, Corrected Proof.
12. Chesbrough, H., Open business models: How to thrive in the new innovation landscape. 2006: Harvard Business School Press.
13. Allen, K., Growing and managing an entrepreneurial business. 1999.
14. Pieper, R., V. Kouwenhoven, and S. Hamminga, Beyond the Hype-e-Business Strategy in Leading European Companies. 2002.
15. Chesbrough, H., Business Model Innovation: Opportunities and Barriers. Long Range Planning. In Press, Corrected Proof.
16. Kalles, v., B., Subsidie aan starter werkt averechts. Het Financieele Dagblad, 2009.
17. Rossi, M. and M. Sein, Design research workshop: a proactive research approach. Presentation delivered at IRIS, 2003. 26: p. 9–12.
18. Sein, M., et al., Action Design: Where Design Meets Action In Research. MIS Quarterly, 2010. conditionally accepted for publication.
19. Sarasvathy, S., Effectuation: elements of entrepreneurial expertise. 2008: Edward Elgar Publishing.
20. Oude Nijhuis, H. and B. Kijl, Leer beleggen als Warren Buffett - Zijn beleggingsstrategie in theorie & praktijk. 2009, Arnhem: Sonsbeek Publishers. 160.
21. Bevelin, P., Seeking wisdom: from Darwin to Munger. 3rd edition ed. 2007: PCA Publications L.L.C.

Session 5:

Supply Chains

Inter Organizational Relationships Performance in Third Party Logistics: conceptual framework and case study*

Romana Aziz[1], Jos van Hillegersberg[1], Kuldeep Kumar[2]

[1] Department of Information Systems and Change Management, University of Twente
{r.aziz, j.vanhillegersberg}@utwente.nl
[2] Department of Decision Sciences and Information Systems, College of Business Administration, Florida International University, USA
kumark@fiu.edu

Extended abstract: This paper presents some of the results of the **I**nter-organizational **S**upply-chain **EA**sing (ISEA) project which was targeted at improving control and coordination in buyer – supplier relationships between Shell Chemicals Europe (SCE) and their portfolio of logistic service providers (LSPs). ISEA project was embedded in the larger Transumo Diploma project which was funded by the Dutch Government (www.transumo.nl).

At the time of this research SCE was experiencing day-to-day firefighting, which occurred when unresolved issues bounced back from the LSPs to SCE's staff. Moreover SCE was also facing difficulties in managing their relationships with the LSPs. SCE did not feel that they had enough leverage over their LSPs and they did not understand why some LSPs were causing more work relative to other LSPs. The senior SCE managers were wondering if they had contracted the right mix of LSPs.

ISEA project investigated the buyer supplier relationships in third party logistics from four related angles, i.e., why some LSPs perform better than others? how can we select an optimum LSP portfolio and how can we improve the performance of portfolio by reallocating volume levels among LSPs? what process and IT related issues negatively impact logistics process? and how can we solve issues that negatively impact logistics process?

This paper presents a part of our research conducted along the first theme, i.e., why some LSPs perform better than others? In this paper our main research question is how can we analyze and understand the relationship performance of a shipper with its network of LSPs? The main research question leads to three sub questions: 1) What are the main factors that affect inter organizational relationship performance in third party logistics? 2) How do these different factors affect performance in third party logistics? and 3) How do these factors affect relationship performance in a real life context? We developed a conceptual framework to answer the first two sub questions and this framework served as a conceptual lens for our empirical research in the form of a case study to answer the third sub question.

Based on review and integration of current literature and exploratory interviews with practitioners we propose a conceptual framework and propositions centered on inter organizational relationship performance factors in third party

logistics. Five main dimensions of inter organizational relationships are identified which affect performance in third party logistics:

1. Commitment: Based on information sharing, goal congruence, power imbalance in favor of shipper and trust.
2. Adaptation: Based on supplier dependencies and learning orientation.
3. Conflict resolution: Based on control and relational norms.
4. Compatibility: Based on planning capabilities, IT use and management structure.
5. Communication: Based on quality and formality.

In order to assess the validity of our conceptual model we include a case study in this paper. The case study is based on Shell Chemicals Europe and their portfolio of seventeen third party logistic service providers. The data for this case study was collected via observation, interviews, documents and questionnaires. The data collected during the case study generated insights that confirmed our proposed propositions.

The main contribution of this paper is in the form of an original framework describing the elements of relationships that affect performance in third party logistics. The findings presented in this paper are relevant for practitioners and academics. Practitioners can use these findings as a prescriptive resource while targeting their efforts for performance improvement in logistics outsourcing and use these insights to develop effective relationship management strategies. Our contribution for the academics is in the form of a conceptual framework and propositions for relationship performance factors in third party logistics and testing of propositions in a qualitative case study. The framework and the propositions can serve as a useful basis for the development of a more general theory for relationships development in third party logistics.

Keywords: Third party logistics, logistics performance factors, inter organizational relationships, logistics portfolio.

[*] **Original Contribution**: Romana Aziz, Jos van Hillegersberg, Kuldeep Kumar, (2010) Inter Organizational Relationships Performance in Third Party Logistics: conceptual framework and case study, In: Pioneering Solutions in Supply Chain Management. A Comprehensive Insight into Current Management Approaches (Kersten W., Blecker T., Luthje C.) Erich Schmidt Verlag GmbH &Co. KG, Berlin 2010, Vol.Operations and Technology Management Series Vol. 14 pp.105-126, ISBN:9783503126859

Architecture Support for Flexible Chain Management

R. Seguel, R. Eshuis, and P. Grefen

Information Systems Group, School of Industrial Engineering,
Eindhoven University of Technology, The Netherlands.
{r.e.seguel,h.eshuis,p.w.p.j.grefen}@tue.nl

Abstract. In competitive markets, organizations collaborate in business chains using dynamic service outsourcing to deliver complex products and services. To enable the flexible formation of business chains, organizations can use protocol adaptation to ensure that their business protocols are compatible. In this paper, we present three different software architectures that enable the flexible formation and enactment of different chain structures, in which the protocol adaptation component is a key enabler. We show the feasibility of the approach by extending software architecture definitions from the literature for each flexible chain formation case.

Keywords: Business Chains, Dynamic Service Outsourcing, Interacting Services, Protocol Adaptor, Service Adaptation, Software Architecture

1 Introduction

Nowadays, the production of complex products and services in competitive markets involves a number of autonomous organizations [8] that collaborate in business chains. By locating the customer order decoupling point (CODP) in the business chain we identify three chain structures: demand chain, supply chain and hybrid demand/supply chain [13]. The CODP indicates how deeply the customer order penetrates into a business chain. Due to the shorter life-cycles and increasing complexity of products and services [6], the organizations in a business chain collaborate in a just-in-time fashion, using dynamic service outsourcing. In dynamic service outsourcing, an organization outsources a part of its business process, for instance order fulfillment, to a partner that is selected at the last possible moment from the marketplace [6]. This way, the organizations collaborate by invoking functionalities from each other according to their business protocols in their public process view [2, 4]. Each public process view abstracts an underlying private business processes that is executed by the organization.

In current dynamic service outsourcing approaches [7,8], the tacit assumption is made that interacting protocols are compatible: each sent message is received and processed by the other party, and thus no deadlocks occur. However, collaborating organizations have their own protocol that specifies their own way of working and they may easily have incompatible protocols. Since organizations collaborate in a just-in-time fashion, incompatible business protocols hinder the flexible formation of business chains. To configure business chains in a flexible way the organizations can use protocol adaptation to ensure that their business protocols are compatible [13].

The goal of this paper is to present a supporting software architecture for each flexible chain formation case that we described in [13]. The software architectures are an extension and integration of two software architectures from the literature [7,8] that support the formation and enactment of supply and demand chain networks. The presented software architectures include business protocol adaptation as a key component to support the flexible configuration and enactment of business chains. The protocol adaptation component constructs an adaptor to resolve (if possible) incompatibilities between interacting services during chain formation, using any existing adaptation method [1, 10–12, 14–16].

Although there is some work on cross-organizational workflow collaboration using process views [2, 8, 9] they do not include adaptation in their architecture definitions. In this paper, we focus on adaptation of behavioral mismatches rather than interface mismatches. An interface mismatch is due to differences in the formats and specifications of messages exchanged that can be resolved using schema mapping and transformation tools [11, 15]. We will extend our approach adding interface adaptation in future work. However, we do not expect that this actually impacts the presented software architectures.

The contributions of this paper are three software architectures to support the flexible formation of business chain [13], in which the protocol adaptation component is a key enabler.

The remainder of this paper is organized as follows. Section 2 describes the three flexible chain formation cases. Section 3 presents the architecture support for flexible formation of demand chains. Section 4 details the architecture to enable the flexible configuration of supply chains. Section 5 describes the architecture for flexible hybrid demand/supply chain formation. Section 6 discusses a third-party adaptation factory and Section 7 presents the conclusions and future work.

2 Flexible Business Chain Management

There are three scenarios for flexible formation of business chains [13]. Each scenario defines the responsibility of a partner to construct a protocol adaptor to resolve (if possible) incompatibilities between their business protocols. We show in Figure 1 a business chain to illustrate the adaptation responsibility for each business chain formation scenario. The letters outside the services in Figure 1 represent the place where the protocol adaptor is constructed.

This way, in a *flexible demand chain formation* scenario [13], the service consumer defines its business protocol according to the customer business requirements. Since the CODP is moved to the last provider in the demand chain, the responsibility of constructing an adaptor is always of the provider; see ⓑ and ⓓ in Figure 1. In a *flexible supply chain formation* scenario [13], the service

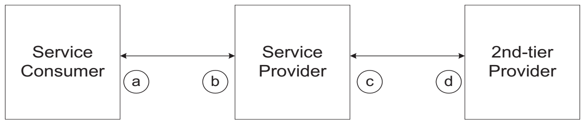


Fig. 1: Business Chain to Identify Adaptation Cases

provider sets its business protocol in conformance with market standards like SCOR [3] or eTOM [5]. For example, the service provider can be a big retail vendor like Dell. The service consumer searches the standard service provider in the marketplace to define its business protocol. In the supply chain the CODP is moved to the service consumer, and thus the responsibility of building an adaptor is always of the service consumer; see ⓐ and ⓒ in Figure 1. In a *flexible hybrid demand/supply chain formation* scenario [13], the service consumer and service provider form a demand chain while the service (1st-tier) provider and the 2nd-tier provider form a supply chain. Then, the CODP is at the service provider. This way, the responsibility of building an adaptor is always of the service (1st-tier) provider: it has to build an adaptor to resolve mismatches with the service consumer and 2nd-tier provider; see ⓑ and ⓒ in Figure 1.

To describe the flexible formation of a business chain, we explain the hybrid demand/supply chain case since it includes the other two business chain cases. We illustrate this case in Figure 2, in which the companies W and X operate in demand chain mode and companies X and Y operate in supply chain mode. Then, the company X constructs a protocol adaptor to resolve the mismatches with companies W and Y, using one of the adaptation methods presented in [1, 10–12, 14–16]. In this example we use the method presented in [14]. Note that the communication of messages is shown in the figure by the arrows crossing the organization borders, indicating send (source) and receive (target) actions. The company X constructs the protocol adaptors at the public process view and it deploys each adaptor in front of its business protocols. Then, the company X offers the protocol adaptors and its business protocol in the marketplace. Next, the companies W and Y select X, and then they deploy the business protocols in the architecture components.

3 Architecture for Flexible Demand Chain Formation

The CrossWork architecture [8] was developed to support the dynamic formation and enactment of a Network of Automotive Excellence. In the CrossWork business scenario, the service consumer is an Original Equipment Manufacturer (OEM) organization like BMW or MAN. The OEM defines a certain goal or solution objective that is sent to the service provider. Then, the provider forms the chain by selecting the 2nd-tier providers from the marketplace to meet the goal. The provider composes a global business protocol with the local protocols of the 2nd-tier providers to coordinate them. Then, the provider enacts the global process that enacts the local protocols in the 2nd-tier providers to later send the

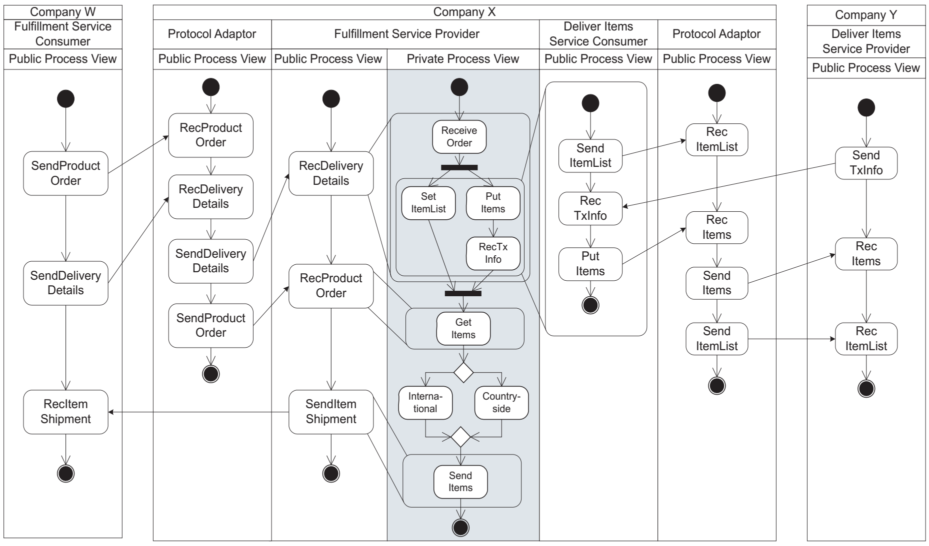


Fig. 2: Adaptation Case for Flexible Hybrid Demand/Supply Chain Formation

result back to the OEM. We extend the CrossWork architecture by adding the architecture components to support protocol adaptation. In Figure 3, we illustrate the CrossWork architecture (white boxes) plus the extension (grey boxes). The extended CrossWork architecture enables the flexible formation of a demand chain at design-time and at run-time.

At design-time, the extended CrossWork architecture is triggered by the service consumer that defines the solution objective in the goal support module according to the customer business requirements. The consumer defines its business protocol according to the goal. Then, the consumer selects the service provider from the marketplace that meets the goal. The consumer sends the goal definition and its business protocol to the service provider. Then, the service provider checks the compatibility between its business protocol and the consumer protocol. If the protocols are incompatible, the adaptor factory module constructs a protocol adaptor to resolve mismatches. The adaptor factory implements the adaptation method for tightly and loosely coupled interacting services [1, 10–12, 14–16]. Then, the provider deploys the business protocol in the protocol enactment module and the protocol adaptor in the adaptor enactment module. Similarly, the service consumer deploys its business protocol in the protocol enactment module.

Next, the service provider decomposes the goal into a required set of protocols (component services) in the goal support module. Then, the team formation module finds the 2nd-tier providers in the marketplace according to the set of protocols. Next, the composition module composes the set of protocols into a global business protocol. Note that the 2nd-tier providers are not only selected by protocol compatibility. It means that while there are parts of the global

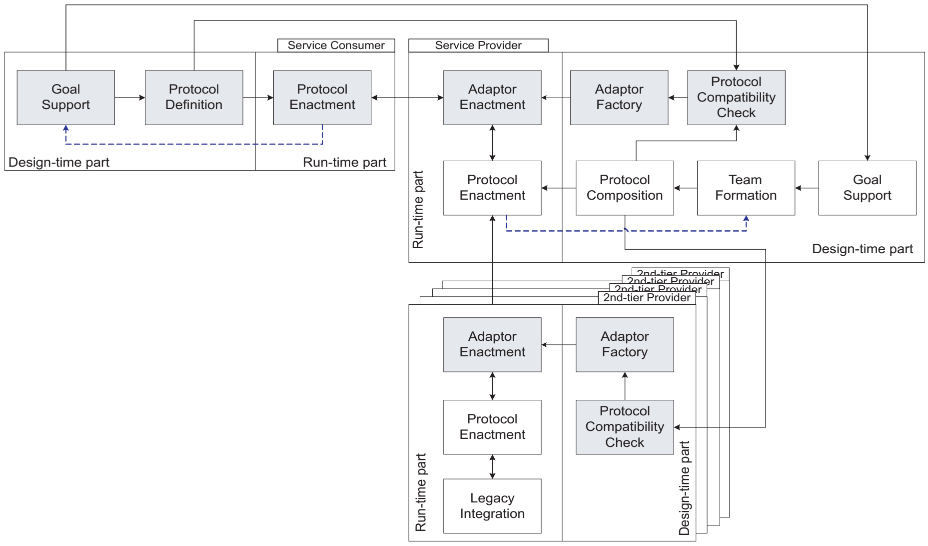


Fig. 3: Architecture Extension of CrossWork [8] for Flexible Demand Chain Management

protocol that are compatible with the 2nd-tier providers, there are other parts that are incompatible with the 2nd-tier provider protocols. This way, the 2nd-tier providers check the compatibility between their business protocols and the set of protocols that compose the global protocol. If they are incompatible, the adaptor factory at the 2nd-tier providers generates a protocol adaptor. Next, each 2nd-tier provider deploys the business protocol in the protocol enactment module and the adaptor in the adaptor enactment module. Finally, the service provider deploys the global protocol in the protocol enactment module. This way, the protocols can be later executed by the service consumer and provider, enacting the demand chain with the 2nd-tier providers.

At run-time, the extended CrossWork architecture enables the formation of the demand chain when the consumer business protocol is being enacted. This is illustrated with the ‘reverse’ dotted arrow from the protocol enactment module to the goal support module in Figure 3. This way, the consumer stops the business protocol execution to configure the demand chain (at design-time) to later continue the enactment of the business protocols. Similarly, the service provider can configure the demand chain with 2nd-tier providers when its protocol is being executed. Note that if the service provider acts as integrator only [6], then the adaptation is only needed at the 2nd-tier provider. The service provider uses the protocol sent by the consumer as global business protocol. Then, the service provider is protocol compatible with the service consumer since they interact to only synchronize the control flow of the global business protocol. The global business protocol interacts with the 2nd-tier providers, which can need adaptation. In the basic CrossWork architecture, if the global protocol cannot be

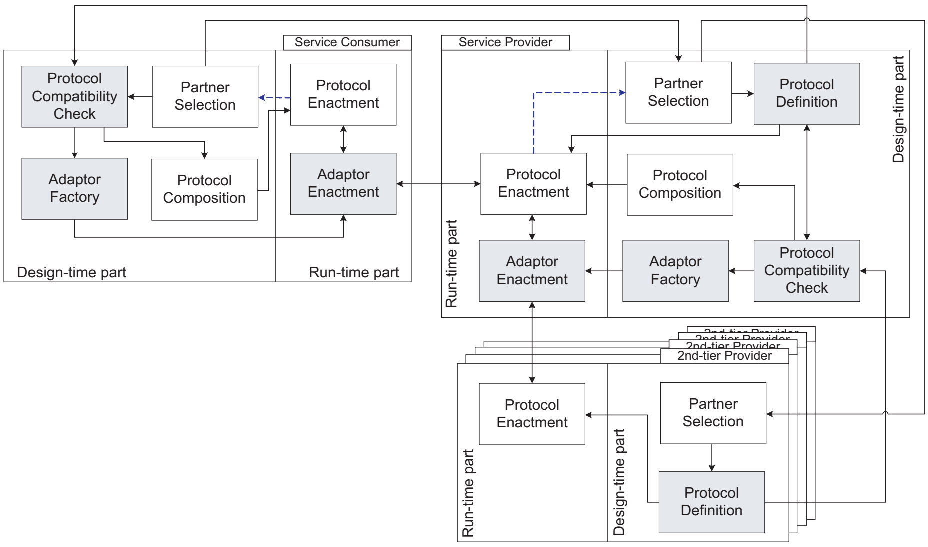


Fig. 4: Architecture Extension of CrossFlow [7] for Flexible Supply Chain Management

composed then the system backs up to the team formation or even to the goal support module to correct it [8]. However, in the extended architecture, these backtrack steps are needed only if the adaptor factory module at the 2nd-tier provider indicates the mismatch cannot be resolved and no adaptor can be constructed [12, 14]. Note that technically the business protocols and the adaptor can be enacted using the same enactment engine or two different enactment engines.

4 Architecture for Flexible Supply Chain Formation

The CrossFlow architecture [7] was developed to support the configuration of a supply chain, using dynamic service outsourcing. In the CrossFlow business scenario, the service consumer outsources a non-core part of its business protocol to a service provider. The service consumer takes the decision of outsourcing during the execution of its business protocol, and thus it selects the provider at the last possible moment. This way, the basic CrossFlow architecture is mainly focused on run-time supply chain configuration. Note that the architecture was developed in the pre web services era, but it conceptually supports the collaboration of two interacting services which share the business protocols at the public process view. We extend the CrossFlow architecture to support the flexible configuration of a supply chain at design-time and at run-time. We illustrate the basic architecture (white boxes) plus the extension (grey boxes) in Figure 4.

At design-time, the extended CrossFlow architecture is triggered by the consumer that selects the service provider from the marketplace to outsource its protocol. Then, the provider sends its standard business protocol to the consumer that checks the compatibility with its protocol. If the business protocols are incompatible, then the service consumer constructs a protocol adaptor. Next, the service consumer couples the outsourced protocol part with its business protocol in the composition module for later deployment in the enactment module. Then, the consumer deploys the coupled business protocol in the protocol enactment module and the adaptor in the adaptor enactment module. Finally, the service provider deploys its business protocol in the enactment module after it sent the protocol definition. Similarly, the service provider can outsource part of its protocol by selecting 2nd-tier providers from the marketplace. Then, the 2nd-tier providers send their standard business protocol to the provider that checks the compatibility with its protocol. If the protocols are incompatible, then the provider adaptor factory module generates the protocol adaptor. Then, the provider couples the outsourced protocol part with its business protocol in the composition module. Next, the provider deploys the coupled protocol and the adaptor in the enactment modules while the 2nd-tier providers deploy their protocols too. This way, the protocols can be later executed by the service consumer and provider, enacting the supply chain with the 2nd-tier providers.

At run-time, the extended CrossFlow architecture supports the configuration of the supply chain when the consumer business protocol is being enacted. This is illustrated with the 'reverse' dotted arrow from the protocol enactment module to the partner selection module in Figure 4. Therefore, the consumer stops the business protocol execution to configure the supply chain (at design-time) to later continue the enactment of the business protocols. Then, the service provider can also configure a supply chain with 2nd-tier providers when its protocol is being enacted. Note that if the service provider acts as integrator only [6], then the adaptation is only needed at the service consumer. The service provider preselects the 2nd-tier providers before it is selected by the consumer. Once the provider is selected, it sends the standard protocol to the consumer. The service provider is protocol compatible with the 2nd-tier providers since they interact to only synchronize the control flow of the business protocol. On the other hand, the consumer protocol interacts with the standard business protocol, which can need adaptation. Like the extended CrossWork architecture, the CrossFlow architecture technically support the enactment of the business protocols and the adaptor using the same enactment engine or two different enactment engines.

5 Architecture for Flexible Hybrid Demand/Supply Chain Formation

To support the flexible configuration of a hybrid demand/supply chain, we define the architecture as an extension of the CrossWork (demand chain) and CrossFlow (supply chain) architectures. We illustrate the architecture for flexible formation of a hybrid demand/supply chain in Figure 5. It supports the configuration of the hybrid chain at design-time and at run-time.

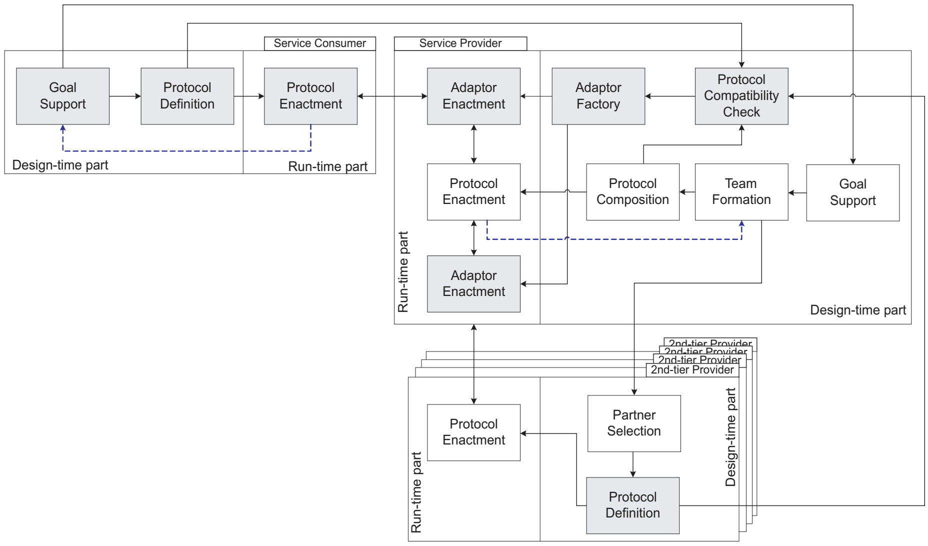


Fig. 5: Architecture for Flexible Hybrid Demand/Supply Chain Management

At design-time, the architecture is triggered by the service consumer that configures a demand chain with the provider. The consumer defines the solution objective in the goal support module according to the customer business requirements. Then, the consumer defines its business protocol according to the goal and selects the service provider from the marketplace that meets the goal. Then, the consumer sends the goal definition and its business protocol to the service provider. Next, the service provider checks the compatibility between its business protocol and the consumer protocol. If the protocols are incompatible, the adaptor factory module constructs a protocol adaptor to resolve mismatches. Then, the provider deploys the business protocol in the protocol enactment module and the protocol adaptor in the adaptor enactment module while the service consumer deploys its business protocol in the protocol enactment module.

At this stage, the service provider configures a supply chain with 2nd-tier providers. Next, the service provider decomposes the goal into a required set of protocols (component services) in the goal support module. Then, the team formation module finds the 2nd-tier providers in the marketplace according to the set of protocols. Next, the composition module composes the set of protocols into a global business protocol. Then, the 2nd-tier providers send their standard business protocol to the provider that checks the compatibility with its protocol. If the protocols are incompatible, then the provider adaptor factory module generates the corresponding protocol adaptors. Finally, the service provider deploys the global protocol and the adaptors in the enactment modules while each 2nd-tier provider deploys its protocol in the enactment module too. This way, the protocols can be later executed by the services by enacting the demand chain between the consumer and provider and the supply chain between the provider and 2nd-tier providers.

At run-time, the extended architecture supports the configuration of the hybrid demand/supply chain when the consumer and provider protocols are being enacted. This is illustrated with the 'reverse' dotted arrow in Figure 4. The consumer stops the business protocol execution to configure the demand chain (at design-time) to later continue the enactment of the business protocols. Similarly, the service provider stops the protocol execution to configure the supply chain (at design-time) with the 2nd-tier providers to later continue with the protocol enactment.

Note that if the service provider acts as integrator only [6], then the adaptation is needed for compatibility with either the 2nd-tier providers or the service consumer. In both cases the adaptor is constructed and deployed by the service provider. In the first case, like in the extended CrossWork case, the service provider uses the protocol sent by the consumer as global business protocol. Then, the provider and consumer protocols are compatible since they interact to only synchronize the control flow. Thus, the global business protocol parts interact with the 2nd-tier providers, which can need adaptation. In the second case, like in the extended CrossFlow case, the service provider pre-selects the 2nd-tier providers before it is selected by the consumer. Then, the provider has to generate an adaptor if the consumer and provider protocols are incompatible. The service provider and the 2nd-tier providers are compatible since they interact to only synchronize the control flow of the global business protocol parts.

6 Third-party Adaptor Factory

The architectures that we defined for flexible formation of business chains support the participation of a trusted third-party that provides adaptation as a service (AaaS). The third-party is an adaptor factory that constructs and enacts protocol adaptors to support the flexible configuration of business chains. This way, the service consumer, service provider or 2nd-tier providers can buy the adaptation service from the trusted third-party, and thus they do not need to add the adaptor factory module in their architectures themselves. The participation of the third-party does not cause conceptual changes to the architectures defined previously, but technically it needs to add the technology to ensure quality of services and security, which are outside the scope of this paper.

7 Conclusions and Future Work

We have presented three software architectures that considers protocol adaptation component as a key enabler of flexible chain formation. Any existing protocol adaptation approach can be used to realize the actual adaptation. The presented architectures enables the flexible formation of business chains between organizations that collaborate in a just-in-time fashion to meet a solution objective, which is very important in competitive markets.

There are several directions for future work. We plan to extend our approach adding interface adaptation to the adaptor factory. We are currently extending our approach to define a reference architecture in which the presented three software architectures can be described. Moreover, we will extend our approach to deal with adaptation of running business chains that deadlock due to the propagation of changes on the partner business protocols.

References

1. A. Brogi and R. Popescu. Automated generation of BPEL adapters. In *Proc. ICSOC'06*, pages 27–39. Springer, 2006.
2. D. Chiu, S. Cheung, S. Till, K. Karlapalem, Q. Li, and E. Kafeza. Workflow view driven cross-organizational interoperability in a web service environment. *Information Technology and Management*, 5(3-4):221–250, 2004.
3. Supply Chain Council. SCOR: Supply-Chain Operations Reference model; version 9.0, 2008. http://www.supply-chain.org.
4. R. Eshuis and P. Grefen. Constructing customized process views. *Data and Knowledge Engineering*, 64(2):419–438, 2008.
5. TM Forum. eTOM: enhanced Telecom Operations Map; release 8.0, 2008. http://www.tmforum.org/.
6. P. Grefen. *Mastering E-Business*. Routledge, 2010.
7. P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig. CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. *Computer Systems Science & Engineering*, 1(5):277–290, 2000.
8. P. Grefen, R. Eshuis, N. Mehandjiev, G. Kouvas, and G. Weichhart. Internet-based support for process-oriented instant virtual enterprises. *IEEE Internet Computing*, 13(6):65–73, 2009.
9. D-R. Liu and M. Shen. Business-to-business workflow interoperation based on process-views. *Decision Support Systems*, 38(3):399–419, 2004.
10. R. Mateescu, P. Poizat, and G. Salaün. Adaptation of service protocols using process algebra and on-the-fly reduction techniques. In *Proc. ICSOC'08*, pages 84–99. Springer, 2008.
11. H.R. Motahari Nezhad, G.Y. Xu, and B. Benatallah. Protocol-aware matching of web service interfaces for adapter development. In *Proc. WWW'10*, pages 731–740. ACM, 2010.
12. R. Seguel, R. Eshuis, and P. Grefen. Constructing minimal protocol adaptors for service composition. In *Proc. WEWST '09*, pages 29–38. ACM, 2009.
13. R. Seguel, R. Eshuis, and P. Grefen. Business protocol adaptation for flexible chain management. In Robert Meersman, Tharam Dillon, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2010*, volume 6426 of *Lecture Notes in Computer Science*, pages 438–445. Springer, 2010.
14. R. Seguel, R. Eshuis, and P. Grefen. Minimal protocol adaptors for loosely coupled services. In *Proc. ICWS'10*, pages 417–424. IEEE, 2010.
15. Z. Shan, A. Kumar, and P. Grefen. Towards integrated service adaptation - a new approach combining message and control flow adaptation. In *Proc. ICWS '10*, pages 385–392. IEEE, 2010.
16. D.M. Yellin and R.E. Strom. Protocol specifications and component adaptors. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 19:292–333, 1997.

Session 6:

Process Model Transformations

Transforming Standard Process Models to Decentralized Autonomous Entities

Pieter Hens[1], Monique Snoeck[1], Manu De Backer[1,2,3,4], and Geert Poels[2]

[1] K.U.Leuven, Dept. of Decision Sciences and Information Management, Naamsestraat 69, 3000 Leuven, Belgium
[2] Universiteit Gent, Dept. of Management Information and Operations Management, Tweekerkenstraat 2, 9000 Gent, Belgium
[3] Universiteit Antwerpen, Dept. of Management Information Systems, Prinsstraat 13, 2000 Antwerpen, Belgium
[4] Hogeschool Gent, Dept. of Management and Informatics, Kortrijksesteenweg 14, 9000 Gent, Belgium

Abstract. Today, in state of the art process engine architectures, process models are executed by a central orchestrator (i.e. one per process). There are however a lot of drawbacks in using a central orchestrator, including a single point of failure and performance degradation. Decentralization algorithms that distribute the workload of the central orchestrator exist, but the result still suffers from a tight coupling between the different decentralized orchestrators and therefore has a decreased scalability. In this paper, we show practical transformations to decentralize a process model into autonomous, independent process engines. This solves the fundamental problems of the classical decentralization algorithms, increases the availability of the global process flow and makes it easier to re-specify and redeploy process models.

1 Introduction

In the last couple of years, process modeling received increasing attention from researchers and practitioners. Especially with the arrival of service oriented computing, process modeling became even more important. Starting from atomic services, new aggregate services can be built by combining the atomic services and describing an execution flow between the different entities. This way composite services are created, which can again be used in other compositions. When these compositions are described with a specific executable language (e.g. BPEL4WS [1] or BPMN 2.0 [2]), automated enactment using a process engine can be accomplished. The description of the process flow can be interpreted by a process engine (or orchestrator), which coordinates and triggers the described work.

Typically, the execution of each composite service (or process) is coordinated by one central entity (Fig. 1a, coordinator C0). This central orchestrator is initiated upon a request from a client and starts the execution of the workflow logic described in the composite service (Fig. 1a, tasks T1, T2 and T3). This is called CENTRALIZED ORCHESTRATION [3].

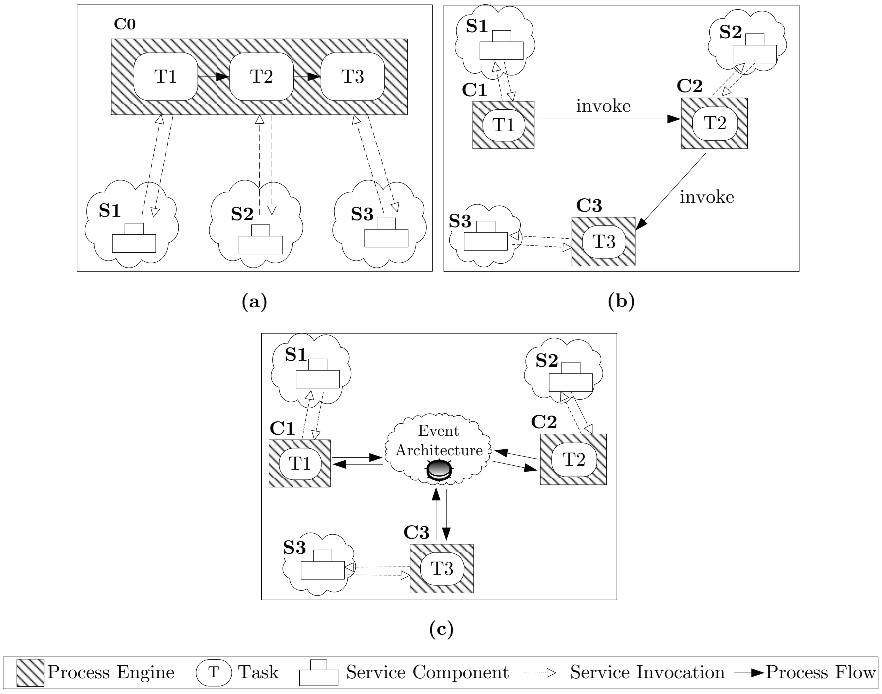


Fig. 1. Centralized, Decentralized and Event-Based Orchestration

The use of a central orchestrator per process struggles, however, with major problems in today's highly decentralized world. Using a central orchestrator (or execution engine) for a composite service creates: (a) *a single point of failure*, the services (work items) are distributed and decentralized (Fig. 1a: S1, S2 and S3), but the decision logic and coordination of the workflow is still located at one point (Fig. 1a: C0). Failure of the coordinator means failure of the entire process, even if the services are still available; (b) *unnecessary network traffic*, all (data) traffic from- and to- services invoked by the orchestrator runs through this central orchestrator, even if the data is of no importance to the orchestrator itself (e.g. data from S1 to S2 in Fig. 1a); and (c) *a performance bottleneck*, the number of process instances can run up very quickly and if all are coordinated at one point in the IT infrastructure, performance decreases significantly [4,5,6,7].

To overcome this bottleneck, solutions are given to decentralize the coordination work [4,5,6]. This results in separated process engines, which remove the need for a central orchestrator and decentralize the workflow logic (Fig. 1b, process engines C1, C2 and C3). This is DECENTRALIZED ORCHESTRATION.

Simple decentralization of the process flow fixes the fundamental problems of centralized orchestration (single point of failure and performance degradation), but not to a full extend [7]. Execution engines are still mutually tightly coupled in the process enactment infrastructure. For example, the start of execution engine C2 in Fig. 1b relies on its invocation by execution engine C1. C2 isn't autonomous and has to rely on decisions made by C1 (i.e. its request to start

C2). The logic of the next step in the process is located with the caller (C1), and not with the callee (C2). This tight coupling creates inflexible IT infrastructures and decreases scalability of the process architecture [8].

To solve the tight coupling, we proposed an extension to decentralized orchestration, which uses an event-based architecture as the communication paradigm in decentralized orchestration (see Fig. 1c) [7]. Decentralized Event-Based Orchestration will create autonomous process engines, capable of assessing their environment and deciding on their own when to initiate their execution (which is a useful property in process management [9]). It also creates a highly loose coupled infrastructure, which makes changes to the process flow relatively easy ('plug and play' of process engines). Notice that we've introduced an event driven architecture to support the decentralization of the process flow (full arrowhead arrows in Fig. 1), not for the invocation of services (open arrowhead arrows in Fig. 1), which has already been accomplished by many others (SOA and EDA [10]).

To gain the benefits from decentralized event-based orchestration, fundamental transformations of the modeled process flow are necessary. In this paper we'll introduce the practical transformations involved in transforming a standard (global) process to a decentralized event-based orchestrated process. The outcome of this transformation is a process model that can be executed by several event-based process engines (or orchestrators). Each orchestrator will be autonomous and distributed, which increases scalability and removes the single point of failure.

In the next section we briefly explain decentralized event-based orchestration, followed by the positioning of the transformations in process development and enactment (Sect. 3). In Sect. 4 we show the actual transformations involved in transforming a process model to a decentralized event-based model. In Sect. 5 we end the paper with a conclusion and some implications of this research.

2 Decentralized Event-Based Orchestration

Decentralized event-based orchestration is the coordination of a single process flow by multiple, autonomous orchestrators [7]. Each orchestrator coordinates a little piece of the global, entire process flow. Thus, combined, they coordinate the global process as modeled by the process modeler. Communication between the orchestrators is accomplished by means of an event based architecture. An event based architecture is a communication paradigm that uses a publish/subscribe interaction scheme. An event is something that happens, and when an event occurs, a notification of this event occurrence is published in the architecture. The architecture then routes this notification to interested parties (the subscribers).

Using a publish/subscribe interaction scheme accomplishes loose coupling between two communicating entities. These include: space decoupling (unawareness of interaction partners), time decoupling (interaction partners don't need to be active at the same time) and synchronization decoupling (asynchronous send and receive) [11,7]. Using an event based architecture for decentralized orchestration thus removes the tight coupling between the different distributed execution engines, which makes the process architecture more scalable. New process engines which consume already published events, can simply be added to the process architecture without making any changes to the already running infrastructure. Note that the supporting entities in an event based architecture (the *cloud* in Fig. 1c) are also loosely coupled and don't add another single point of failure. Many solutions exist that distribute the event based architecture itself [8].

A second consequence of using an event driven architecture in a process decentralization setting is that each execution engine becomes autonomous. A decentralized orchestrator can asses its environment, and when the environment is in a certain state (i.e. some specific events happened), it starts its execution. An orchestrator doesn't rely anymore on messages that request its initiation, in stead it decides for itself when to start. After execution, the orchestrator publishes a notification of its occurrence. This event alters the environment, whereupon other orchestrators *may* react and execute their process logic. A chain of these event publications and consummations (assessment of the environment) results in the execution of the process flow as modeled by the process modeler.

3 Deploying a Process Specification

Transforming a process model to a decentralized event-based model happens at deployment time. This enables the process modeler to not take decentralization into account when designing the process model. Figure 2 shows a process specification-deployment structure. First a process modeler designs a global, fully specified, executable process model. This model not only specifies the flow, but also specifies which service(s) will handle which tasks defined in the process flow (service invocations, see top part of Fig. 3). After process specification, the process can be deployed. It is at this time that the specified process will undergo a transformation which decomposes the process into smaller parts. Note again that our decentralization focuses on transforming the process logic, not the invocation of services.

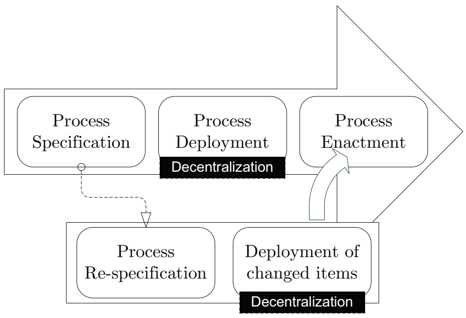


Fig. 2. Deployment and re-specification of a process model in an event-based orchestration

To decentralize the process flow, a unit of decomposition has to be chosen. The unit of decomposition can be anything from a *task* to a group of process elements (*tasks*, *gateways*, . . .). Each unit of decomposition will be deployed to a separate execution engine, resulting in a one-to-one mapping between unit of decomposition and coordinator. This is illustrated in Fig. 3, with the unit of decomposition a *task*. Each task in the original process flow (T1, T2, T3 and T4) is to be deployed on its own process engine (C1, C2, C3 and C4).

After deployment, re-specification and redeployment of the process model can be done with little effort. The process modeler can re-specify the global process model, after which only the changed items in the process flow will need to be redeployed (see Fig. 2). The already existing not-changed items can be left running without interruption. This is possible due to the space decoupling property of event-based orchestration [7].

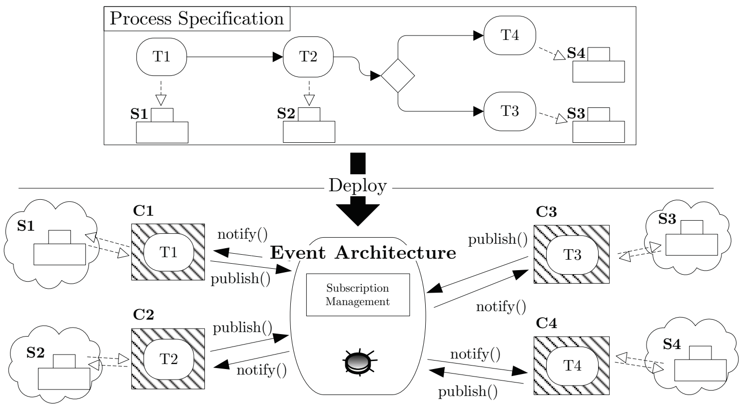


Fig. 3. Deploying a process specification to a decentralized event-based orchestration

4 Transformation

To illustrate the transformations involved in deploying a process specification to a decentralized event-based orchestration, we use BPMN 2.0 [2] as the notation in which the process is specified. Besides a workflow notation, BPMN 2.0 has a token based execution semantics. This makes it possible to directly execute process models defined in BPMN 2.0. Process engine solutions like jBPM [12] and Activiti [13] already support this feature. Because BPMN 2.0 defines a vast amount of entities that can occur in the process model, we define the scope for our transformations to the Standard Process Models as defined by [14]. A Standard Process Model embodies *process elements* which are connected to each other by *transitions*. A process element is either an activity, an AND-Split, a XOR-Split, an OR-Split, an AND-Join and a XOR-Join. These elements correspond to the basic control flow patterns, together with the multi-choice control flow pattern [15].

The outcome of our decentralization is also compliant with the BPMN 2.0 metamodel. This way, any BPMN 2.0 process engine can eventually run the decentralized process model (if it supports the used concepts and a publish/subscribe communication architecture).

4.1 Translating the Unit of Decomposition

We choose a *task* as the unit of decomposition in our transformation. This guarantees a fine grained decentralization. Each task gets translated to a separate process containing that same original task, (multiple) start event(s) and one end event (see Fig. 4 for an example). The unique end event indicates the notification of the tasks accomplishment. This end event is transcribed in BPMN 2.0 as a *throw signal event*. Signal events indicate events that are not process bound, multiple processes can have start events that are triggered from the same broadcasted signal. The semantics of a signal-event resemble closely an event-notification of an event-based architecture. They are thus the most appropriate notation to symbolize our loosely coupled event structure.

As start for the new process, event rules need to be calculated. An event rule is a rule stating in which situation this new process can start. For example, the start rule `(EventA AND EventB) XOR (EventC)`, simply says that the process starts after the occurrence of event A and event B or after the occurrence of event C. For each task, this rule is deduced from the original global process model (the input for the transformation). Event rules are transcribed by using a *catch signal event*. A conjunction in the event rule is indicated by displaying the start event with a *parallel multiple* marker. Disjunctions in the event rule are denoted by using multiple start events. This notation ensures that an event rule is always expressed in a Disjunctive Normal Form (DNF). Figure 4 shows an example of the resulting new process for one unit of decomposition (task).

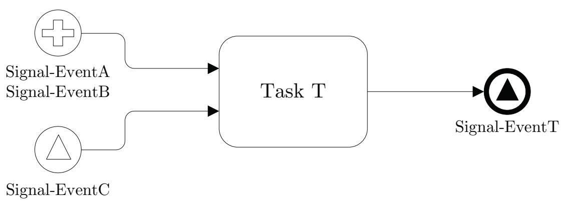


Fig. 4. Result of a single task after the decomposition of a process flow

4.2 Event Rules

To find the start event rule of a task, the preceding elements in the process flow have to be investigated:

Sequence Flow. The most basic start rule for a task is that it can only start after the successful completion of the preceding task in the sequence flow. Figure 5a shows the corresponding transformation. Task X starts after the completion of task A. Our unit of decomposition is a task, thus it is put in a separate process, with as start event a catch of the signal indicating the end of task A.

Exclusive Gateway. If the incoming flow of a task is connected with an exclusive gateway, the start of the task is dependent on the successful execution of one of the tasks preceding the exclusive gateway. The event rule for a task connected with an exclusive gateway is thus a disjunction of the signals indicating the completion of the process elements preceding the exclusive gateway. Figure 5b shows this transformation.

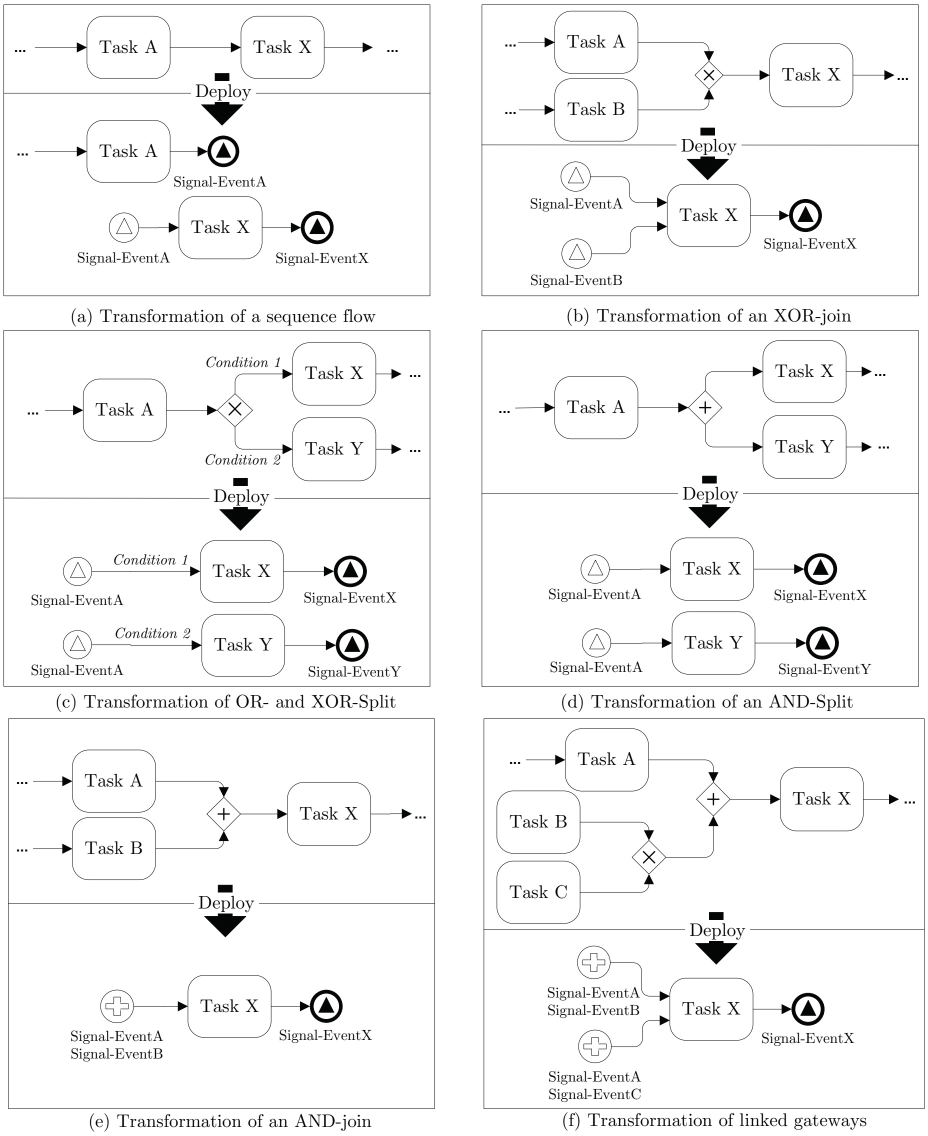
Conditions on preceding sequence flows. BPMN 2.0 states that the conditions belonging to an OR- and XOR-split are put on the sequence flows succeeding the OR- and XOR- gateway (*conditional flow*). If the condition is valid, that specific path in the process is 'chosen'. These conditions should be conveyed to the newly created decentralized process and displayed on the correct sequence flow (see Fig. 5c). This means that the (decentralized) task will only start when the environment is in a certain state (i.e. some events happened) and when the condition on the respective sequence flow is true. It is possible that, when multiple exclusive gateways are linked together, multiple conditions should be true before the task can start. All these conditions are put in conjunction on the respective sequence flow.

Parallel Gateway. Figure 5d and 5e illustrate the transformation for a task with its incoming sequence flow connected to a parallel gateway (either AND-split or AND-join). The event rule becomes a conjunction of the completion-notifications of the process elements preceding the parallel gateway. Observe that for an AND-split only *one* signal event (`Signal-EventA`) is used to trigger the multiple catch events (for tasks X and Y). This reduces the number of different event messages that need to be exchanged in the decentralized orchestration, compared to creating a separate signal event for each flow outgoing the AND-gateway.

Linked Gateways. Gateways can also be directly linked together by sequence flows (see Fig. 5d). In this situation the event rule has to be calculated recursively according to the transformations described above. The eventual rule is put in DNF so that it can be represented in the BPMN schema (see Sect. 4.1). Figure 5d shows an example. Following the transformations stated above results in an event rule for task X: $TaskA \wedge (TaskB \vee TaskC)$ and in DNF: $(TaskA \wedge TaskB) \vee (TaskA \wedge TaskC)$.

4.3 Formal Implementation

We have implemented the transformation in the Atlas Transformation Language (ATL) [16]. ATL is available as a plugin in the Eclipse Modeling Framework and provides a way to declaratively describe the transformation of a source model (supported by a metamodel) to a target model. Figure 6a shows the transformation structure. As input, the transformation takes a source model which conforms to the BPMN 2.0 metamodel (any BPMN Diagram Interchange file [2]). The output of the transformation is also a model conforming to the BPMN 2.0 metamodel. The output file (a BPMN Diagram Interchange file) can be directly uploaded in any process engine supporting BPMN 2.0. If the

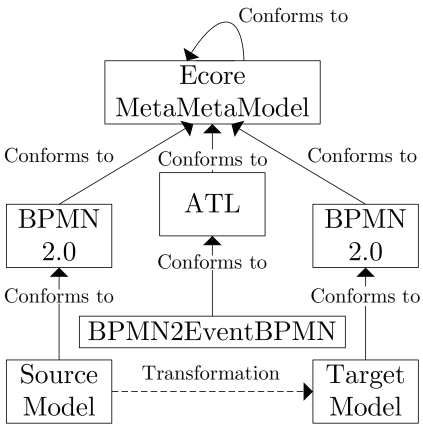


(a) Transformation of a sequence flow
(b) Transformation of an XOR-join
(c) Transformation of OR- and XOR-Split
(d) Transformation of an AND-Split
(e) Transformation of an AND-join
(f) Transformation of linked gateways

Fig. 5. Deploying Standard Process Flow elements to a decentralized event-based orchestration

engine implements signal event communication in a publish/subscribe fashion, the benefits described in [7] will become available.

A small excerpt of the ATL transformation code is found in Fig. 6b. The transformation contains only one matching rule[1] which translates a task in the source model to a new resulting process as described in Sect. 4.1.



(a) ATL transformation structure

```
rule Task2Process {
 from
  s: BPMN20!Task
 to
  newTask: BPMN20!Task (
   id <- s.id,
   name <- s.name
  ),
  endev: BPMN20!EndEvent (
   id <- 'EndEvent' + s.id,
   eventDefinitions <- outputDef
  ),
  outputDef: BPMN20!SignalEventDefinition (
   id <- 'outputDef' + s.id,
   signalRef <- outputSignal
  ),
  outputSignal: BPMN20!Signal (
   id <- 'Signal' + s.id),
  ),
  process: BPMN20!Process (
   id <- 'sid-Process' + s.id,
   flowElements <- Sequence{newTask},
   flowElements <- Sequence{outputSeqFlow},
   flowElements <- Sequence{endev},
   flowElements <- s.getEventRule()
    -> collect(e | thisModule.createStartEvents(e))
  )
```

(b) Excerpt of the ATL BPMN2EventBPMN transformation code

Fig. 6. ATL Transformation

5 Conclusion, Implications and Future Research

In this paper we introduced the practical transformations necessary to transform a standard process model to a decentralized event-based orchestration.

Using an event based communication paradigm in a decentralized orchestration creates highly flexible, autonomous entities, which increase scalability and availability of the process flow. By doing the transformations at deployment, the process modeler doesn't need to know the decentralization details. Deployment of a changed process flow can also be done fairly quickly, without the need to interrupt the current (unchanged) process entities.

Another implication of working with an event based architecture in process enactment is that event-logs of the running processes become readily available.

[1] A matching rule is a rule that matches an entity from the source model to new entities in the target model

This enables easier access to process mining [17] or complex event processing [18]. Yet another application can be agent based development, where the operations of the agents are noted in a process flow-like style.

Future research includes widening the scope of the transformable process elements to a level 2 process modeling [19], which includes subprocesses, intermediate events, transactions, ... as well as including data management. We also intend to prove the correctness of the transformation rules with process algebra and formally validate the added value by testing on availability (stress testing) and scalability of the decentralized event-based process flow.

References

1. Oasis: Web service business process execution language version 2.0. Oasis Standard
2. Object Management Group: Draft proposal for: Bpmn 2.0, beta 2. http://www.omg.org/cgi-bin/doc?dtc/10-06-04 (May 2010)
3. Barros, A., Dumas, M., Oaks, P.: Standards for web service choreography and orchestration: Status and perspectives. BPM Workshops 61–74
4. Chafle, G., Chandra, S., Mann, V., Nanda, M.: Decentralized orchestration of composite web services. Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (2004) 134–143
5. Muth, P., Wodtke, D., Weissenfels, J., Dittrich, A., Weikum, G.: From centralized workflow specification to distributed workflow execution. Journal of Intelligent Information Systems **10**(2) (1998) 159–184
6. Nanda, M., Chandra, S., Sarkar, V.: Decentralizing execution of composite web services. ACM SIGPLAN Notices **39**(10) (2004) 170–187
7. Hens, P., Snoeck, M., De Backer, M., Poels, G.: Decentralized Event-Based Orchestration. In: Third International Workshop on Event-Driven Business Process Management. (2010)
8. Mühl, G., Fiege, L., Pietzuch, P.: Distributed Event-Based Systems. Springer-Verlag New York, Inc. Secaucus, NJ, USA (2006)
9. Jennings, N., Norman, T., Faratin, P., OBrien, P., Odgers, B.: Autonomous agents for business process management. Applied Artificial Intelligence **14**(2) (2000)
10. Michelson, B.: Event-driven architecture overview. OMG report (2006)
11. Eugster, P., Felber, P., Guerraoui, R., Kermarrec, A.: The many faces of publish/subscribe. ACM Computing Surveys (CSUR) **35**(2) (2003) 131
12. jBPM. http://jboss.org/jbpm
13. Activiti. http://www.activiti.org/
14. Kiepuszewski, B., ter Hofstede, A., van der Aalst, W.: Fundamentals of control flow in workflows. Acta Informatica **39**(3) (2003) 143–209
15. van Der Aalst, W., Ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distributed and parallel databases **14**(1) (2003) 5–51
16. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P.: ATL: a QVT-like transformation language. In: the 21st ACM SIGPLAN symposium. (2006)
17. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: A survey of issues and approaches. Data & Knowledge Engineering **47**(2) (2003) 237–267
18. Luckman, D.: The power of events: an introduction to Complex Event Processing. Addison-Wesley (2002)
19. Silver, B.: BPMN Method and Style. Cody-Cassidy Press (2009)

Unified Patterns to transform business rules into an event coordination mechanism[1]

Willem De Roover and Jan Vanthienen

Department of Decision Sciences & Information Management,
Katholieke Universiteit Leuven, Belgium
(willem.deroover;jan.vanthienen)@econ.kuleuven.be

Abstract. Business rules define and constrain various aspects of the business, such as vocabulary, behavior and organizational issues. Enforcing the various rules of the business in information systems is not straightforward, because different mechanisms exist for the transformation of business rules into model driven implementations, leading to partial solutions for process management, data constraints, audit constraints, etc. In this paper, we examine if and how business rules, not only data rules, but also process rules, timing rules, authorization rules, etc., can be expressed in SBVR and translated using patterns into a more uniform event mechanism, such that the event handling could provide an integrated enforcement of business rules of many kinds.

The need for a unified framework Business rules should be on the one hand comprehensible so that they can be understood by business people and on the other hand formal so that they can be enforced by information systems. The Semantics of Business Vocabulary and Business Rules (SBVR), a new standard for business modeling within the Object Management Group (OMG), has such property.

SBVR is a language to describe the structure and the meaning of vocabulary and business rules in terms of formalized statements about the meaning. This also makes SBVR a suitable base language for defining process-aware rules, but it does not contain a vocabulary with process related concepts such as agents, activities, process states and events. To this end we extended SBVR with a vocabulary for expressing process-related concepts, called the EM-BrA^{2}CE Vocabulary.

Business Rule Types SBVR extended with the EM-BrA^{2}CE Vocabulary allows us to define three groups of business rules: (1) Data rules constrain particular manipulations of data. (2) Control-flow rules constrain the execution of activities. (3) Organizational aspects constrain the authorization to perform and see particular activities.

Patterns for transforming business rules into event rules Enforcing the different types of rules of the business in information systems is not straightforward, different mechanisms exist for the (semi-)automatic transformation of

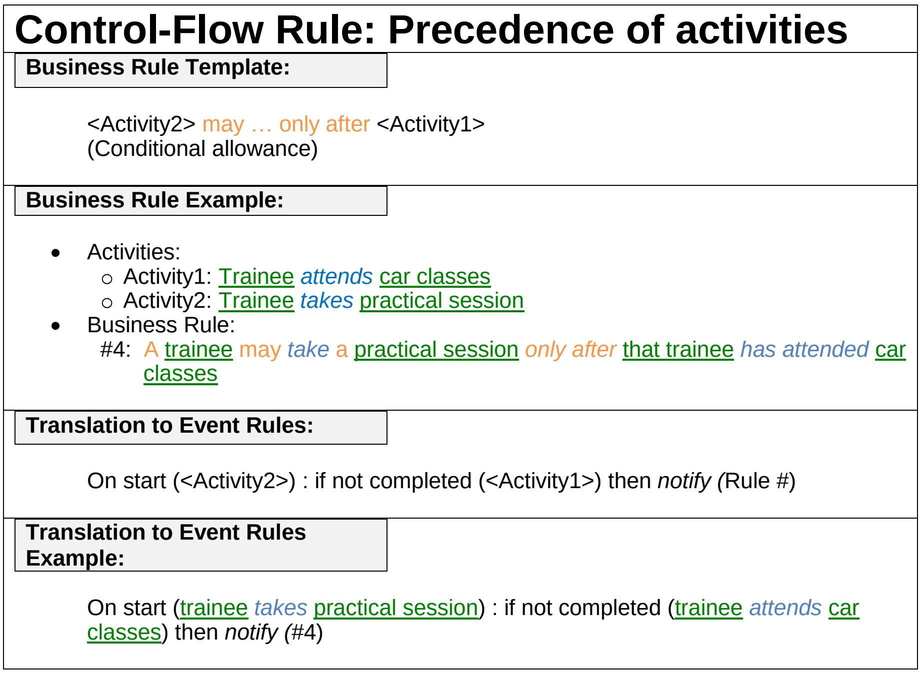


Fig. 1. Control-Flow rule: Precedence of Activities

various business rule types, leading to partial solutions for data constraints, process mangement and audit constraints. Event handling provides a more uniform enforcement of business rules of many kinds, not only data rules, but also control-flow rules and organizational rules.

To this end, we provide a pattern mechanisme to transform SBVR business rules into event-driven enforcement rules and notifications. For each type of rule we define a general template. The rule template generates a set of Event-Condition-Actions rules once a business rule is defined. The Event-Condition-Action rules are equivalent to the SBVR Business rules but have the advantage that they make clear when they have to be checked. Example templates for Data aspects(integrity constraints, derivations rules), Control flow aspects(precedence of activities (see figure 1)) and organizational aspects(Authorization rules) are provided.

Conclusion By transforming the business rules into Event-Condition-Action rules we provide a more uniform event mechanism, such that event handling can provide an integrated enforcement of business rules of many kinds.

References

1. De Roover, W., Vanthienen, J.: Unified patterns to transform business rules into an event coordination mechanism. In: 4th International Workshop on Event-Driven Business Process Management Proceedings. (2010) 61–73

Papers of Interest

Towards Robust Conformance Checking (Extended Abstract)

A. Adriansyah, B.F. van Dongen, and W.M.P. van der Aalst

Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
a.adriansyah,b.f.v.dongen,w.m.p.v.d.aalst@tue.nl

The growing complexity of processes in many organizations stimulates the adoption of business process management (BPM) techniques. Process models typically lie at the basis of these techniques and generally, the assumption is made that the operational business processes as they are taking place in practice conform to these models. However, recent experience has shown that this often isn't the case. Therefore, the problem of checking to what extent the operational process conforms to the process model is increasingly important.

In [1], we provide a robust method for calculating conformance between a log and a process model. First, we introduce flexible models that provide an abstraction of many languages and allow for the modeling of complex control flow constructs, such as OR-split/joins and multiple tasks that represent the same activity. We provide semantics for these models, but without specifying how to execute them. Instead, we show that in the context of a case that has been recorded in the log, we can construct instances of the model that maximize certain conformance metrics. Finally, using experiments on simulated data (comparable in size to real-life data sets), we show that our approach calculates fitness correctly in the presence of complex constructs, where existing approaches do not.

Given a process model and an event log, our approach does not only solve the problem of having inaccurate conformance results, but also opens possibilities to compare conformance values for a given log between different models made in different languages. With existing approaches, conformance comparison between two process models and a given log require both models to be created in the same process modeling language. We soften the requirement such that models can be in different modeling languages as long as they can be represented as abstract models.

The work presented in this paper provides a solid basis for robust conformance checking. Since our flexible models do not have executable semantics, we do not rely on state-space exploration (which is required in Petri-net based conformance checking). Using the approach presented in [1], we plan to define metrics that capture different dimensions of conformance. Furthermore, our approach can easily be extended to identify the "skipping" of activities, i.e. by identifying which tasks were executed but not logged.

Our approach is fully implemented in Version 6 of the ProM framework, that can be obtained from http://www.processmining.org, evaluated using simulated event logs and compared against an existing conformance technique based on Petri nets. By experiments, it is proven that the proposed approach provide better insights to conformance between a given process model and an event log, particularly when advanced control flow constructions (e.g. OR split, OR join) exist in the model.

References

1. A. Adriansyah, B.F. van Dongen, and W.M.P. van der Aalst. Towards Robust Conformance Checking. In *Proceedings of the 6th International Workshop on Business Process Intelligence (BPI 2010 - accepted)*, 2010.

Abstracting Common Business Rules to Petri Nets (Abstract)

Kees van Hee[1], Jan Hidders[2], Geert-Jan Houben[2], Jan Paredaens[3], and Philippe Thiran[4]

[1] Eindhoven University of Technology, The Netherlands
[2] Delft University of Technology, The Netherlands
[3] University of Antwerp, Belgium
[4] University of Namur, Belgium

These days Business Information Systems (BIS) have a highly responsible task: they execute large parts of the *business processes* autonomously. So organizations become more and more dependent on their BIS. Business processes usually require a certain order in which activities have to be executed. On top of that, there are many other *business rules* that should be met by the execution of business processes, e.g., the famous Sarbanes-Oxley rules (cf. [1]) and the General Accepted Accounting Principles (GAAP). Business rules can be required by any stakeholders, such as management, government, shareholders and business partners, both clients and suppliers. In this paper we take it for granted that a BIS has some unknown errors and that the system is for all stakeholders more or less a black box. Instead of a human auditor we propose here an approach where the BIS is extended by an independent *monitor system* that checks the essential business rules on the fly and that reports violations of business rules or that interrupts the BIS to prevent the occurrence of a violation.

As basic terminology we use workflow notions such as *task*, *case*, *agent* and *resource*. This is consistent with the terminology for *accounting information systems* where the REA (Resource, Event and Agent) data-model is used frequently. This model is an Entity-Relationship model for accounting information systems [2–4]. We assume that business processes are described by traces which are here defined as sets of events where events are partial functions that map event properties to property values. Event properties can be either a standard mandatory property such as **time**, **case**, **task**, **agent** or a certain type of resource that is involved in the event such as money or used materials. A property value can be either a task name, an agent, a basic value, or a rational number where the latter is also used to represent time stamps. The values of standard properties are restricted so they have property values of the right type.

The syntax of the BRL, the business rule language we present here, is defined by the following abstract grammar:

$$F ::= \neg F \mid (F \wedge F) \mid \langle P_S = E, \ldots, P_S = E \rangle \mid E = E \mid E < E.$$
$$E ::= \mathcal{X} \mid D_S \mid \Sigma(\mathcal{X} : F)E \mid (E + E).$$

Here F denotes boolean formulas and E expressions that return a property value. The non-terminals P_S and D_S represent event properties and property values, respectively, and $\mathcal{X}$ represents variables that range over property values, which includes time-stamps. The value of the formulas and expressions are defined w.r.t. to a certain trace. There are three special workflow constructs with specific workflow semantics: formulas of the form $e_1 < e_2$, formulas of the form $\langle p_1 = e_1, \ldots, p_k = e_k \rangle$ and expressions of the form $\Sigma(x : \varphi)e$. The first compares the values of e_1 and e_2 according to the ordering defined over the property values, which includes the time ordering over time-stamps. The formula $\langle p_1 = e_1, \ldots, p_k = e_k \rangle$ states that the trace contains at least one event ev such that $ev(p_1) = v_1, \ldots, ev(p_k) = v_k$ where $v_1, \ldots, v_k$ are the values of the expressions $e_1, \ldots, e_k$ for the trace. The expression $\Sigma(x : \varphi)e$ is a sum quantifier which sums the values of expression e for each property value x that occurs in the trace and satisfies the formula φ. For example, $\Sigma(x_t : (x_t = x_t))(\Sigma(x_p : \langle \mathbf{time} = x_t, \mathbf{payed} = x_p \rangle)x_p)$ computes the total amount of money payed.

The sum quantifier allows us to formulate business rules about aggregates of used and produces resources such as "for each case at each moment the total amount of used bread does not exceed the amount of previously delivered bread". In addition it can also be used to formulate more standard workflow rules concerning precedence, such as "an event for task a is always followed by an event for task b". For example, the formula $(\Sigma(x : \langle \mathbf{time} = x, \mathbf{task} = a \rangle)1) > 0$ checks if at least one event for task a has occurred.

In the full paper it is shown how for certain types of rules in BRL we can define a monitor system that is able to check them on the fly and in parallel to a business information system. This is done by translating parts of the evaluation of business rules into the execution of a Petri net [5] by using *history-dependent Petri nets* [6]. This is an efficient way of checking rules because it can be done in an incremental way, i.e., event by event, using a Petri net engine. If the Petri net cannot execute a transition, then a rule is violated and this can be reported, or it may generate an interrupt for the BIS and trigger an exception handler. In the future we will also try to transform larger classes of business rules.

References

1. Berg, D.: Turning Sarbanes-Oxley projects into strategic business processes. Sarbanes-Oxley Compliance Journal (2004)
2. Romney, M.B., Steinbart, P.J.: Accounting Information Systems. 11 edn. Pearson International Editions (2009)
3. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. International Journal of Accounting Information Systems **3**(1) (2002) 1–16
4. McCarthy, W.E.: The REA accounting model: A generalized framework for accounting systems in a shared data environment. The Accounting Review **57**(3) (1982) 554–578
5. Reisig, W.: Petri Nets: An Introduction. Volume 4 of Monographs in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin (1985)
6. van Hee, K., Serebrenik, A., Sidorova, N.: Token History Petri Nets. Fundam. Inf. **85**(1-4) (2008) 219–234