Patrick van Bommel
Stijn Hoppenbrouwers
Sietse Overbeek
Erik Proper
Joseph Barjis (Eds.)

LNBIP 68

The Practice of Enterprise Modeling

Third IFIP WG 8.1 Working Conference, PoEM 2010
Delft, The Netherlands, November 2010
Proceedings

ifip

Springer

Lecture Notes in Business Information Processing 68

Series Editors

Patrick van Bommel Stijn Hoppenbrouwers
Sietse Overbeek Erik Proper
Joseph Barjis (Eds.)

The Practice of Enterprise Modeling

Third IFIP WG 8.1 Working Conference, PoEM 2010
Delft, The Netherlands, November 9-10, 2010
Proceedings

Springer

Volume Editors

Patrick van Bommel
Stijn Hoppenbrouwers
Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
E-mail: {pvb,stijnh}@cs.ru.nl

Sietse Overbeek
Delft University of Technology
Faculty of Technology, Policy and Management
Section of Information and Communication Technology
P.O. Box 5015, 2600 GA Delft, The Netherlands
E-mail: s.j.overbeek@tudelft.nl

Erik Proper
Public Research Centre Henri Tudor
29, avenue John F. Kennedy, 1855 Luxembourg-Kirchberg, Luxembourg
E-mail: erik.proper@tudor.lu

Joseph Barjis
Delft University of Technology
Faculty of Technology, Policy and Management
Section of Systems Engineering
P.O. Box 5015, 2600 GA Delft, The Netherlands
E-mail: j.barjis@tudelft.nl

Library of Congress Control Number: 2010937669

ACM Computing Classification (1998): J.1, H.3.5, H.4.1

ISSN	1865-1348
ISBN-10	3-642-16781-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-16781-2 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180 5 4 3 2 1 0

Preface

These are the proceedings of the Third IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling, held in Delft (The Netherlands) on November 9 and 10, 2010. It followed the success of PoEM 2008 and 2009 (both held in Stockholm), which each attracted over 50 participants from all over the world, representing both industry and academia. This indicates that enterprise modeling (EM) has gained popularity both in the academic community and among practitioners. The interactive format of the previous conferences sparked constructive interaction between research and practice. PoEM 2010 further strengthened this interaction.

The PoEM conferences contribute to establishing a dedicated forum where the use of EM in practice is addressed by bringing together researchers, users and practitioners. The main focus of PoEM is EM methods, approaches, and tools, and how they are used in practice. The goal of the conference was to further a better understanding of the practice of EM and improve the theory behind the practice, contributing to improved EM practice and to the sharing of knowledge.

For this third edition, the founders of PoEM, Anne Persson and Janis Stirna, passed the torch for the first time; we hope we lived up to the high standards set by them and thank them for their initiative, commitment, and excellent work. PoEM will return to Scandinavia next year, and will remain to do so every other year.

The 17 high-quality papers (out of 44 submissions) presented at PoEM 2010 display a welcome diversity in topics while being duly centered around the enterprise modeling theme. A number of submissions reflected the trend for both practitioners and academics to look into domains and conceptualizations that are more and more distant from those that are the focus in traditional information systems engineering, addressing a continuation of the move towards dedicated and far reaching "business-orientation." Also, we observe that the field is slowly but surely maturing, as indicated by an increase in detail and specialization of the contributions.

In its 2010 edition, PoEM saw relatively few submissions concerning enterprise architecture. It is very likely this is because this year PoEM was co-located with two other events: the Practice-Driven Research in Enterprise Transformations (PRET 2010) working conference, and the Trends in Enterprise Architecture Research (TEAR 2010) workshop. Proceedings of both events also appear in the Springer LNBIP series. The three events together constituted Enterprise Engineering Week, with Erik Proper at the helm.

We would like to extend warm thanks to everyone involved in making PoEM 2010 happen: the members of the PC, the people who submitted papers, keynote speakers Etienne Rouwette (Radboud University Business School) and Jeroen van Grondelle (Be Informed), and everyone involved in the organization.

September 2010

Patrick van Bommel
Stijn Hoppenbrouwers
Sietse Overbeek
Erik Proper
Joseph Barjis

Organization

POEM 2010 was organized by the Radboud University Nijmegen and the Delft University of Technology.

Executive Committee

Conference Chair	Stijn Hoppenbrouwers, Radboud University Nijmegen
PC Chair	Patrick van Bommel, Radboud University Nijmegen
PC Co-chair	Sietse Overbeek, Delft University of Technology
PC Co-chair	Erik Proper, Radboud University Nijmegen

Program Committee

Marko Bajec	University of Ljubljana, Slovenia
Rimantas Butleris	Kaunas University of Technology, Lithuania
Wolfgang Deiters	Fraunhofer ISST, Germany
Eric Dubois	CRP Henri Tudor, Luxembourg
Mathias Ekstedt	KTH, Sweden
Gustaf Ericsson	Alfa Laval AB, Sweden
Xavier Franch	Universitat de Catalunya, Spain
Remigijus Gustas	Karlstad University, Sweden
Terry Halpin	LogicBlox, Australia
Jarl Hoglund	Allmentor AB, Sweden
Stijn Hoppenbrouwers	Radboud University Nijmegen, The Netherlands
Pontus Johnsson	KTH, Sweden
Havard Jorgensen	Commitment AS, Norway
John Krogstie	Norwegian University of Science and Technology, Norway
Michel Léonard	University of Geneva, Switzerland
Peri Loucopoulos	Loughborough University, UK
Raimundas Matulevicius	University of Tartu, Estonia
Graham McLeod	PROMIS Solutions AG, South Africa
Daniel Moody	University of Twente, The Netherlands
Christer Nellborn	Nellborn Management Consulting AB, Sweden
Bjorn Nilsson	Anates AB, Luxembourg
Andreas Opdahl	University of Bergen, Norway
Sietse Overbeek	Delft University of Technology, The Netherlands
Oscar Pastor	Valencia University of Technology, Spain
Anne Persson	University of Skövde, Sweden

Naveen Prakash	GCET, India
Erik Proper	Radboud University Nijmegen, The Netherlands
Jolita Ralyté	University of Geneva, Switzerland
Peter Rittgen	Vlerick Leuven Gent Management School, Belgium
Colette Rolland	University of Paris 1 Pantheon Sorbonne, France
Kurt Sandkuhl	Jonkoping Technical University, Sweden
Ulf Seigerroth	Jonkoping International Business School, Sweden
Keng Siau	University of Nebraska-Lincoln, USA
Pnina Soffer	University of Haifa, Israel
Maarten Steen	Novay, The Netherlands
Janis Stirna	University of Stockholm, Sweden
Renate Strazdina	Ernst and Young SIA, Latvia
Patrick van Bommel	Radboud University Nijmegen, The Netherlands
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania
Eric Yu	University of Toronto, Canada

Table of Contents

Comparing Two Techniques for Intrusion Visualization

Vikash Katta[1,3], Peter Karpati[1], Andreas L. Opdahl[2],
Christian Raspotnig[2,3], and Guttorm Sindre[1]

[1] Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
{kpeter,guttors}@idi.ntnu.no
[2] University of Bergen, NO-5020 Bergen, Norway
andreas.opdahl@uib.no
[3] Institute for Energy Technology, NO-1751 Halden, Norway
{vikashk,christir}@hrp.no

Abstract. Various techniques have been proposed to model attacks on systems. In order to understand such attacks and thereby propose efficient mitigations, the sequence of steps in the attack should be analysed thoroughly. However, there is a lack of techniques to represent intrusion scenarios across a system architecture. This paper proposes a new technique called misuse sequence diagrams (MUSD). MUSD represents the sequence of attacker interactions with system components and how they were misused over time by exploiting their vulnerabilities. The paper investigates MUSD in a controlled experiment with 42 students, comparing it with a similar technique called misuse case maps (MUCM). The results suggest that the two mostly perform equally well and they are complementary regarding architectural issues and temporal sequences of actions though MUSD was perceived more favourably.

Keywords: requirements engineering, security, experiment, threat modeling.

1 Introduction

The increased web presence of modern enterprises due to e-commuting, e-commerce, and distributed, inter-organizational workflows have also created new opportunities for computer crime, thus accentuating the need to focus on enterprise security [1]. Security must be in focus on many levels, from high level strategy and managerial policies and down to the competence and awareness of each single employee and correct implementation and operation of each ICT application. Whether talking about the construction of new information systems or the daily operation of existing ones, a vital precondition for improving their security is the ability to learn from previous failures. One possibility is to look at textual descriptions of successful attacks, like [2], but there may be several advantages in combining this with more generic visual models of attacks, both related to understandability, knowledge reuse, and integration with model-based systems engineering. Many techniques have been proposed to capture threat and attack-oriented information, for example attack trees [3], misuse cases [4] and CORAS [1], and more recently misuse case maps (MUCM) [5], focusing on modelling complex intrusions. This technique highlights the relation between security

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 1–15, 2010.

and system architecture, and thereby provides integrated overviews of user-oriented security threats, mitigations and architecture. Even though MUCM has its strengths to show intrusion across the system architecture, it might be confusing to follow the sequence of steps of the intrusion. Evaluations of MUCM [5,6] have suggested a need for better visualization of the sequence of attack steps. To the authors' knowledge, there is a lack of techniques to visualize such sequences.

This paper proposes a new threat modelling technique called misuse sequence diagrams (MUSD), aiming to give an overview on the sequence of the attacker's actions during an intrusion for different stakeholder groups. The technique is based on UML sequence diagrams [7] and misuse cases [2], and utilizes security concepts like vulnerability exploitation and mitigation. Since MUSD was meant to improve on some shortcomings of MUCM, we found it interesting to evaluate experimentally whether MUSD really provided improvement over MUCM related to these issues. Hence we performed a controlled experiment with 42 students to evaluate the participants' performance with the two techniques, as well as their opinions about the techniques.

The two techniques are relevant to enterprise-modelling practice because service-orientation has made information systems increasingly distributed across internal and external organization boundaries. Hence, software architecture has moved from being a technological concern inside "black-box" information systems, to being a "white-box" concern on the boundary between organization and technology. The techniques we present are, to the best of our knowledge, the first attempt to conceptualize this boundary and support it with useful modelling notations. We hope MUCMs and MUSDs can contribute to understanding distributed service-oriented information systems and their architectures from an enterprise and organizational context.

The rest of the paper is structured as follows. Section 2 discusses background and related work. Then, section 3 presents misuse sequence diagrams. Sections 4 and 5 present the research method and the experiment results, respectively. Section 6 discusses the findings, and section 7 concludes the paper.

2 Background and Related Work

According to the definitions of RFC 2828 [8], a *vulnerability* is a weakness in a system's design, implementation, or operation/management that can be exploited to violate its security policy. A *threat* is a potential for violation of security, depending on circumstance, capability, and an action / event that could cause harm. A *countermeasure/mitigation* is something that reduces a threat or attack by eliminating or preventing it, minimizing the harm caused, or by reporting it to enable corrective action.

Misuse cases (MUC) [4,9] are used for threat modelling and security requirements elicitation. While use cases (UC) present the required behaviour of a system, MUC capture undesired behaviour, extending UC with new elements like misuser, misuse case and mitigation use case, as well as new relations like threaten and mitigate. MUC allow an early focus on security in the development process and facilitate discussion among a wide group of stakeholders.

Misuse case maps (MUCM) [5] is a recently proposed technique combining MUC and use case maps (UCM) [10] for an integrated view of security issues and system architecture. MUCM can be used to visualize the trace of an intrusion on the

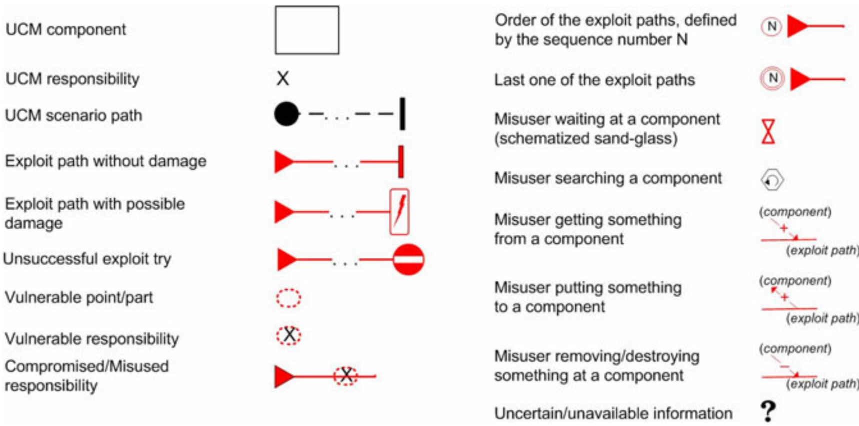

Fig. 1. MUCM notation and its interpretation

architecture of the system, visually based on the UCM notation extended by vulnerabilities, exploit paths and mitigations. An experiment performed to evaluate MUCM showed that it significantly improved the understanding of intrusions and identification of mitigations when compared to MUC combined with a system architecture outline [5].

Fig. 1 shows the MUCM notation. It extends UCM's basic notation with *intrusions*, represented by *exploit paths* that cut through *vulnerable parts* of the system. Each path starts with a triangle and different symbols at the end indicate the outcome. The number on the path indicates its order in a bigger sequence of paths. The system may have vulnerable points or parts which are susceptible to threats, materialized in the attack if the exploit path crosses the vulnerability.

Fig. 2 shows a partial example about a technical entry into the computer system of a company for penetration test purposes [2]. The tester first made 3 unsuccessful attack attempts against the Apache server and firewall. The fourth attempt utilised an undocumented Solaris feature (portmapper - rpcbind - bound to port 32770) to get the dynamic port of the mount daemon (mountd) from the portmapper and direct an NFS request to it, thus succeeding to remotely mount and download the target file system.

As for other related methods, [11] proposes a framework for object-oriented security requirements analysis based on UC, MUC and security use cases for the elicitation and analysis of security requirements in embedded systems. In addition to the framework, misuse sequence diagrams have been introduced to *better* explain a single misuse case scenario. [12] proposes an aspect-oriented methodology for designing secure applications. The methodology uses sequence diagrams for three purposes: describing functionality (primary model), describing attacks on the functionality (misuse model), and describing the incorporation of security mechanisms (security-treated model). The MUSD technique proposed here is similar to those of [11] and [12]. However, they neither visualize complex multi-stage intrusion scenarios nor how vulnerabilities of system components are exploited and mitigated.

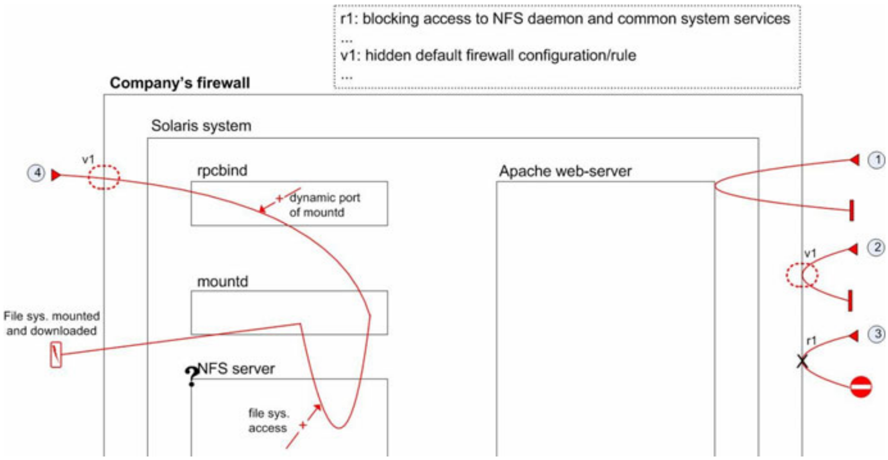

Fig. 2. MUCM for the (part of) penetration test example from the experiment

CORAS [1] is a method for security analysis using system descriptions based on UML diagrams as an input to traditional risk analysis techniques, such as HazOp [13] and Fault Tree Analysis (FTA) [14]. While CORAS offers a specialized set of diagrams for security risk analysis, only using UML diagrams as an input, the MUSD is solely based on MUC and SD. MUSD is not a method for security risk analysis, just a technique for visualizing complex intrusion scenarios.

It could be tempting to compare MUSD to STAIRS [15], which is a method with focal point on refinement of interactions in UML. STAIRS allows among others for the specification of behaviour that shall not be allowed in an implementation, but does not refer to the security or risk domain. However, MUSD presents a specialized notion for security and intrusion visualization, while STAIRS' focus is on a general refinement of UML based specifications thus a comparison does not seem relevant.

UMLSec is an UML extension for secure systems development which can be used "to evaluate UML specifications for vulnerabilities using a formal semantics of a simplified fragment of UML" [16]. Its focus is on formal verification of specifications (e.g. for a protocol) against different adversary types. Although sequence diagrams are included in UMLSec, they rely on other diagram types specifying the broader context. Furthermore, UMLSec applies heavyweight methods which need specific trainings. MUSD aims to facilitate discussion of different stakeholder groups allowing competency transfer and trade-off considerations early in the system development.

3 MUSD

Misuse sequence diagrams (MUSD) combine misuse cases (MUC) and sequence diagrams (SD), to depict and analyze complex intrusion scenarios. MUSD show involved objects, their vulnerabilities and how these were misused. The notation extends UML sequence diagrams as shown in Fig. 3. Just like by the MUC notation, regular and misuse symbols can be combined in the same MUSD diagram. Attack-related

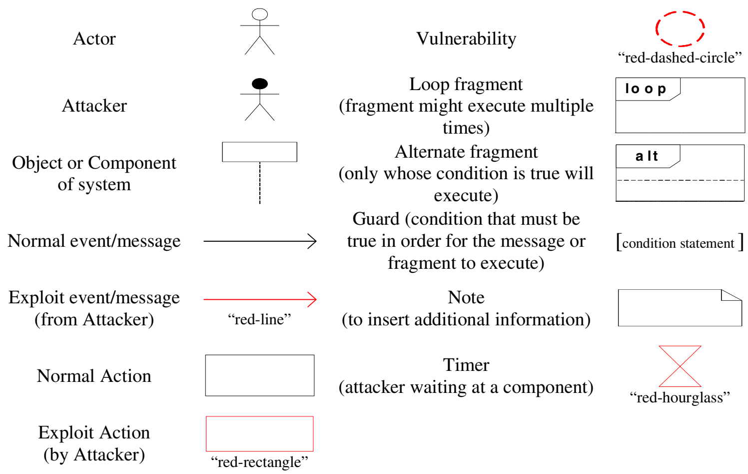

Fig. 3. MUSD basic notation and its interpretation

symbols are shown in red color since inversion (a la misuse cases) does not work for arrows. Exploit messages are messages originating from the attacker with the intention of harming the system. Intrusions are represented by one or more exploit messages using vulnerabilities of the objects in the system. Action symbols are used to represent parts of the intrusion scenario which are unclear or not detailed enough. Exploit actions are performed by the attacker with the intention of attacking the system. Objects which are part of an action will have their lifelines covered by the rectangle denoting the action.

The sequence of exploit messages and actions shows the steps taken by the attacker. These steps are mostly causally related; each building on the result of the previous steps. Notes might appear for explanations. Fig. 4 presents a part of the MUSD depicting the same penetration test case as shown with a MUCM in Fig. 2.

4 Research Method

The purpose of the experiment was to evaluate whether MUSD would be better than MUCM for conveying attack sequences. On the other hand, a gain in this respect could result in other weaknesses instead, especially it would be natural to suspect that MUSD would be poorer than MUCM in conveying the relationship between attacks and system architecture. Hence, the experiment compared the understanding of case descriptions resulting from usage of two notations, both for attack sequence and architectural aspects. Understanding may be a goal in its own right in enterprise systems development, but more often it will be the basis for various problem solving activities. Hence the subjects performed one task measuring understanding by a set of True/False questions, and another addressing problem solving in terms of identifying threats and possible mitigations in the given cases. In the following the experiment design, variables and hypotheses are explained in more detail.

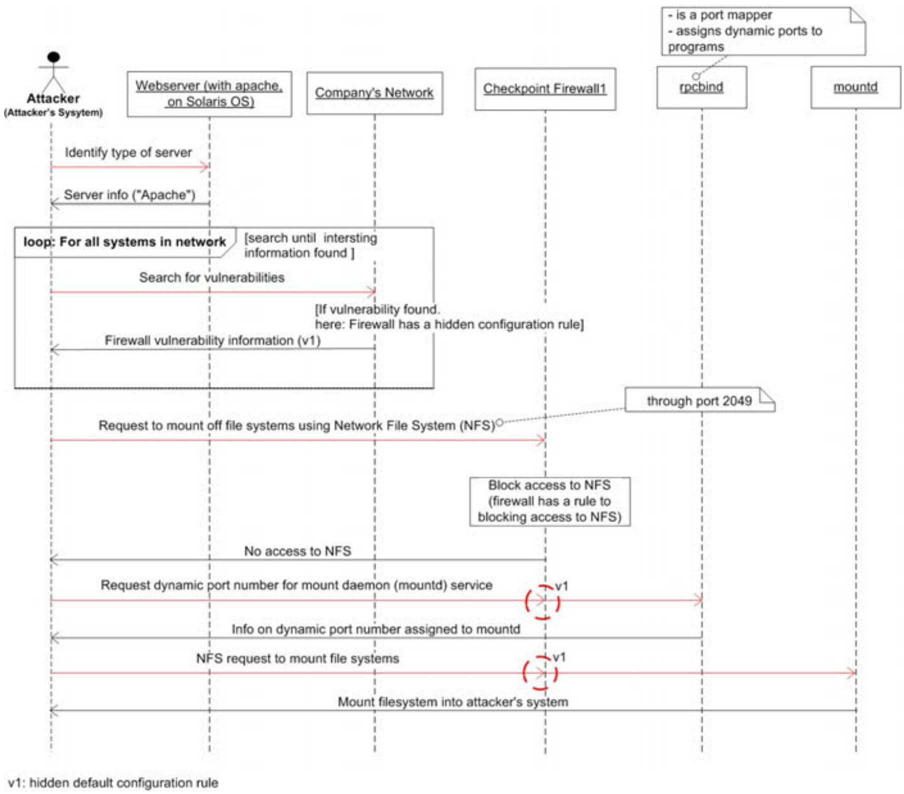

Fig. 4. MUSD for the (part of) penetration test example from the experiment

4.1 Experimental Design

To compare the two techniques, we conducted a controlled experiment with 42 subjects who used the two techniques individually on two different cases described in the literature, a *bank intrusion* [2] and a *penetration test* [2]. To control for the order in which the techniques were used and the cases were solved, a Latin-Squares experimental design was used as shown in Table 1. The within-experiment data regarding understanding, performance and perception thereby became paired, comprising two dependent samples. Table 1 shows the order in which the techniques were used and cases were solved by the groups. Group 1's experiment sheet is available at [17].

Table 1. Latin-Squares experimental design used in the experiment

Case order: Technique order:	Bank intrusion before penetration test	Penetration test before bank intrusion
MUCM before MUSD	Group 1	Group 2
MUSD before MUCM	Group 3	Group 4

4.2 Variables

After controlling for the participants' *backgrounds*, for each combination of technique and case, three types of tasks were solved: an *understanding* task, a *performance* task and a *perception* task. Table 2 summarizes the main variables used in the analysis.

Background was measured by a pre-task questionnaire addressing the participants' self-assessed knowledge of SD, MUSD, UCM, MUCM and security in general on a 5-point scale. They were also asked to report their numbers of completed *semesters of ICT studies* and months of *ICT-relevant work experience*.

Table 2. Variables used in the experiment

Name	Explanation
TECH=MUCM, TECH=MUSD	The *technique* used in that part of the experiment, either MUCM or MUSD.
CASE=BANK, CASE=PEN	The *case* solved in that part of the experiment, either BANK (the bank intrusion) or PEN (the penetration test).
KNOW_MOD, KNOW_SD, KNOW_MUSD, KNOW_UCM, KNOW_MUCM, KNOW_SEC	The participants' self-assessed knowledge about *systems modelling* (KNOW_MOD), *sequence diagrams* (KNOW_SD), *misuse sequence diagrams* (KNOW_MUSD), *use case diagrams* (KNOW_UCM), *misuse case diagrams* (KNOW_MUCM) and *security analysis* (KNOW_SEC) on a 5-point scale, where 1 is “Never heard about it” and 5 is “Expert”.
STUDY	The participants' self-reported *semesters* of ICT-studies.
JOB	The participants' self-reported *months* of ICT-relevant work experience.
UND_1, UND_2, ...	The 20 statements about the case, scored by the participants as either true or false. A correct assessment is scored 1 and a wrong one -1.
UND_MUCM, UND_MUSD, UND_NEUT	Sum of scores for the statements about of MUCM (architectural issues), MUSD (temporal sequence issues) and neutral aspects of the problem cases.
UND_TOT	Sum of scores for the all twenty statements about the problem cases.
VULN, MITIG	The numbers of unique vulnerabilities and mitigations identified by the participants.
VUMI	The sum of unique vulnerabilities and mitigations identified by the participants.
PER_1, PER_2, ...	Scores on the 5-point Likert scales for the individual statements about perception of the techniques.
PER_PU, PER_PEOU, PER_ITU	Average scores on the 5-point Likert scales for the four statements about perceived usefulness of, perceived ease of use of and intention to use of the techniques.
PER_AVE	Average scores on the 5-point Likert scales for all the twelve statements about the techniques.

Understanding was measured by 20 true/false questions about the case, scoring 1 point for a correct answer, -1 for incorrect, and 0 points for no answer. The statements were designed so that 6 of the statements addressed aspects where MUCM was assumed to be a better technique, 6 addressed aspects where MUSD was assumed to be better and 8 addressed aspects where neither technique was assumed to have an advantage. Specifically, we assumed MUCM would perform best for statements relating to architecture and that MUSD would perform better for statements about the sequence of activities. In each group, half the statements had a positive and half a negative formulation (so that equal numbers of *true* and *false* answers were expected).

Performance was measured by asking the participants to identify and list as many vulnerabilities and mitigations as they could in the planned system. Then the numbers of unique (and relevant) vulnerabilities and mitigations were counted. Three example vulnerabilities were given for both cases. Even though the type and criticality of the vulnerabilities are important, these issues were out of scope for the experiment. Such issues can be considered as a part of the future work.

Perception was measured by a post-task questionnaire adapted from the Technology Acceptance Model (TAM) [18,23]. 4 questions addressed *perceived usefulness* (PU), 4 addressed *perceived ease of use* (PEOU) and 4 investigated the participants' *intention to use* (ITU) the technique in the future. One statement in each group was negative, with a lower score reflecting a positive opinion, inverted before analysis.

Table 3. Hypotheses of the comparison experiment

$H1_1$	Better score for architectural Q's with MUCM than MUSD. That is, more questions about architectural issues were assessed correctly with MUCMs than with MUSDs.	UND_MUCM[MUCM] > UND_MUCM[MUSD]
$H2_1$	Better score for temporal sequence Q's with MUSD than MUCM. That is, more questions about temporal sequence issues were assessed correctly with MUSDs than with MUCMs.	UND_MUSD[MUCM] < UND_MUSD[MUSD]
$H3_1$	Different numbers of statements in the NEUT group were assessed correctly with MUCMs and MUSDs.	UND_NEUT[MUCM] ≠ UND_NEUT[MUSD]
$H4_1$	Different numbers of statements were assessed correctly with MUCMs and with MUSDs.	UND_TOT[MUCM] ≠ UND_TOT[MUSD]
$H5_1$	Different numbers of vulnerabilities were identified with MUCMs and with MUSDs.	VULN[MUCM] ≠ VULN[MUSD]
$H6_1$	Different numbers of mitigations were identified with MUCMs and with MUSDs.	MITI[MUCM] ≠ MITI[MUSD]
$H7_1$	Different numbers of vulnerabilities and mitigations were identified with MUCMs and with MUSDs.	VUMI[MUCM] ≠ VUMI[MUSD]
$H8_1$	The usefulness of MUCMs and MUSDs were perceived differently.	PER_PU[MUCM] ≠ PER_PU[MUSD]
$H9_1$	The ease of use of MUCMs and MUSDs were perceived differently.	PER_PEOU[MUCM] ≠ PER_PEOU[MUSD]
$H10_1$	The intentions to use MUCMs and MUSDs again were different.	PER_ITU[MUCM] ≠ PER_ITU[MUSD]
$H11_1$	MUCMs and MUSDs were perceived differently.	PER_AVE[MUCM] ≠ PER_AVE[MUSD]

4.3 Hypotheses

The hypotheses for our experiment are listed in Table 3. The corresponding set of null hypotheses, as well as an additional set of hypotheses about correlations between understanding ($H1_1$-$H4_1$), performance ($H5_1$-$H7_1$) and perception ($H8_1$-$H11_1$), are omitted here for space reasons.

4.4 Experimental Procedure

The 42 participants of the experiment were recruited from a class of second year computer science students, receiving financial support for an excursion as "payment" for participating. Each group consisted of 10 or 11 students solving the task under equal conditions (same room, same time limits). The experiment comprised 10 steps:

1. Filling in the pre-experiment questionnaire (2 min)
2. Reading a short introduction to the experiment (1 min)
3. Using the first assigned technique on the first assigned case:
   a. Reading the introduction to the first technique (1.5 pages, 9 min)
   b. Reading the textual description of case 1 while looking at the related diagrams (3-4 pages, 12 min)
   c. Answering 20 true/false questions about the case (8 min)
   d. Finding as many vulnerabilities and mitigations as possible (11 min)
   e. Filling in post-experiment questionnaire (4 min)
4. Easy physical exercises as a break (2 min)
5. Repeat steps 3a-e for the second technique and case (7+ 14 + 5 + 10 + 4 min)

The duration of the steps was decided dynamically. There was always enough time to finish the steps, except for steps 3c and 5c, which we stopped before everyone could finish because we wanted to see how efficient the participants were.

5 Results

5.1 Comparing Backgrounds

We used Kruskal-Wallis H tests of four independent groups for all background variables to control for differences between the participant groups. We found no significant differences, neither with respect to *knowledge*, *study semesters* nor *job months*.

We used 2-tailed Wilcoxon signed-rank tests to compare the participants self-assessed knowledge backgrounds in the following areas. They reported being significantly more knowledgeable about *systems modelling* than *security analysis* (KNOW_MOD = 2.79, KNOW_SEC = 1.81, p = .000), as well as about *sequence diagrams* versus *use case maps* (KNOW_SD = 3.20, KNOW_UCM = 2.74, p = .003). There was no significant difference in knowledge about *misuse sequence diagrams* and *misuse case maps* (KNOW_MUSD = 1.37, KNOW_MUCM 1.36).

The participants reported between 2 and 4 semesters of ICT-studies, with a mean of 3.05 and a small standard deviation. They reported 2.07 months of ICT-relevant work experience. Here the standard deviation was higher due to three outliers with 6, 19 and 54 months. All the others reported 2 months or less.

Table 4. Comparison results for understanding

Statement group	Understanding					
	MUCM Mean	MUCM St.Dev.	MUSD Mean	MUSD St.Dev.	Z	Sign. (exact)
MUCM group (UND_MUCM)	3,24	2,01	2,33	2,21	-1,87	p=0.031
MUSD group (UND_MUSD)	2,36	2,37	3,62	2,44	-2,89	p=0.001
NEUT group (UND_NEUT)	5,62	1,83	5,62	1,95	-0,23	–
All statements (UND_ALL)	11,21	5,20	11,57	4,25	-0,22	–

Table 5. Effect and sample sizes for the significant differences in understanding

Statement group	Understanding				
	Pooled st.dev.	Effect size	Cohen	Hopkins	Sample size required
MUCM (UND_MUCM)	2,087	0,44	medium	small	169
MUSD (UND_MUSD)	2,376	-0,53	medium	small	114

5.2 Understanding

We performed Wilcoxon signed-rank tests of two paired groups for all understanding variables using exact significances to compare how well the participants assessed statements about their cases using MUCM and MUSD (see Table 4). As expected, use of MUCM gave more correct answers on architectural questions than MUSD ($p = .031$, 1-tailed), while MUSD yielded more correct answers than MUCM for questions about temporal sequence ($p = .001$, 1-tailed). There were no other significant differences. Hence, H1 and H2 are confirmed while H3 and H4 are rejected.

The effect sizes for the groups with significant differences proved to be medium according Cohen's classification [19] or small according Hopkins's [20]. It was 0.436 for MUCM group (UND_MUCM) case and -0.53 for MUSD groups (UND_MUSD) case (see Table 5). The positive value here means that MUCM had the advantage while the negative favours the MUSD technique.

We also used Wilcoxon signed-ranks to control for differences in the participants' understanding of the two different cases and their performance depending on technique order, but found no significant differences.

5.3 Performance

We had two completely blank responses for the task of identifying vulnerabilities and mitigations. Both came from groups of 11 participants. After removing the outliers,

Table 6. Comparison results for performance

Identification task	Performance					
	MUCM Mean	MUCM St.Dev.	MUSD Mean	MUSD St.Dev.	Z	Sign. (exact)
Vulnerabilities (VULN)	4,18	1,75	4,08	1,67	-0.75	–
Mitigations (MITI)	3,22	1,72	3,42	1,54	-0.18	–
Both (VUMI)	6,75	3,23	7,32	3	-0.49	–

we performed Wilcoxon signed-rank tests of two paired groups for all performance variables using exact 2-tailed significances to compare how well the participants identified vulnerabilities and mitigations using MUCM and MUSD. We found no significant results, thus rejecting H5, H6, and H7.

We also used Wilcoxon signed-ranks to compare the participants' performance for the two cases for the first versus second technique used. We found that significantly more vulnerabilities and mitigations (2-tailed case for both) were identified for the bank intrusion case.

We did not attempt to rate, categorise or otherwise analyse the vulnerabilities and mitigations in further detail, leaving this for further work. In particular, further work should investigate whether there are systematic differences between the types of vulnerabilities and mitigations identified using the two techniques.

5.4 Perception

Finally, we performed Wilcoxon signed-rank tests of two paired groups for all perception variables using exact 2-tailed significances to compare how the participants perceived the techniques in terms of usefulness and ease of use and whether they intended to use them again in the future. The participants perceived MUSDs significantly more positively than MUCMs, both for perceived usefulness ($p = .002$), perceived ease of use ($p = .000$), intention to use ($p = .000$) and on average for all perception questions ($p = .000$). Hence, hypotheses 8, 9, 10 and 11 are all confirmed.

All individual questions also gave significant results for the Wilcoxon signed-rank test in favour of MUSD, except statement 5, which did not give a significant result in either direction. This was related to the PU variable and stated that the technique "would be useless in analyzing security vulnerabilities of computer systems."

The effect sizes for the perception measures were higher than for understanding: 0.547, -1.19, -0.957 and -1.027 for PU, PEOU, ITU and their averages (see Table 8).

We also used Wilcoxon signed-ranks to compare the participants' perceptions for the bank intrusion case and the penetration test case and to compare their perceptions of the first technique they used with the second one. The only significant difference was that the participants perceived their first technique used as more useful ($p = .023$).

Table 7. Comparison results for perception, all p-values 2-tailed

TAM variable	Perception					
	MUCM Mean	MUCM St.Dev.	MUSD Mean	MUSD St.Dev.	Z	Sign. (exact)
Perceived usefulness (PU)	3,68	0,70	4,05	0,67	-3,05	p = .002
Perceived ease of use (PEOU)	3,26	0,76	4,06	0,59	-4,44	p = .000
Intention to use (ITU)	3,07	0,82	3,80	0,72	-4,16	p = .000
Average (AVE)	3,34	0,65	3,97	0,59	-4,25	p = .000

Table 8. Effect and sample sizes for the significant differences in perception

TAM variable	Perception				
	Pooled st. dev.	Effect size	Cohen	Hopkins	Sample size required
Perceived usefulness (PU)	0,677	-0,55	medium	small	108
Perceived ease of use (PEOU)	0,672	-1,19	large	large	23
Intention to use (ITU)	0,762	-0,96	large	moderate	35
Average (AVE)	0,613	-1,03	large	moderate	31

6 Discussion

6.1 Main Findings

The experimental comparison indicates that the two techniques are complementary in terms of understanding. They aid understanding of different aspects about intrusions into systems, i.e., the architecture (MUCM) and the sequence of events, actions (MUSD). They are equal in terms of performance, i.e., they encourage users to identify similar numbers of vulnerabilities and mitigations in the planned system, although further analysis is needed to investigate if they encourage identifying the same types of vulnerabilities and mitigations. However, MUSD is perceived more positively by users, i.e., they rate the technique more highly in terms of perceived usefulness, perceived ease of use and intention to use. The difference is more marked for perceived ease of use. A summary on the decisions about the hypotheses can be seen in Table 9.

It must be admitted, of course, that the results of this experiments are not particularly surprising. MUSD was better for attack sequence knowledge, and MUCM for architectural knowledge, just as suspected. As for the participants' preference towards MUSD, this may plausibly be explained by the fact that the students were previously familiar with sequence diagrams, but not with use case maps, hence MUSD may have required a smaller learning effort than MUCM. Although the result was in many ways as expected, it still feels useful to have such suspicions confirmed by controlled experiments rather than letting them remain on the purely speculative level.

Table 9. Results of hypothesis testing (A = accept, R = reject)

$H1_1$	$H2_1$	$H3_1$	$H4_1$	$H5_1$	$H6_1$	$H7_1$	$H8_1$	$H9_1$	$H10_1$	$H11_1$
A	A	R	R	R	R	R	A	A	A	A

6.2 Implication for the MUSD Technique

Our intention is to combine MUSD with other related techniques to provide complete intrusion models. As indicated, MUCM could reflect the architecture related aspects and MUSD the order of the steps. In the future, it would be interesting to see how MUSD could be used along with other attack modelling techniques.

Since UML sequence diagrams (SD) are well known, it would be easy to adapt and use MUSD in system development projects. This is supported by the results of the experiment, indicating a positive perception towards MUSD. Further investigation is needed to explore how MUSD can utilize various SD notations like Decomposition and InteractionUse to improve the modelling.

6.3 Threats to Validity

In [21] conclusion, internal, construct and external validity are suggested as relevant categories for the validity of experiments. *Conclusion validity* concerns the relationship between the technique used and the outcome in terms of scores for tests and post-task questionnaire. One important question is whether the sample size is big enough to justify the conclusions drawn. The main effect claimed about understanding was a significant advantage for MUCM regarding architecture related statements and for MUSD regarding event sequence related statements. MUSD was also significantly more favoured than MUCM regarding PU, PEOU, ITU and their averages.

Denoting the Type I error probability by α and the Type II error probability by β, the following relationship holds:

$$N = \frac{4(u_{\alpha/2} + u_{\beta})^2}{ES^2} .$$

If we use $\alpha = 0.05$ and $\beta = 0.20$, we get $N = 32/(ES)^2$ [21] as a required sample size or $ES = \sqrt{32/N}$ as required effect size. The results are presented in Table 5 and 8.

Hence, we have sufficient observations only for PEOU, ITU and the averages (AVE). With the actual sample size of 42 we would have needed an ES of at least 0.873 for the three other cases. We note that these effect sizes must be used with caution, because our data are not in general normally distributed.

Internal validity assesses whether the observed outcomes were due to the treatment or to other factors. To avoid the threat of selection bias, Latin-Squares experimental design was used where all participants tried both techniques. Moreover, the participants were randomly assigned to the four groups and a pre-experiment questionnaire was used to control for confounding factors. The Latin-Squares design eliminated potential problems with learning effects, boredom or fatigue as the participants tried the techniques in different orders. Furthermore, all subjects were in the same auditorium with equal working conditions, and sitting far apart with no interaction between them.

Construct validity concerns whether it is legitimate to infer from the measures made in the experiment to the theoretical constructs that one was trying to observe. With respect to the performance data, we were trying to observe the understanding and problem solving effectiveness achieved using the respective techniques, and this was measured by scores on 20 true/false statements and the number of vulnerabilities and mitigations identified. Of course, there are other ways to explore understanding or effectiveness, but the ability to answer questions about a case should be a reasonable approximation of understanding, and the identification of vulnerabilities and mitigations would be relevant problem solving tasks in secure software engineering.

External validity is concerned with the question of whether it is possible to generalize from the experimental setting to other situations, most importantly to industrial systems development. The use of students instead of practitioners is a notable threat but as observed in [22], the level of competence is more relevant for performance in an experiment than whether the person is student or practitioner. Our participants were soon to finish their second year of computing studies. They had learnt about system modelling techniques (e.g. sequence diagrams). Moreover, the task in this experiment was to learn two new methods and apply them on a task explained during the experiment – the difference between students and practitioners would probably have been bigger if the task was about using a method well-known to practitioners.

7 Conclusions and Further Work

The paper has presented misuse sequence diagrams (MUSD) for visualizing system intrusions and compared it with a similar technique, MUCM, through a controlled experiment. The results indicate that MUSD and MUCM are complementary techniques having their strengths on visualizing temporal sequence of actions and architectural issues respectively. Experimental comparison between MUSD and other approaches not invented by the authors are planned for the future.

Further work is needed to improve MUSD for better intrusion modelling. For example, MUSD could be extended to incorporate UML sequence diagram features like InteractionUse and Decomposition. It is also needed to look into how MUSD could be combined with other techniques to provide a complete picture of intrusions. In particular, this experiment indicates that it would be interesting to run future experiments where MUSD and MUCM are used together instead of as competitors, to see if this gives better results than using either technique alone. It would be also interesting to see whether MUSD can be extended to model dependability issues other than security and to evaluate it also through larger industrial case studies, since controlled experiments can only investigate tasks of a fairly limited size.

References

1. Aagedal, J.Ø., et al.: Model-based Risk Assessment to Improve Enterprise Security. In: Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC 2002). IEEE, Los Alamitos (2002)
2. Mitnick, K.D., Simon, W.L.: The Art of Intrusion. Wiley Publishing Inc., Chichester (2006)

3. Schneier, B.: Secrets and Lies: Digital Security in a Networked World. Wiley, Chichester (2000)
4. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering 10(1), 34–44 (2005)
5. Karpati, P., Sindre, G., Opdahl, A.L.: Illustrating Cyber Attacks with Misuse Case Maps. Accepted to 16th International Working Conference on Requirements Engineering: Foundation for Software Quality, RefsQ 2010 (2010)
6. Karpati, P., Opdahl, A.L., Sindre, G.: Experimental evaluation of misuse case maps for eliciting security requirements. Submitted to 18th IEEE International Conference on Requirements Engineering, RE 2010 (2010)
7. Unified Modeling Language, `http://www.uml.org` (accessed 4.6.2010)
8. Internet Security Glossary, `http://www.apps.ietf.org/rfc/rfc2828.html` (accessed 22.6.2010)
9. Opdahl, A.L., Sindre, G.: Experimental comparison of attack trees and misuse cases for security threat identification. Information and Software Technology 51(5), 916–932 (2009)
10. Buhr, R.J.A.: Use Case Maps: A New Model to Bridge the Gap Between Requirements and Detailed Design. In: 11th Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA 1995), Real Time Workshop, p. 4 (1995)
11. Markose, S., Xiaoqing, L., McMillin, B.: A Systematic Framework for Structured Object-Oriented Security Requirements Analysis in Embedded Systems. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 1, pp. 75–81 (2008)
12. Georg, G., Ray, I., Anastasakis, K., Bordbar, B., Toahchoodee, M., Houmb, S.H.: An aspect-oriented methodology for designing secure applications. Information and Software Technology 51, 846–864 (2009)
13. Redmill, F., Chudleigh, M., Catmur, J.: Hazop and software Hazop. Wiley, Chichester (1999)
14. IEC 61025: Fault tree analysis (FTA), IEC Standard (2006)
15. Runde, R.K., Haugen, Ø., Stølen, K.: The Pragmatics of STAIRS, Research Report 349 (January 2007)
16. Jürjens, J.: Secure Systems Development with UML. Springer, Heidelberg (2005)
17. Karpati, P.: `http://www.idi.ntnu.no/~kpeter/ExampleSheet_Group1.pdf` (accessed 24.6.2010)
18. Davis, F.D.: Perceived usefulness, perceived ease of use and user acceptance of information technology. MIS Quarterly 13, 319–340 (1989)
19. Cohen, J.: Statistical power analysis for the behavioral sciences, 2nd edn. Lawrence Erlbaum, New Jersey (1988)
20. Hopkins, W.G.: A New View of Statistics. University of Queensland, Brisbane (2001)
21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic, Norwell (2000)
22. Arisholm, E., Sjøberg, D.I.K.: Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software. IEEE Transactions on Software Engineering 30, 521–534 (2004)
23. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. MIS Quarterly 27(3), 425–478 (2003)

Needs-Driven Bundling of Hosted ICT Services

Jaap Gordijn[1], Floris de Haan[2], Sybren de Kinderen[3], and Hans Akkermans[1]

[1] VUA Amsterdam, The Netherlands
{gordijn,elly}@cs.vu.nl
[2] OGD, The Netherlands
frhaan@gmail.com
[3] CRP Henri Tudor, Luxembourg
sybren.dekinderen@tudor.lu

Abstract. Increasingly ICT (Information & Communication Technology) services become *commercial* services. For example, an Internet Service Provider (ISP) offers email, web browsing and content hosting as commercial services. In this paper we present an approach, e^3*service* , to semi-automatically generate such services, satisfying a stated complex customer need. The e^3*service* approach elicits the customer need, the consequences satisfying the need, and services satisfying the need. We show how e^3*service* works in practice using a running, industry strength, case study.

Keywords: service, need, bundling.

1 Introduction

Today's economy increasingly becomes a *service* economy. We consider a service as having an intangible nature [11], a processual nature [5], and producing valueable outcomes [9]. So, we take mainly a *commercial* perspective on 'service'.

The focus in this paper is on commercial *ICT* (Information & Communication Technology)-services. ICT-services are just like normal commercial services; only additionally, ICT-services can be provisioned online. Examples from our case study partner include virtual desktops, accessible via thin-clients, and backup services. Since ICT services are provisioned online, it is therefore important that ordering of these ICT-services can also be done online.

Increasingly, such ICT services are sold as service *bundles*. Service bundles consist of more elementary services. These elementary services may be offered by multiple suppliers. Usually, suppliers bundle services to satisfy more complex customer needs. As an example, consider an Internet Service Provider (ISP). A bundle of an ISP often contains a connectivity service (e.g. for web surfing), an emailbox service, and a webpage hosting service. Note that all these elementary services may be offered by multiple ISPs (suppliers).

To support an online ordering process for ICT services we have developed the e^3*service* ontology. This ontology allows for reasoning about customer needs, services and service bundles. The e^3*service* ontology is capable of representing a service catalogue both from a customer and supplier perspective. The

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 16–30, 2010.

customer perspective catalogue contains concepts such as 'customer need' and 'consequence' (consequence of satisfying a need). The supplier perspective catalogue entails concepts such as 'service' and also 'consequence'. The notion of consequence is used to connect the supplier perspective with customer perspective. The e^3*service* ontology has also reasoning capabilities as it (1) can semi-automatically derive a set of consequences as a result of satisfying a customer need, (2) can match the found consequences with available service bundles, and (3) can extend service bundles with additional services which also might be of value for the customer.

In sum, the contribution of this paper is that we propose the e^3*service* ontology for reasoning about customer needs, consequences, and service bundles. Additionally, we show how e^3*service* works in a real-life case study.

Finally, it is important to understand that the e^3*service* ontology is different from ontologies in the field of web-services, (e.g. WSMO [8]). The e^3*service* ontology is about *commercial* services, whereas web-services provide a platform to solve interoperability and orchestration between *software components*.

This paper is organized as follows. In Sect. 2 we introduce the running case study. Sect. 3 presents the e^3*service* ontology. In Sect. 4 we show how to reason with the e^3*service* ontology about customer needs and service bundles. Finally, in Sect. 5 we present lessons learned and in Sect. 6 our conclusions.

2 The OGD Case Study

OGD is a Dutch ICT service provider (+/- 750 employees) that has recently started to provide hosted ICT-services. Currently, OGD offers Historium, an online back-up service and Officium a virtual workspace for a client that is accessed remotely through a thin client environment (called 'hosted desktop').

Offering hosted services is relatively new to OGD. As a result, OGD currently lacks a coherent idea of the benefits of their service offerings, how individual services are interrelated, and what customer needs their services satisfy. This knowledge is either fragmented throughout OGD, or unknown altogether. For many OGD-employees, this can be problematic. Junior account managers may have trouble in stating why a service is interesting for a customer, while the marketing department may have trouble describing the service in offering texts.

Therefore, OGD wants to create a service catalogue that provides a uniform idea of (1) the benefits of the individual services from OGD, (2) relations between these services and (3) the customer needs these services satisfy.

In addition, OGD is interested in the e^3*service* software ontology to (1) train junior account managers and (2) structure the dialogue for personnel selling services by phone.

3 The e^3*service* Ontology

To represent the service catalogue of an enterprise, we utilize the e^3*service* ontology. This ontology takes two perspectives on services: (1) the supplier perspective, and (2) the customer perpective.

3.1 The Supplier Perspective

The supplier perspective of the e^3*service* ontology is largely based on [2,3]. Fig. 1 shows the e^3*service* ontology expressed as a high-level UML class diagram. Fig. 2 shows a sample of the supplier perspective service catalogue (cf. the e^3*service* ontology). Due to lack of space, we only discuss the most important parts of the ontology in detail.

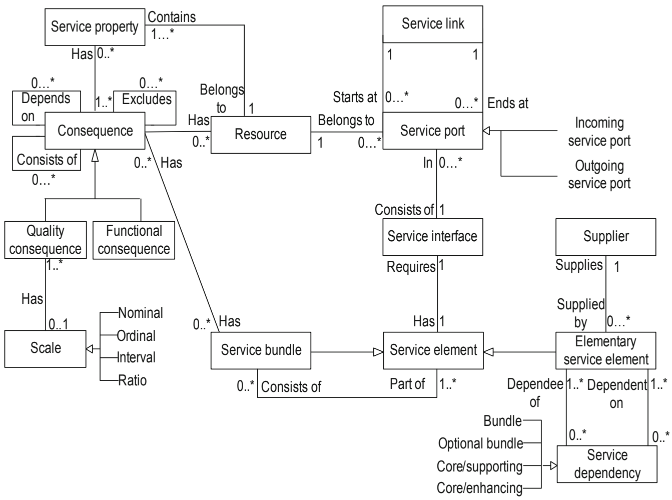

Fig. 1. The e^3*service* supplier perspective ontology

Service element. A service element is a (composed) activity that provides the service. Service elements can be *elementary*; then they can not decomposed further, or service elements can be a service *bundle*; then the bundle is composed of other service elements. Service elements are provisioned by a *supplier*. Service elements have service *interfaces*, which group service *ports*. Service ports provide or request *resources*, which are units of service delivery that can be provisioned in their own right and in a commercially feasible way. Service ports can be connected to each other via service *links*. Resources have service *properties*. As properties can not provisioned independently, they are part of resources.

Case study: For the OGD case (see Fig. 2), service elements include two e-mailing solutions: (1) hosted Exchange basic, a solution offering basic e-mailing capabilities such as sending and receiving e-mail through a standard e-mail client and (2) hosted Exchange complete, a solution that offers the same basic e-mail capabilities, only then supplemented with extra features such as e-mail access via the mobile phone, a mailbox that can be shared with others, etc.

Service dependencies. Next we model dependencies between individual services. Various kind of dependencies between services may exist, cf. [2,3] (where S_1 and S_2 denote service elements):

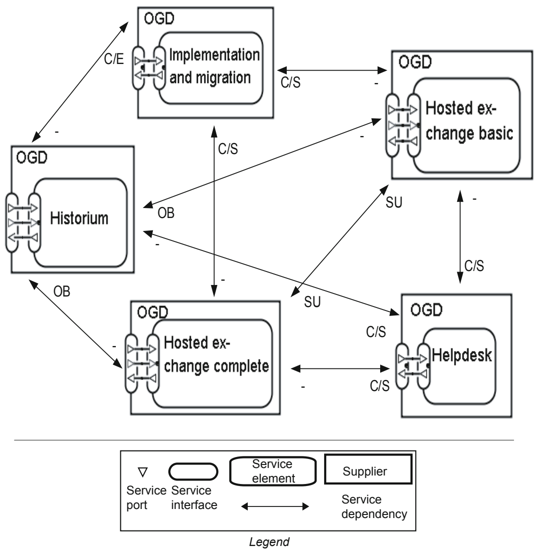

Fig. 2. Sample of the supplier perspective service catalogue

- S_1 is in a *Bundled* dependency with S_2 if S_1 is not provided separately from S_2 for *commercial* reasons. In literature, this is refered to as pure product bundling [12]. Notation: BU.
- S_1 is in a *Core/Supporting* dependency with S_2 if S_1 cannot be provided (for technical or legal reasons) without also providing S_2. The supporting service can be supplied by the same supplier as the supplier of the core service, but another case is that the supporting service is supplied by someone else. Notation: C/S.
- S_1 is in a *Core/Enhancing* relationship with S_2 if (1) S_2 possibly adds value to S_1, (2) acquisition of S_1 is obligatory for acquisition of S_2 and (3) S_1 can be acquired separately from S_2. Notation: C/E.
- S_1 is in a *Optionally Bundled* relationship with S_2 when (1) S_2 possibly adds value to S_1 and (2) S_1 and S_2 can be acquired separately. Note here that, as opposed to the C/E relationship, S_1 does not have to be acquired before S_2 can be acquired. Notation: OB.
- S_1 *excludes* S_2 if the supplier of S_1 prevents the customer to consume S_2, for example because S_2 is offered by a competitor, or because joint consumption is legally prohibited. Notation: EX.

Case study: The following service dependencies are of interest:

- Helpdesk support is a standard supporting service for the basic services.
- Implementation and migration is a standard supporting service for both hosted Exchange service offerings.
- Implementation and migration is an *enhancing* service for the online back up service historium and as such, not included as a standard supporting service.

The reason for this is twofold: (1) installation is relatively straightforward, requiring a single installation of a software tool on a server, and (2) OGD deems historium not to be as 'business critical' as an e-mail service.
- Hosted Exchange complete (the hosted e-mail solution from OGD) is in an OB relationship with Historium (the online backup service from OGD) to indicate that account managers often offer functionality of one of these services in combination with functionality of the other service. However, both services can also be acquired separately.
- Hosted Exchange complete is in an C/E relationship with the service Exchange brick-level. This indicates that account managers often offer functionalities provided by these services in combination, but also that they never sell Exchange brick-level separate from hosted Exchange complete. This is because the Exchange brick level is a component that allows for making backups of individual mailboxes and as such, only makes sense in combination with a basic e-mail service.

Generate possible service bundles. Based on the individual services and dependencies that exist between these services, we generate all possible service bundles. These generated bundles together form the service catalogue of OGD. For example: From the dependency OB(Hosted Exchange complete, Historium) we generate two possible bundles: {Historium (OGD)} and {Historium (OGD), Hosted Exchange complete (OGD)}. A sample of the pregenerated bundles of hosting services can be found in Table 1.

Service consequence. Services may have consequences. A consequence is anything that results from consuming (a combination of) valuable service properties offered by resources of a service (see [10]). There exist several supply-side types of consequences (functional and quality) and relations between these consequences. Since consequences are used in both the customer and supplier perspective of e^3*service* , they form the glue between both ontological perspectives.

Case study: For the OGD case (see Fig. 2), we identify the consequences from the individual services. For a sample of the identified consequences, see Table 1. First, we find functional consequences, such as 'send and receive e-mail' for the hosted Exchange services, and 'generic backup capability' for the Historium service. Second, we find quality consequences, such as 'e-mail access by phone' and 'e-mail access via web browser' for the service Hosted Exchange complete and 'within 4 hours' and 'within 7 hours' for the service Historium.

Then we group quality consequences using the concept of a *scale*. We group 'e-mail access by phone' and 'e-mail access via web browser' under the *nominal* scale 'e-mail access method' because the preference ordering depends on the customer. We group 'within 4 hours' and 'within 7 hours' under the *ordinal* scale 'response time', because this indicates the different response times in recovering a lost dataset. If someone considers response time important, the shorter response time is always preferred over the longer one.

Table 1. Sample of generated service bundles and their respective functional consequences

Service bundle	Functional consequences
Hosted Exchange basic	Access to email, contact support, don't receive unwanted email, don't receive viruses through email, lower total cost of ownership, send and receive email, see progress of open calls, single point of contact
Hosted Exchange complete	Access to email, contact support, don't receive unwanted email, do not receive viruses through email, lower total cost of ownership, make appointments with colleagues, make group email addresses, option to give colleagues access to your email, send and receive email, see progress of open calls, single point of contact
{Historium, hosted Exchange complete, Exchange brick-level }	Access to email, contact support, don't receive unanted email, do not receive viruses through email, lower total cost of ownership, make appointments with colleagues, make group email addresses, option to give colleagues access to your email, send and receive email, see progress of open calls, single point of contact, automatic back-up, contact support, data is secure, easy to use backup, free software updates, know if a backup was successful, receive a report with usage statistics, restore server to original state, see progress of open calls
{Historium, hosted Exchange basic}	Access to email, contact support, don't receive unwanted email, do not receive viruses through email, backup happens without active involvement of local ICT personnel, contact support, data is secure, easy to use backup, free software updates, know if a backup was successful, receive a report with usage statistics, restore server to original state, see progress of open calls, single point of contact

Identify positive consequences for the bundles, additional to the consequences of the individual services. Next, we review if the service *bundles* lead to additional positive consequences.

Case study: For OGD, we find for example that the services in a bundle such as {Historium (OGD), Hosted Exchange complete (OGD)} are supported by the same helpdesk. Thus, for such a bundle of hosted services from OGD, 'single point of contact' can be an additional positive consequence for the bundle.

3.2 The Customer Perspective

Want. A want is a specific, supplier-independent solution that is commercially feasible to be provisioned on its own. As a want indicates a solution available in the market, at least one supplier should be willing to provide the solution.

Case study: Fig. 4 shows various wants: Hosted desktop, online backup, hosted Exchange, mailbox backup. The wants correspond to elementary services offered by OGD.

Consequence. As already explained, the consequences for the customer are the same as the supplier conquence, as the notion of consequence is used to match the supplier perspective with the customer perspective.

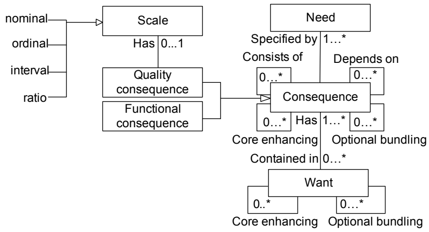

Fig. 3. The e^3*service* customer perspective ontology

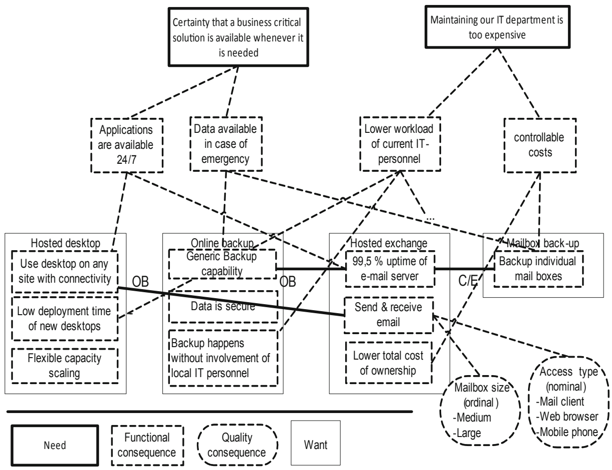

Fig. 4. Sample of the customer perspective service catalogue

Case study: Fig. 4 presents various functional consequences: send and receive email, generic backup capability; and various quality consequences: mailbox size and access type. We consider 95% uptime as a functional consequence because account managers indicate that cosumers articulate a direct need (applications are available 24/7) for this consequence.

Consequence dependency. Two types of dependencies between consequences may be of interest for the customer: The Core/Enhancing dependency and the Optional Bundling dependency. These dependencies have alreay been discussed for the supplier perspective.

Case study: The Generic backup capability is an Optional Bundling relationship with 99.5% uptime of email server. Both consequences can be separately obtained, but can add value to each other when sold in a bundle.

The 99.5% uptime of email server is in a core/enhancing relationship with Backup of individual mailboxes. The Backup of individual mailboxes consequence can only meaningfully be obtained in combination with a mailbox.

Need. A need represents a problem statement or goal, independently from a solution direction (see[1]).

Case study: Account managers from OGD often mention two customer concerns that lead organizations to acquire hosted services from OGD: (1) an ICT solution should be available when needed, and (2) an IT department is too costly.

4 Reasoning about Hosted ICT Services

4.1 Generic Reasoning Structure of e^3*service*

Fig. 5 shows the high-level reasoning process of e^3*service* . The customer starts the process by selecting a need out of the known needs in the service catalogue. Then the custumer chooses consequences and therefore selects the valuable features s/he wants to obtain from a service. Next, the chosen customer consequences are matched to the supplier consequences, as annotated to service bundles, to find service bundles that can offer these consequences.

In case no service bundles are found, the customer is asked to reconsider the desired consequences. If service bundles are found, chances are high that these bundles come with additional positive and negative consequences (e.g. costs). Therefore, the customer is asked to score these additional consequences, and to consider trade-offs.

In case a bundle is rejected as a result of considering trade-offs, a so-called critique step is done to identify which consequence is responsible for the negatively scored bundle. The customer can then restart the reasoning process, by deselecting the consequence responsible for the negatively scored bundle.

The customer may decide to obtain the bundle, or to consider value-enhancing consequences. The customer then searches for services that provide added value to the basic service bundle already selected. The reasoning process restarts then by considering additional consequences for value enhancing services.

We have implemented this reasoning process in a Java-based software reasoner. The reasoner uses the customer and supplier based service catalogue, and

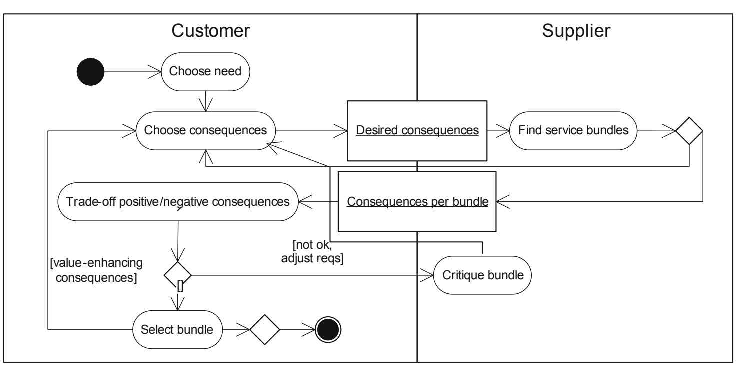

Fig. 5. The generic reasoning structure of e^3*service*

selects by following the aforementioned reasoning process a service bundle. For the OGD case at hand, we illustrate a typical scenario of service bundle selection.

4.2 The Reasoning Process for the OGD Case

We now display the reasoning process for hosted services. In particular, we focus on customer-supplier interaction and reasoning with value-enhancing services. We take the following scenario as a starting point:

'An antique dealer's current e-mail solution is not reliable enough. His server is outdated and sometimes crashes which causes some e-mail to not be received.'

Step 1: Choose need and choose consequences. Of the two most commonly heard needs from OGD, the need 'we need certainty that a business critical technical solution is available whenever needed' is selected for the antique dealer since this comes closest the problem he currently faces: An unreliable e-mail solution.

Next, the antique dealer chooses the consequence '99,5 % uptime of e-mail server' and proceeds to score the attached quality consequences from the nominal scale 'e-mail access method' and from the ordinal scale 'mailbox size'. We assume the antique dealer provides the importance scores presented in Table 2.
Outcome of this step: see Table 2.

Table 2. Mailing preferences entered by antique dealer

For the selected functional consequence, the quality consequences - per scale - are:
We found the following scale of quality consequences: e-mail access method
Please assign a score from 1(not important) to 10(vital) to each of these consequences
e-mail using Microsoft Outlook 6
e-mail access using a web browser 8
e-mail access by mobile phone 2
We found the following scale of quality consequences: mailbox size
1: large mailbox
2: small mailbox
please attach an importance rating (1-10) to this ordinal scale: 8

Step 2: Find and rank service bundles. We find the bundles 'Hosted Exchange Basic' and 'Hosted Exchange complete' for the must-have functional consequence '99,5 % uptime of e-mail server'. For this, the reasoning process:

1. Matches the customer and supplier perspective consequences '99,5 % uptime of e-mail server';
2. The supplier perspective consequence '99,5 % uptime of e-mail server' is traced to the supplier-specific resources 'Hosted Exchange basic' and 'Hosted Exchange complete'
3. The supplier-specific resources are traced to the service ports that correspond to the service bundles 'hosted Exchange basic' and 'hosted Exchange complete'.

Next, bundle scores are calculated for the found bundles. For the bundle 'Hosted Exchange basic' this calculation is presented below.

- 'small mailbox' scores 0.25 because this consequence (1) is defined on an ordinal scale and (2) is defined on an ordinal scale that contains one additional, higher ranked, consequence: 'large mailbox'. We use the Rank-Order Centroid method (ROC) [4], which can transform a qualitative ranking of a consequence into a quantitative ranking whose values are normalized to a value [0...1], to score items on an ordinal scale; the higher ranked consequence scores then 0.75, the lower ranked consequence 0.25.
- Other consequences present in the bundle, such as 'e-mail access method: E-mail access using a web browser' score 1 as they are defined on a nominal scale.

Note here that the scoring of consequences from an ordinal scale is performed differently from consequences defined on an ordinal scale. See Table 3 for a resulting ranked service bundles, as provided by the software reasoner.

Outcome of this step: The service bundles 'Hosted Exchange Basic and 'Hosted Exchange Complete, ranked according to how well they fit with the customer preferences (see Table 3).

Table 3. Tool output for two bundles of mailing services, ranked according to how well they fit with customer preferences

```
...
So the possible bundles, sorted according to preference, are:
1: hosted Exchange Standard Complete with the score: 5.3
2: hosted Exchange Basic with the score: 4.1
Please select a bundle. If none is to your liking, select 0 (zero).
```

Step 3: Trade off positive/negative consequences. The bundle 'Hosted Exchange complete' is more in accordance with the desired consequences than the bundle 'Hosted Exchange basic'. Yet, from the pricing models the antique dealer also observes that the complete bundle costs €10 per month per user, while the basic bundle costs €5 per user per month. For a fair weighing of these costs against the benefits provided by each bundle, the antique dealer therefore decides to also score negative consequences, such as '12 month commitment to using OGD services' and 'hosted Exchange complete fee'.

The provided scores are presented in Table 4. We can see that although the antique dealer scores the fee for Exchange complete higher than the fee for Exchange basic, Exchange complete still outranks Exchange basic, albeit with a smaller margin than for the bundle scores that are based on the positive consequences only (again, see Table 4). Thus, the antique dealer decides to acquire the bundle 'Hosted Exchange complete'.

Outcome of this step: The antique dealer chooses the bundle 'Hosted Exchange complete'.

Table 4. Trade of positive and negative consequences for hosted services

```
Would you also like to score all negative consequences to see how a bundle scores on a trade-off
between benefits and sacrifices?(y/n) y
Please assign a score from 1(does not really matter) to 10(this consequence must be lacking from
the bundle) to each of the following consequences:
no switching of suppliers for 12 months 3
hosted Exchange complete fee 8
hosted Exchange basic fee 4
Taking the negative scores into account, the new ordering of bundles is:
1. hosted Exchange Standard Complete Score: 4.2
2. hosted Exchange Basic: 3.6
Please select a bundle. If none is to your liking, select 0 (zero).
```

Step 4: Find value-enhancing services. Next, the antique dealer chooses consequences that OGD considers to be interdependent in demand with the functional consequences already included in the selected bundle. For the consequence '99,5 % uptime of e-mail server', we find two functional consequences: 'back up individual mail boxes' and 'generic back-up capability' (for reference, see the customer perspective catalogue in figure 4). Because the antique dealer maintains his backups on a server that can itself be considered an antique, we assume that the antique dealer is interested in both functional consequences.

Next, the reasoning process considers the quality consequences relevant for the scored functional consequences. As can be observed from the customer perspective service catalogue however (figure 4) none of the value-enhancing consequences contain quality consequences.

Outcome of this step: The additional, value-enhancing, consequences {back up individual mail boxes, quick recovery of individual mailboxes, automatic backup, data is secure, back-up happens without involvement of local IT personnel}.

Step 5: Find and choose bundle. To find bundles, the reasoning process uses the value-enhancing consequences as input, plus the consequences from the service bundle already selected ('Hosted Exchange complete').

As with the basic consequences, the reasoning process finds service bundles by (1) matching these consequences with all supply-side consequences (2) tracing the consequences on the supplier perspective to service bundles that satisfy these consequences.

Outcome of this step: The bundle 'Hosted Exchange complete Historium Exchange brick-level.

Step 6: Trade-off positive and negative consequences. The antique dealer receives a specification of the positive and negative consequences for the bundle 'Hosted Exchange complete Historium Exchange brick-level', plus a specification of the pricing model for this bundle. After reviewing the pricing model (see Table 5), the antique dealer decides to score the negative consequences for the bundle 'Hosted Exchange complete Historium Exchange brick-level' also.

Table 5. Pricing model for the bundle Hosted Exchange complete Historium Brick-level

For this bundle, we found the following pricing model:
*Hosted Exchange complete Historium Brick-level Exchange pricing model
This pricing model is of the type: single discount pricing model.
The price of this bundle is build up as follows:
* The two-part pricing model Hosted Exchange standard complete pricing model is build up as follows: X euro installation fee + 10 euro monthly/User. This pricing model is attached to the same port as the consequence hosted Exchange complete fee.
* The n-block pricing model Historium pricing model is build up as follows: Condition: Server 1: 60 euro monthly + Condition 2: Server 2: 55 euro/monthly + Condition 3: Server 3 and on: 50 euro/monthly/server. This pricing model is attached to the same port as the consequence online back-up fee.
* The usage-based pricing model Exchange brick-level pricing model is build up as follows: 50 euro/monthly. This pricing model is attached to the same port as the consequence back-up individual mailboxes fee.

The antique dealer has already agreed to the fee for hosted Exchange complete, and so attaches a score '2' (from 1 to 10, where 1 indicates 'does not matter' and 10 'won't have') to the negative consequence 'hosted Exchange complete fee'. The antique dealer attaches the score '6' to the negative consequence 'online back-up fee', considering that while he pays 60 euro monthly (due to the single server he has) he also does not have to worry about having to maintain a backup anymore. Finally, the antique dealer considers the negative consequence 'individual mailboxes fee' as a won't have and thus scores this as a '10'. He considers the investment in a dedicated backup for his e-mail not to be worthwhile €50 monthly, also when considering that he already has a generic backup capability.

Outcome of this step: No possible bundles, proceed with the critique-step.

Step 7: Find culprit. For the negative consequence that is scored as a won't have, the reasoning process now performs a critique-step to find the positive consequences attached to the cost-source. As such, the antique dealer is presented with the dialogue in Table 6.

Outcome of this step: The consequence 'Back up individual mail boxes' is found as the culprit.

Step 8: Choose consequences. Next, the antique dealer is asked to what place in the reasoning process he should return to adjust his requirements.

As can be seen in the dialogue from Table 7, the antique dealer selects (2): Start at value-enhancing requirements. He now returns to the basic bundle

Table 6. Critique: Find the preference that is too expensive to fulfill

The bundle Hosted Exchange complete Historium Exchange brick-level contains the won't have Brick level fee
We shall now investigate what positive consequences from this bundle require the negative consequence Brick level fee
The following group of positive consequences:
1: Back up individual mail boxes
1: Quick recovery of individual mail boxes
requires the negative consequence back-up individual mailboxes fee

Table 7. Adjusting preference scores

1. Start from group of detailed functional consequences you had initially chosen. This was:
*don't receive viruses through e-mail
*don't receive unwanted e-mail
*99,5 % uptime of e-mail server
...
2. Start at the value-enhancing consequences
*Back up individual mail boxes
*quick recovery of individual mailboxes
3. start all over, ie: from the need.
other (> 3): quit.

'Hosted Exchange complete' and answers 'no' when asked if he is interested in the value-enhancing consequence 'Back up individual mail boxes' and 'quick recovery of individual mailboxes'.

Outcome of this step: The consequences 'Back up individual mail boxes' and 'quick recovery of individual mailboxes' are no longer used as input for the reasoning process.

Step 9: Find service bundles. The reasoning process now again finds service bundles for the set of consequences belonging to the bundle 'Hosted Exchange complete Historium Exchange brick-level' *minus* the consequence 'Back up individual mail boxes'. Thus, the bundle 'Hosted Exchange complete Historium' is eventually found. Since this bundle contains an online e-mailing capability and online back-up functionality and the antique dealer finds the costs for both functionalities acceptable, he decides to acquire this service bundle.

Outcome of this step: The bundle 'Hosted Exchange complete Historium'.

5 Lessons Learned and Conclusions

Lesson 1: The reasoning performed by the software tool is similar to the reasoning performed by account managers. In reaction to a demonstration of our software reasoner, the account managers stated that the reasoning process performed by the reasoner is similar to how they sell services. In particular:

- Account managers start from a problem that the customer has, independently of the available solution. This is similar to our separation between needs (problem/goal) and consequences (valuable features that act as solution directions for satisfying the need).
- Constraints do influence decisions on what bundle the customer acquires
- Account managers first seek out a service that satisfies the basic requirements of the customer. Only thereafter, crossselling and upselling is performed. This is similar to our process of first agreeing on a basic service, and only then seeking out any value-enhancing services.

Lesson 2: Graphical representations are useful for together exploring services, but for usage on a daily basis a textual version is more adequate. The service catalogue that OGD uses on a daily basis is a stripped down *textual* version of the conceptual models presented in this paper. This textual representation is preferred by marketing personnel, who perceive a bullet list of services and needs more helpful in writing promotional texts than a conceptual model. Also, the account managers prefer a textual representation of the service catalogue, so that they can write a personalized offer text for customers. OGD-personnel did however declare that the formal conceptual models allowed for a structured exploration of their service offerings, and what needs these offerings satisfy.

6 Conclusion

In this paper we have presented the e^3*service* ontology. This ontology takes two perspectives on commercial services: the customer perspective and the supplier perspective. Both the customer and the supplier perspective contain the notion of 'consequence' of a service, the 'consequence' is therefore used to derive service bundles (supplier perspective) from customer needs (customer perspective). Moreover the e^3*service* ontology is capable of deriving extra, value adding, services to the services already found.

The reasoning process as done by the e^3*service* ontology comes close to the kind of reasoning performed by account managers. Therefore, the e^3*service* ontology can be used to explain the sales process to new account managers.

Two possible lines of further research can be foreseen. First, the service bundles can be derived dynamically, during the matching process of customer consequences and service consequences. Currently, all possible bundles are generated upfront the reasoning process. Second, the reasoning process could include the 'supplier of the supplier'. Now, the reasoning process is restricted to the supplier (OGD) satisfying the customer. But the supplier sometimes becomes a customer because additional services need to be obtained from others to provision a service.

Acknowledgements. The authors wish to thank OGD for the case study. This paper is based on [7, Ch 8] and [6].

References

1. Arndt, J.: How broad should the marketing concept be? Journal of Marketing 42(1), 101–103 (1978)
2. Baida, Z., Gordijn, J., Akkermans, H., Saele, H., Morch, A.Z.: Finding e-service offerings by computer-supported customer need reasoning. International Journal of E-Business Research (IJEBR) 1(3), 91–112 (2005)
3. Baida, Z., Gordijn, J., Akkermans, H., Saele, H., Morch, A.Z.: How e-Services Satisfy Customer Needs: a Software-aided Reasoning, ch. IX. Idea Group, USA (2006)
4. Hutton Barron, F., Barrett, B.E.: Decision quality using ranked attribute weights. Management Science 42(11), 1515–1523 (1996)
5. Bitner, M.J., Gremler, D.D., Zeithaml, V.A.: Services marketing: Integrating customer focus across the firm. McGraw-Hill, Irwin (2008)
6. de Haan, F.: Reasoning about e-services at operator group delft. Master's thesis, Vrije Universiteit Amsterdam (2009)
7. de Kinderen, S.: Needs-driven service bundling in a multi-supplier setting - The computational e^3 service approach. PhD thesis, Vrije Universiteit Amsterdam (2010)
8. Fensel, D., Lausen, H., Polleres, A., de Bruin, J., Sollberg, M., Roman, D., Domingue, J.: The Web Service Modeling Ontology. Springer, Berlin (2006)
9. Grönroos, C.: Service management and marketing: customer management in service competition. Wiley India Pvt. Ltd., Chichester (2007)
10. Gutman, J., Reynolds, T.J.: Laddering theory-analysis and interpretation. Journal of Advertising Research 28(1), 11 (1988)
11. Lovelock, C.: Service Marketing - People, Technology, Strategy, 4th edn. Prentice-Hall, Englewood Cliffs (2001)
12. Stremersch, S., Tellis, G.J.: Strategic bundling of products and prices: A new synthesis for marketing. Journal of Marketing 66(1), 55–72 (2002)

Enterprise Modeling for Business Intelligence

Daniele Barone[1], Eric Yu[2], Jihyun Won[3], Lei Jiang[1], and John Mylopoulos[3]

[1] Department of Computer Science, University of Toronto, Toronto (ON), Canada
barone,leijiang@cs.toronto.edu
[2] Faculty of Information, University of Toronto, Toronto (ON), Canada
yu@ischool.utoronto.ca
[3] DISI, University of Trento, Trento, Italy
jihyun.won@studenti.unitn.it,jm@disi.unitn.it

Abstract. Business Intelligence (BI) software aims to enable business users to easily access and analyze relevant enterprise information so that they can make timely and fact-based decisions. However, despite user-friendly features such as dashboards and other visualizations, business users still find BI software hard to use and inflexible for their needs. Furthermore, current BI initiatives require significant efforts by IT specialists to understand business operations and requirements, in order to build BI applications and help formulate queries. In this paper, we present a vision for BI that is driven by enterprise modeling. The Business Intelligence Model (BIM) aims to enable business users to conceptualize business operations and strategies and performance indicators in a way that can be connected to enterprise data through highly automated tools. The BIM draws upon well-established business practices such as Balanced Scorecard and Strategy Maps as well as requirements and conceptual modeling techniques such as goal modeling. The connection from BIM to databases is supported by a complementary research effort on conceptual data integration.

Keywords: Business Intelligence, Key Performance Indicators, Strategic Planning, Analytics, Enterprise Modeling, Conceptual Modeling.

1 Introduction

In all kinds of enterprises, from businesses to government to healthcare, data is becoming increasingly abundant. As more and more operations are conducted or supported digitally, massive amounts of data can be collected and analyzed. Organizations are taking advantage of computational capabilities to slice and dice the data, pose ad hoc queries, detect patterns, and to measure performance. The vision of the data-driven enterprise holds promise for greater strategic agility and operational efficiency [1], supported by a range of software tools under the general label of Business Intelligence (BI).

Yet the benefits of BI can be elusive. Despite the availability of voluminous data, meaningful and productive use of that data remains a major hurdle for BI initiatives. Data exists throughout the enterprise to serve numerous different

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 31–45, 2010.

purposes, and have diverse semantics and representations. Much of the IT implementations of business operations are not directly suitable or comprehensible for enterprise level decision making. There is a huge conceptual distance between business thinking and decision making on the one hand, and the raw data that is the lifeblood of daily operations on the other. BI initiatives therefore can be very costly, take many months, require serious commitment from business stakeholders and IT personnel, and still produce results that are of uncertain benefits.

We argue that the benefits of BI and the data-driven enterprise can be more easily attained by constructing a smoother path between business thinking and IT implementation. The core of this vision is a conceptual model for representing a business viewpoint of data. Business decision makers do not want to think in terms of tuples in databases, or dimensions in star schemas. They think in terms of customer satisfaction, market share, opportunities and threats, and how to rearrange business processes. These concepts then need to be mapped to IT implementations in a coherent and effective way that minimizes manual effort.

We propose a Business Intelligence Model (BIM) that draws upon well-established concepts and practices in the business community, such as the Balanced Scorecard and Strategy Maps, as well as techniques from conceptual modeling and enterprise modeling, such as metamodeling and goal modeling techniques.

The BIM will be used by business users to build a business schema of their strategies and operations and performance measures. Users can therefore query this business schema using familiar business concepts, to perform analysis on enterprise data, to track decisions and their impacts, or to explore alternate strategies for addressing problems. The business queries are translated through schema mappings into queries defined over databases and data warehouses, and the answers will be translated back into business-level concepts.

The BIM is the foundation for the broader research agenda of the Business Intelligence Network (BIN)[1], which aims to raise the level of abstraction for the next generation of BI tools, so that the benefits of BI will be accessible to all members of the enterprise, with minimal help from specialist intermediaries. The BIN research project is supported by BI industry leaders.

A case study to test the BIM in a real world setting is being conducted at a hospital currently engaged in a BI initiative. In this paper, we outline the key features of the BIM using a hypothetical business setting, loosely based on and extending an example from the BSC Institute[2]. Details of the BIM can be found in [2].

Section 2 of this paper describes the BestTech case study. Section 3 introduces the main features of BIM and its metamodel. Section 4 presents how to use BIM for strategic planning while, in Section 5, its application for operations management. In Section 6, we illustrate analytic queries for the example enterprise setting. Sections 7 and 8 discuss, respectively, related work and conclusions.

---

[1] http://bin.cs.toronto.edu/home/index.php and http://www.nserc-crsng.gc.ca/Partners-Partenaires/Networks-Reseaux/BIN-RVE_eng.asp

[2] Balanced Scorecard Institute http://www.balancedscorecard.org (2010).

2 An Illustrative Enterprise Setting: BestTech Inc.

BestTech Inc. is a fictitious Canadian specialty retailer and e-tailer of consumer electronics, personal computers and entertainment software and maintains a 24 hour computer support task force. BestTech Inc. offers consumers a unique shopping experience with the latest technology and entertainment products, at the right price, with a no-pressure (non-commissioned) sales environment.

In its *strategic planning process* [3], BestTech identifies as its strategic goals increased *profitability* and *visibility* in the Canadian market expanding also into Europe. To achieve these goals, it intends to improve its brand image investing in marketing campaigns but also improving its distribution infrastructure and the quality of service provided to customers. In particular, BestTech needs to overcome the bad reputation it had developed in the Internet community for damages and delays in the products delivered to customers.

BestTech wants to be aware of threats and opportunities in the market and how such situations can influence its business. Moreover, since BestTech cannot manage and control what it cannot measure, it desires to have a clear representation of its operational layer to monitor the organization's performance with real-time data.

Indeed, the executive board wants to communicate its strategies to middle management and frontline workers, and share and monitor performance indicators at all levels, facilitating greater collaboration and coordination among business units and individual employees.

BestTech seeks advanced methods and tools to help conceptualize its business strategies and operations, and to perform analytics on its enterprise data to detect problems, allocate resources efficiently, and make better decisions.

3 The Business Intelligence Model (BIM)

The Business Intelligence Model allows business users to conceptualize their business operations and strategies using concepts that are familiar to them, including: Actor, Directive, Intention, Event, Situation, Indicator, Influence, Process, Resource, and Domain Assumption. These concepts (and their semantics) are synthesized from business and conceptual modeling sources. For example, strategy concepts draw upon the Balanced Scorecard and Strategy Maps [4,5], combined with intentional and social concepts from goal-oriented requirements engineering, notably [6,7,8]. The notion of influence is adopted from influence diagrams [9], a well-known and accepted decision analysis technique. SWOT analysis concepts [10] (strengths, weaknesses, opportunities, and threats) and others have been adopted from OMG's Business Motivation Model standard [11]. The concepts are formalized through metamodeling in terms of abstract concepts such as Thing, Object, Proposition, Entity, and Relationship, taking inspiration from DOLCE [12]. Abstraction mechanisms, such as generalization, aggregation and classification are also provided. Full details can be found in [2].

While the BIM by itself can facilitate understanding of the enterprise, the more fundamental aim, in the context of BI, is to provide a business-friendly way to exploit the vast amounts of data collected by the enterprise. The BIM works together with advanced conceptual data integration technology currently under development, jointly within the BIN business intelligence research project.

In particular, indicators in BIM are connected[3] to enterprise databases or data warehouses through the CIM – the Conceptual Integration Model. CIM provides access to multi-dimensional data through a high-level conceptual model. Mappings are defined so that each construct in the conceptual model is associated with a query on the physical model. At design time, a business analyst would specify in the CIM what information is needed and in what form, so that the system could respond to business user queries in terms of BIM concepts at run-time. The CIM is detailed in [13].

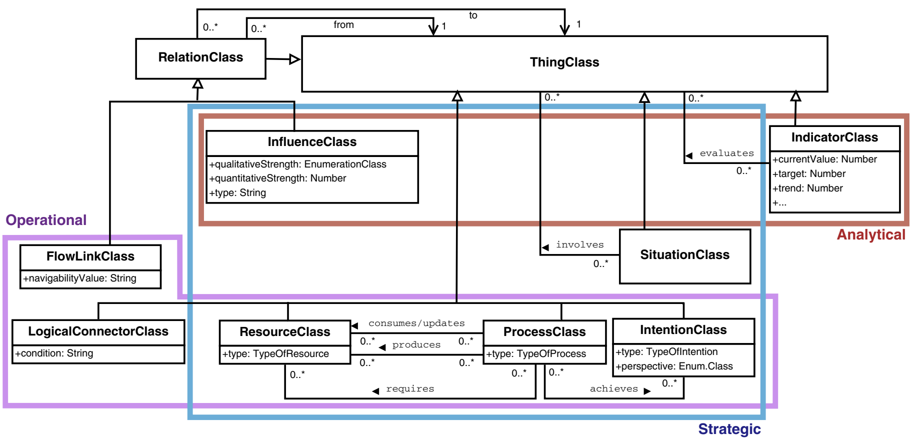

Fig. 1. The BIM fragment which provides Strategic, Operational and Analytic primitives. The "type" attributes are used to represent different business terminology. For example, the type attribute for "ProcessClass" can assume the values: *Initiative, Project, Action, Activity and Task.*

Figure 1 shows the main elements of BIM illustrated in this paper. Three groups of concepts work in concert – strategic, operational, and analytic. Strategic analysis drives analytical BI, while results from analytics direct the focus of operational initiatives, as suggested in [14].

We illustrate how BestTech can use BIM to address strategy, operations, and analytics in the following sections.

4 Modelling and Reasoning about Strategies

To support strategic reasoning, BIM provides constructs for modeling hierarchical goal structures, with alternatives and subgoals and actions for achieving

[3] A semi-automatic approach is under development within BIN to address such an issue.

them. Key performance indicators (KPIs) can be associated with goals at any level. Internal and external environmental factors are modeled as situations, reflecting internal *strengths* and *weaknesses*, and external *threats* and *opportunities*. Resources are allocated to initiatives and processes according to the chosen strategies. We illustrate these aspects using the BestTech example.

4.1 Hierarchy of Goals, Actions and Key Performance Indicators

Figure 2[4] provides a graphical representation of BestTech's strategic plan expressed in BIM. It shows how BestTech translates its vision and strategies into actions, and the one or more KPIs chosen to measure performance towards each of its *strategic goals*[5].

For example, the "Brand image improved" strategic goal is pursued through the "Expand into Europe" initiative, and is monitored by the KPI "Brand awareness score". It has positive influences on the financially-oriented goal of "Revenues increased", and the customer-focused goal of "Market share increased".

Following [4,5], the overall strategic plan is balanced along the four perspectives of *Financial*, *Customer*, *Internal Process* and *Learning & Growth*.

Given this representation of a strategy plan, one is able to perform analysis on possible goals conflicts or to evaluate the satisfaction level of alternative (sub)strategies. BIM provides a mechanism for forward and backward goal reasoning adopted from [15].

4.2 SWOT Situational Analysis

Recognition of strengths, weaknesses, opportunities, and threats is essential to strategic management. Toyota's recall of 9 million vehicles due to sudden unintended acceleration or steering problems was a *weakness* for Toyota that led to its worst ranking in the annual J.D. Power quality survey [16]. This situation led Toyota to adopt a conversion strategy from "Selling more cars" to "Quality of service and customers assistance increased".

BIM models strengths, weaknesses, opportunities, and threats as relations between the primitive constructs *Situation* and *Intention*. Figure 3 shows an example in which a market vacated by a competitor can raise the probability of success for BestTech to increase its market share.

SWOT analysis can help to select among alternative strategies, and to determine their viability. Competitive advantage can be recognized by matching strengths to opportunities. A conversion strategy would convert weaknesses or threats into strengths or opportunities.

---

[4] The graphical representations provided in this paper are not intended as end-user visualization but for the description and illustration of BIM 's features and functionalities.

[5] The term *Strategic goal* is one of the values which can be assumed by the type attribute for the "IntentionClass" in Figure 1. Depending on the context, such attribute can assume other values such as: *Tactical goal*, *Operational goal*, *Soft goal*, etc. We refer to such terms with the general term Intention.

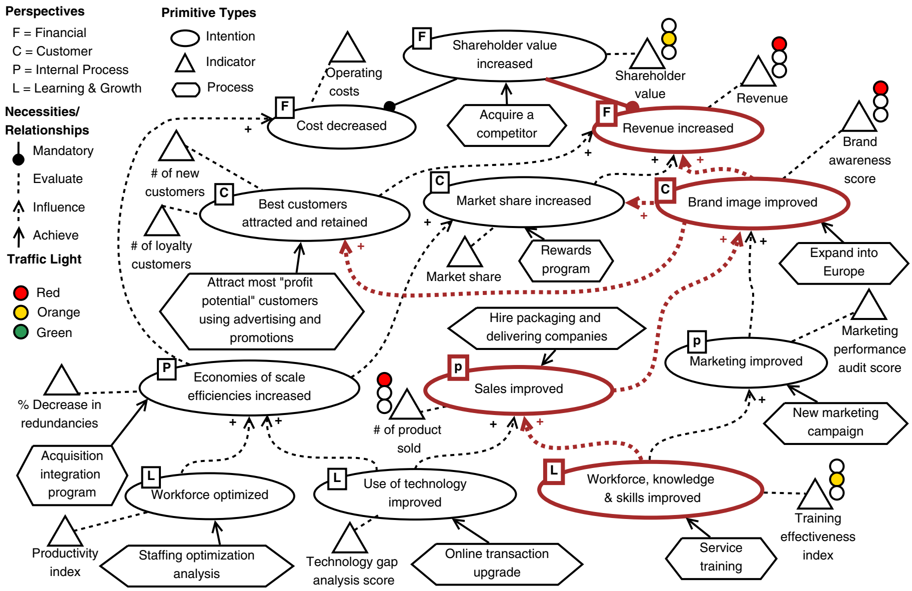

Fig. 2. The BestTech strategy plan, including Financial, Customer, Process, and Learning & Growth perspectives. One of the possible sub-strategies to increase revenue is highlighted in thicker red lines.

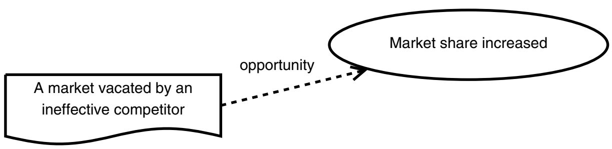

Fig. 3. An opportunity for BestTech to increase its market share

4.3 Allocation and Monitoring of Resources

Resource allocation is a fundamental aspect in a strategic planning process since action plans and initiatives rely on available resources, e.g., *human resources*. Management constantly needs to make decisions about what initiatives to fund or not to fund, and at what levels. Figure 4 shows an example in which a *monetary resource*, namely "Investment on advertising and promotions", is associated with the "Attract high-profit-potential customers with advertising and promotions" initiative.

The "Total investment for advertising" KPI is used to monitor the actual amount of money consumed by the initiative. The KPI target ($6,000) represents the level of funding assigned by the executive board. In this case, BestTech has exceeded the budget already with expenses at $ 7,200, while at the same time, the number of customers attracted (8,000) has surpassed the desired target of 6,000. (Figure 4)

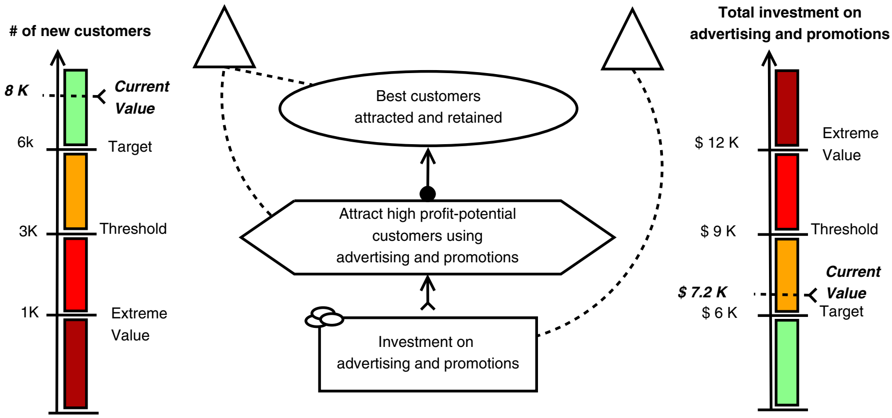

Fig. 4. An example of resource allocation and monitoring

In general, since resources are consumed over time and are (usually) limited, KPIs can be defined on them to monitor their availability and consumption at strategic and operation levels.

4.4 Business Schema

The complete set of goals, objectives, situations, processes, resources, etc., and relationships among them, e.g., strengths, threats, etc., constitutes a *business schema*, which can acquire instance data through the CIM framework introduced in Section 3. An example of such a schema for BestTech is provided in [2].

A business schema is a valuable resource for an organization since, besides providing a big picture of the organization and the business environment in which it operates, allows a number of different kinds of analyses to be performed:

– *forward or backward goal analysis* [15], to reason about conflicts and contributions among goals using the *influence* relationships (positive, negative, qualitative strength, etc.);
– in-depth *situation analysis*, to evaluate those situations which help or hurt the strategies of an organization. In particular, opportunities, threats, weaknesses, strengths are identified to take remedial actions or to set higher target values;
– *consistency check analysis*, to verify that each goal has associated an action (for its achievement) and/or a KPI (for its monitoring);
– *balanced strategy analysis*, to assure whether the overall strategy is well distributed among all the four Balanced Scorecard perspectives or unbalanced toward a specific one;
– *resource analysis*, to evaluate resource consumption and to optimize use of resources, relying on a global overview of their allocation;

These kinds of analyses can be used to support long-term analysis on high-level strategic goals and decisions, as well as shorter term objectives and targets and day-to-day operations, as presented in the next section. The specific analysis techniques are discussed in more detail and illustrated in [2].

5 Modelling and Reasoning about Operations

Operations management needs to ensure that business operations are *efficient* in resource usage, e.g., cost per unit for delivery, and *effective* in meeting customer requirements, e.g., quality of delivery. As described in Section 2, operational goals need to be related to strategic goals. BestTech desires to increase its profit (G1) by, among others, improving its brand image (G4) in the Internet community (the blue top layer in Figure 5).

To achieve these strategic goals, BestTech intends to adopt two approaches at the operational level (bottom purple layer in Figure 5): i) reduce delays in the delivery[6] of products (G14 but also G16) and ii) decrease the probability of defects or damages in the products delivered (G15 and G17). The latter will also help to reduce the number of products returned (related to G9) by increasing at the same time the effective number of products sold (G5).

Indeed, the satisfaction of operational goals (G12-G17) will be propagated to middle goals (middle layer in dark red, G7-G11), such as "(G8) Online sales process improved" and "(G9) Customer satisfaction maximized" which, in turn, will improve the brand image (G4) by avoiding delays, damages and defects in the products delivered. This view of the BIM shows the alignment of the operational layer towards the achievement of strategic objectives.

To monitor and analyze the efficiency and effectiveness of the *Online sales process*, the BIM allows BestTech to define a global view of its workflow (using concepts[7] from the operational group in Figure 1). Figure 6 shows such a workflow in terms of activities and resources produced or consumed, which can be summarized as follows:

A customer makes an on-line order which is accepted or rejected depending on the availability of the products in the inventory. After the payment is performed (in the figure we skipped this for sake of simplicity) the order is processed by the packaging activity which withdraws from the storage the list of products contained in the order. Finally, the package is delivered to the customer.

To analyze the performance of the described process and its impact on the satisfaction of operational and strategic goals, a set of KPIs are defined on the workflow's activities and resources.

These KPIs and the relationships among them constitute what we call a **Indicators Graph** (IG), which is shown in Figure 7.

[6] In Figure 5, the "Delivery (lead) time" is the time between the creation of the order and the receipt of the order.

[7] BIM provides a "light" modeling for business process which can also be used at strategic level. Moreover, referring to the five perspectives presented in [17], BIM aims to cover the *Business Process Context, Informational, and Organizational Perspectives* while other well-known models, e.g., the Business Process Management Notation [18] focus more on the *Behavioural and Functional perspectives.*

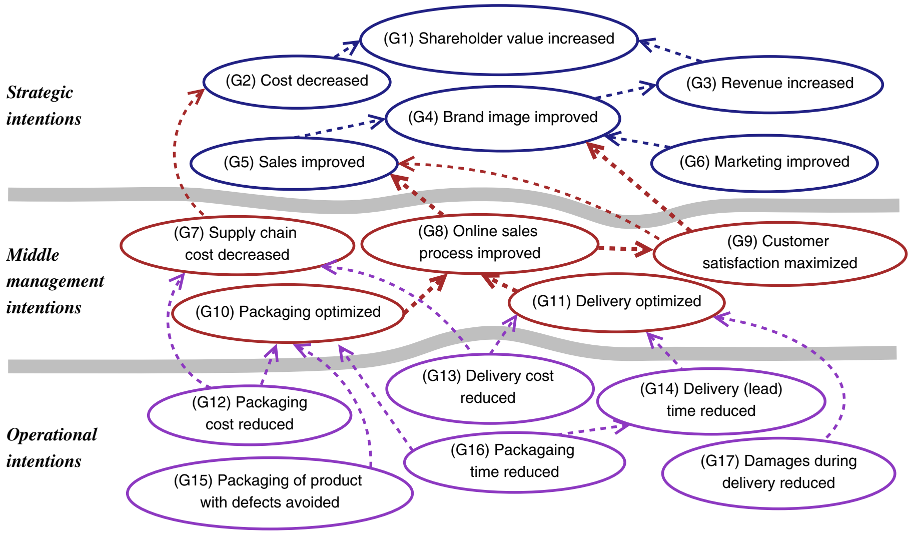

Fig. 5. Strategic, Mid-Level, and Operational intentions leading to increased shareholder value (partial view)

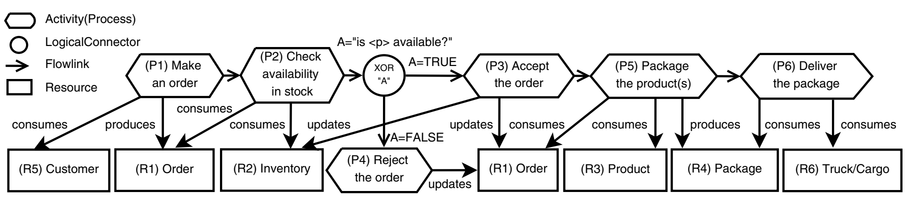

Fig. 6. The Online sales process workflow

In the graph, the cause-effect relationships can have two meanings:

– **deterministic**, the metric used to evaluate the *influencee* is defined as a function of (the metrics of) the *influencers*, e.g., the "Shareholder value" is calculated by the "Revenue" minus the "Operating costs";
– **probabilistic**, the *influencee* depends on the *influencers* through a probabilistic relationship, e.g., "# of stock available at customer first request" is influenced by "Size of safety stock"[8] in a probabilistic manner; which means that, a high value for "Size of safety stock" raises the probability of a high value for "# of stock available at customer first request".

In this paper, we limit the exposition to the **qualitative**[9] representation of such cause-effect relationships (as depicted in Figure 7); however, as described in

[8] Safety stock is a term used to describe a level of stock that is maintained below the cycle stock to buffer against stock-outs.
[9] i.e., the definition of causal dependency arcs among different indicators.

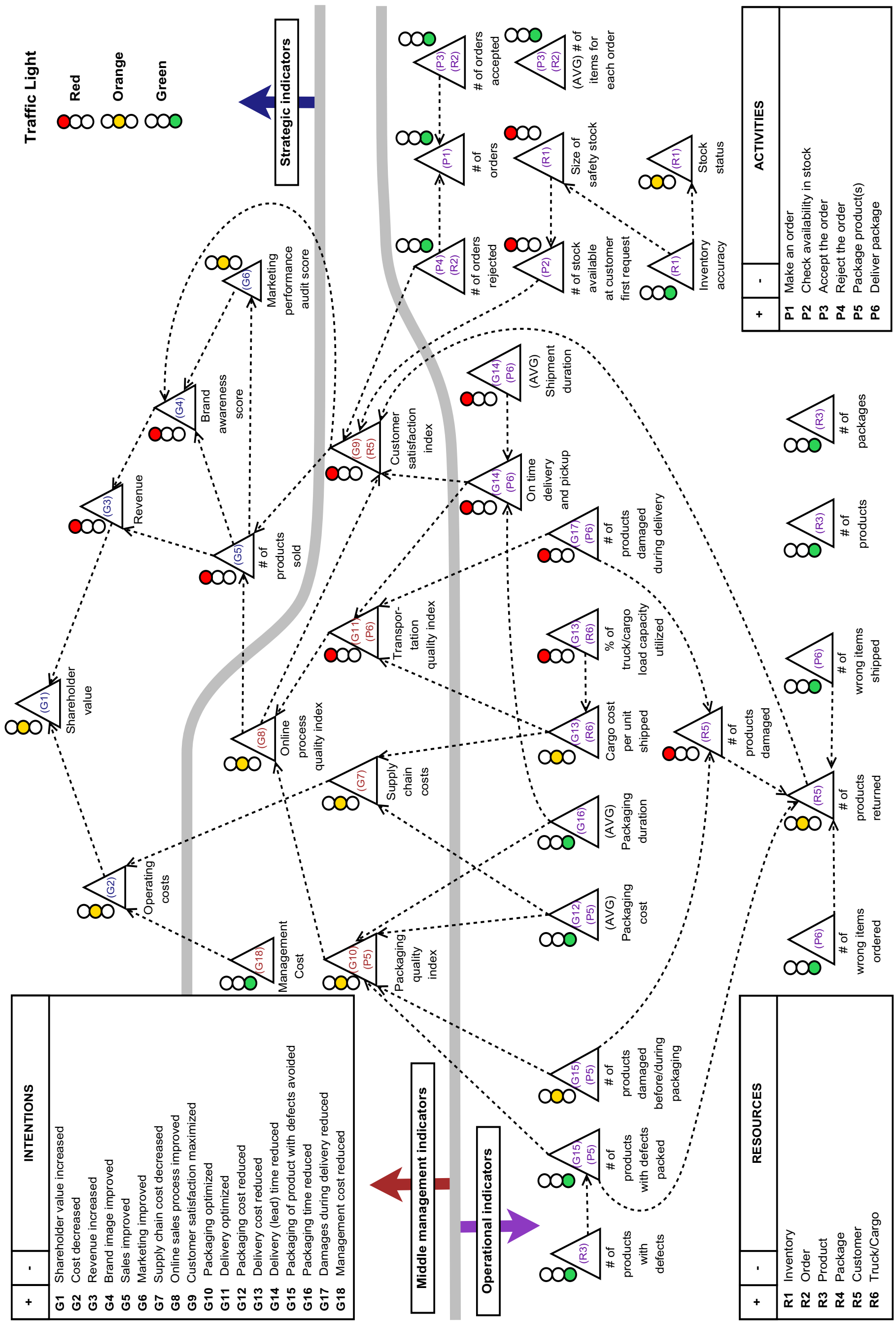

Fig. 7. BestTech's Indicators Graph

Section 8, our final goal is to **quantify**[10] such cause-effect relationships defining also a degree of *soundness* and *completeness* of the Indicators Graph with respect to the specific context.

We recall that, as described in Section 3, we translate the KPIs illustrated in Figure 7 into the CIM 's conceptual model which in turn is mapped to the physical model of enterprise data in data warehouses or databases.

6 Analytics and User Profiles

As described in previous sections, the main goal of strategic planning is to drive the performance of the company as a whole, by enabling senior executives to collaborate on and agree to corporate strategies and by facilitating the sharing of those goals with middle management and frontline workers. This approach sets the foundation for performance management in the form of KPIs which spans the organization from the strategic to the operational.

The Indicators Graph in Figure 7 is an example of such a foundation; it is also a useful input for *Analytic* activities [1] which are used to identify those aspects of the business which need to be further analyzed. Indeed, analytics help to investigate (from many different angles) those factors that have a high impact on business performance by determining the location or cause of major problems.

For example, analytics can help to answer the following questions: *"if profits are declining, is it because of low sales, or increasing expenses? if customer churn rates are on the rise, is it because of poor product quality, or lack of success in customer loyalty initiatives?"* [14].

BIM enables and supports such analytical activities by providing an underlying *performance framework* (see the Analytic box in Figure 1) which is tied to the strategies and operations of the organization. Moreover, since BI is intended to be used by employees at all levels, BIM helps to define ad-hoc *Analytic User Profiles*[11] for different employees.

Examples of profiles and analytics queries for BestTech are the following:

CEO Analytic User Profile:

Q1. *Where are the cost pressures and what the most probable causes?* A CEO can read the "Operating costs" indicator that can be refined into "Management costs" and "Supply chain costs", among others. Figure 7 indicates to the CEO that the latter is the actual problem. At this point the CEO could take a strategic decision adopting cost-cutting measures, but this may exacerbate the already "low" quality of service. Another possibility is to request the manager who is responsible for the "(G7) Supply chain cost decreased" goal to discuss about a possible solution at the operational level (see Q4 below).

[10] A careful reader can observe that there is a main issue in evaluating together deterministic and probabilistic information; this is part of our future research.

[11] A profile is defined by the set of indicators associated with the *Intentions* (e.g., strategic goals, operational goals, etc.) that an actor is responsible for.

Q2. *What is the status of the brand image? Why is it suffering?* The "Brand awareness score" provides such information revealing a difficult situation for BestTech. By refining such information, the CEO discovers that the "Marketing performance audit score" has an orange color, influenced by a red in "# of products sold". Before undertaking any remedial action, the CEO decides to request the middle management to explain why BestTech is not achieving its target in selling products (see Q3 below).

Sales Manager Analytic User Profile:

Q3. *Are we reaching the target in selling products? If not, why not?* The "# of products sold" provides a negative answer. The cause is due to the "Online process quality index" as well as the "Customer satisfaction index". The Sales manager further investigates through the analytic queries Q4 and G5.

Q4. *What are the major issues in the Online sales process?* This question addresses the CEO's investigation for both aspects of efficiency (costs analyzed in Q1) and effectiveness (benefits in Q2 and Q3). In term of effectiveness, Figure 7 shows that "Shipment duration" is too high, causing a red value for "On time delivery and pickup". Moreover, a red value for "# of products damaged during the delivery" represents another issue of effectiveness. In terms of efficiency, a less than optimal "% of truck/cargo load capacity utilized" leads to high costs.

Q5. *Are customers satisfied? if not, why?* Analyzing the "Customer satisfaction index" the manager discovers that different causes exist. First there is delay in the delivery of products (analyzed in Q4); then there is unavailability of stocks at customer's first request (due to a bad organization of the safety stock) and an high "# of products returned" (see Q6).

Q6. *What is the actual number of products returned? What is the most probable cause of it?* The Indicator "# of products returned" is used. Figure 7 shows that the main cause is the "# of products damaged" and, in particular, those products damaged during the delivery (as shown by the "# of products damaged during delivery" indicator). Reducing such issues, will help the manager with the "Sales improved" goal for which he/she is responsible for.

Based on these analytic results, the management team can use the BIM to explore new strategies, and to make trade-offs among competing alternatives. For example, the team may modify the BestTech business schema to include a strategy that outsources delivery to a more reliable delivery company to optimize cargo loads and to reduce damages in shipping. This initiative will reduce the cost of delivery and, at the same time, increase sales and customer satisfaction. According to the business schema this will ultimately improve BestTech's image and profit.

Unlike in current BI practice, these analytics are supported by an explicit enterprise model that supports reasoning about business strategies and operations, with direct connections to actual enterprise data.

7 Discussion and Related Work

Enterprise modeling techniques have been used to help understand business operations and processes, and to lead to the development of IT systems (e.g., [19]).

The Business Intelligence Model aims to extend enterprise modeling to provide business users with more direct access to enterprise data through enterprise models, so that the data can be interpreted and analyzed in terms of familiar business concepts, enabling timely and effective decision making and action.

Modeling techniques in information systems, including most data and process modeling languages as well as UML, focus primarily on static and dynamic ontologies, but not the intentional or social ontologies (with concepts such as actors, goals, or objectives) that are needed for business reasoning [20].

The Zachman framework [21] has long pointed to the need to include motivation ("Column 6") in enterprise modeling, though few modeling techniques have addressed this need specifically. A proposal to include intentional social modeling in enterprise architecture modeling, based on the i* framework, was described in [22].

Most enterprise architecture frameworks today include performance indicators (e.g., the Performance Reference Model in the Federal Enterprise Architecture [23]), and can benefit from more powerful modeling techniques and tools that support business reasoning with connection to enterprise-wide data.

Recent work has incorporated goal modeling in design methodologies for data warehousing [24]. The BIM proposal aims to provide business users query facilities for reasoning about business strategies and operations, with analysis on the data accessible via mappings to databases and data warehouses.

Among recent enterprise modeling approaches, BMM [11] is closest in spirit to BIM. In BIM, concepts adopted from BMM and other sources are placed on an ontological foundation based on DOLCE [12], and integrated with state-of-the-art abstraction mechanisms.

Some BI tools are beginning to include representations of strategies (e.g., [14]), but provide little or no reasoning support.

Other work has also extended i* [7] and related frameworks (e.g., URN [8]) towards enterprise and business modeling, e.g., [25,26]. A recent extension of URN includes indicators [27]. Strategy Maps are modeled in [28] using a modified version of i*.

The BIM aims to unify these various modeling concepts into a coherent framework with reasoning support and connection to enterprise data, built upon a firm conceptual modeling foundation.

8 Conclusions

We have articulated a vision for the next generation of business intelligence in which enterprise modeling provides the foundation for business users to have more direct access and control over enterprise data, their analyses and meaningful interpretation, by using familiar business concepts. The approach aims to address concerns that current BI solutions are costly to develop, requiring significant IT involvement, and are therefore reaching only a small segment of the potential user population – those who are technology savvy. The proposed approach combines the use of familiar business concepts with well founded modeling technologies, as well as mapping technologies to link to databases. Work is underway to test the BIM with the CIO and executive team at a hospital which is currently undergoing a BI initiative. As another line of future work, we are planning to extend the BIM to incorporate uncertainty in strategic modeling and analysis through the use of Bayesian networks [29] in the Indicators Graph. This will enable BIM to support statistical decision making [30] and will complement the logic-based analysis techniques currently within BIM's scope.

Acknowledgments. This work was supported by the Business Intelligence Network (BIN) and the Natural Sciences and Engineering Research Council of Canada. We are grateful to D. Amyot, I. Kiringa, F. Rizzolo and many others for useful discussions.

References

1. Davenport, T.H., Harris, J.G.: Competing on Analytics: The New Science of Winning. Harvard Business School Press, Boston (2007)
2. Barone, D., Mylopoulos, J., Jiang, L., Amyot, D.: Business Intelligence Model, version 1.0. Technical Report CSRG-607, ftp://ftp.cs.toronto.edu/csri-technical-reports/INDEX.html University of Toronto (March 2010)
3. Saxena, P.K.: Principles of Management: A Modern Approach. Global India Publications Pvt Ltd (2009)
4. Kaplan, R.S., Norton, D.P.: Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
5. Kaplan, R.S., Norton, D.P.: Strategy maps: Converting intangible assets into tangible outcomes. Harvard Business School Press, Boston (2004)
6. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program 20(1-2), 3–50 (1993)
7. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proc. 3rd IEEE Int. Symp. on Requirements Engineering, Washington, USA (1997)
8. International Telecommunication Union: Recommendation Z.151 (11/08): User Requirements Notation (URN) – Language definition, http://www.itu.int/rec/T-REC-Z.151/en
9. Howard, R., Matheson, J.: Influence diagrams. Readings on the Principles and Applications of Decision Analysis II (1984)
10. Dealtry, T.R.: Dynamic SWOT Analysis. Dynamic SWOT Associates (1994)
11. Business Rules Group: The Business Motivation Model: Business Governance in a Volatile World (2007), http://www.businessrulesgroup.org/bmm.shtml Ver. 1.3.
12. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 223–233. Springer, Heidelberg (2002)

13. Rizzolo, F., Kiringa, I., Pottinger, R., Wong, K.: The conceptual integration modeling framework: Abstracting from the multidimensional model. Technical report, SITE, University of Ottawa (2010)
14. Quinn, K.: How Business Intelligence should work. Information Builders White Papers (2009)
15. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-oriented requirements analysis and reasoning in the Tropos methodology. Eng. App. Artif. Intel. 18, 159–171 (2005)
16. BusinessWeek: Toyota drops in auto-quality survey; Ford makes top 5 (update1) (June 2010), http://www.businessweek.com/
17. List, B., Korherr, B.: An evaluation of conceptual business process modelling languages. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 1532–1539. Springer, Heidelberg (2007)
18. Business Process Management Initiative: Business process management notation. Specification Version 2.0 (June 2010), http://www.bpmn.org/
19. Stirna, J., Persson, A.: Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice. In: Proceedings of the Workshop on EMMSAD 2007, held in conjunction with the 19th CAiSE 2007 (2007)
20. Mylopoulos, J.: Information modeling in the time of the revolution. Inf. Syst. 23(3-4), 127–155 (1998)
21. Zachman, J.A.: A framework for information system architecture. IBM Systems Journal 26(3), 277–293 (1987)
22. Yu, E., Strohmaier, M., Deng, X.: Exploring intentional modeling and analysis for enterprise architecture. In: Proceedings of the EDOC Workshop on Trends in Enterprise Architecture Research (October 2006)
23. Office of Management and Budget: FEA Practice Guide. (November 2007), http://www.whitehouse.gov/omb/asset.aspx?AssetId=471
24. Mazón, J.N., Trujillo, J.: A hybrid model driven development framework for the multidimensional modeling of data warehouses. SIGMOD Record 38(2) (2009)
25. Bleistein, S.J., Cox, K., Verner, J.M., Phalp, K.: B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. Information & Software Technology 48(9), 846–868 (2006)
26. Andersson, B., Johannesson, P., Zdravkovic, J.: Aligning goals and services through goal and business modelling. Inf. Syst. E-Business Management 7(2) (2009)
27. Pourshahid, A., Amyot, D., Peyton, L., Ghanavati, S., Chen, P., Weiss, M., Forster, A.J.: Business process management with the User Requirements Notation. Electronic Commerce Research 9(4), 269–316 (2009)
28. Babar, A., Zowghi, D., Chew, E.: Using goals to model strategy map for business IT alignment. In: Proceedings of the Workshop on BUSITAL 2010, held in conjunctiun with the 22th CAiSE (2010)
29. Barone, D., Stella, F., Batini, C.: Dependency discovery in data quality. In: Pernici, B. (ed.) Advanced Information Systems Engineering. LNCS, vol. 6051, pp. 53–67. Springer, Heidelberg (2010)
30. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)

Business Modeling Experience for a State Pension Voluntary Insurance Case

Alcedo Coenen[1], Bas van Gils[2], Charlotte Bouvy[3], Roel Kerkhofs[3], and Sander Meijer[4]

[1] Sociale Verzekeringsbank, Amstelveen, The Netherlands
acoenen@svb.nl
[2] BiZZdesign, Enschede, The Netherlands
b.vangils@bizzdesign.nl
[3] O&i - Partners in BPM, Utrecht, The Netherlands
{c.bouvy,r.kerkhofs}@oi.nl
[4] Verdonck, Klooster & Associates, Zoetermeer, The Netherlands
sander.meijer@vka.nl

Abstract. The Dutch governmental Sociale Verzekeringsbank (SVB) recently completed a project on business modeling. The aim was to develop business models that express the strategic goals of customer event orientation, integrated customer services as well as improved agility. An approach was chosen to combine event definition, dynamic process modeling and business rules. In this paper we present our findings. Main conclusion is that, although we used some best practices and common techniques, we had to invent our own ways to combine event definitions, dynamic processes, object models and business rules.

Keywords: Life event, Service Orientation, Business Rule, Dynamic Processes, Governing Process.

1 Introduction

1.1 Situation

The Sociale Verzekeringsbank (SVB) is the organization that implements national insurance schemes in the Netherlands. The SVB takes care of paying child benefit, state pension and other related social insurance payments for Dutch citizens, living inside or outside the Netherlands. The SVB has been doing that for more than a century, by order of the government, and counts some 4.9 million clients.

The organization has about 3500 employees, has implemented about 15 legal regulations, pays about 33 billion euros per year and has a strategy that is directed towards servicing the Dutch citizen excellently. Customer service is characterized by a combination of operational excellence and customer intimacy, and is organized in three main forms:

- Fully automated payment services; the customer receives the benefit without the need to apply for it.
- Self service via internet (www.svb.nl); intelligent dialogues guide the customer through the regulations.
- Service teams, supporting the customer at local desks.

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 46–60, 2010.

1.2 Challenges

The SVB is continuously working on the improvement of its service. Some main strategic subjects are:

- **Customer event orientation and integrated customer services:** making the service as complete as possible from the perspective of the customer, instead of the perspective of the regulation. When e.g. a customer informs the SVB about a move abroad, the service includes all regulations that are affected by this move.
- **Improving agility:** making IT even more flexible and agile than it already is, in order to be able to adapt the execution of existing regulations easily, or to start executing new regulations within a small time window.

In order to achieve these strategic goals, the SVB has formulated a target-architecture with business, application and infrastructural elements. The architecture contains three main notions:

- **Service orientation:** Service orientation is chosen in order to achieve loosely coupled services on a business and application level, which enables the agility targets.
- **Life events:** Life events are considered to be the unit of work. This notion implements the customer event orientation.
- **Business rules:** Business rules are meant to separate concerns between business logic and technical infrastructure, in order to achieve agility. It is also meant to diminish the complexity of process models, by regulating the process dynamically through rules.

1.3 Business Modeling Project

In 2009 the SVB started a business modeling project with the main purpose to prepare for the implementation of new, rule-based technology, based on the target architecture. The project had three main objectives:

1. The project was meant to model life event orientation.
2. The project was meant to model a generic process that is able to deal with all products and services in an integrated way, in order to support the purpose of integrated customer services.
3. The project was meant to translate legislation to business rules. The scope of the project was limited to a specific business domain: the voluntary insurance for state pension in case of a move abroad[1]. The legislation is relatively complex, and has resulted in more than 250 business rules.

[1] Dutch citizens have state pension insurance by law, leading to a state pension from the age of 65. Once a Dutch citizen moves abroad the insurance stops, which leads to a lower state pension at 65. Dutch law offers the possibility of filling this insurance gap by allowing a Dutch citizen living abroad to pay the premium for the insurance voluntarily.

The project results were targeted at a series of models; most notably a List of customer life events (also: life events), a set of business rules, a process model, a business vocabulary, an object model, and other (non-functional) requirements.

1.4 Research Approach and Overview

The article reports how we approached business modeling. There was no a-priori research question, but a result driven mandate to create business models. The findings in this paper are therefore not the result of a structured research methodology. Experience papers like this one are valuable to practitioners and scientists, even if no structured research methodology is followed. This is shown by papers like [2] and [3].

The remaining of the article starts with paragraph 2 explaining the basic approach and concepts, paragraph 3 illustrates some issues that the project had to deal with. The last paragraph 4 contains some conclusions and invitations for research.

2 Basic Approach and Concepts

2.1 Basic Concepts

The business modeling project used the following basic concepts and their interrelations. They can be considered to be the basic ontology for business modeling. A Customer life event (short: event) is the unit of work for the SVB from the perspective of the customer (citizen), it is where it all starts. Events trigger governing processes. The Governing process (see Paragraph 2.5) is the process that orchestrates other business processes:

– Governing processes orchestrate business processes
– Both governing processes and business processes use business rules

A Business process is a unit of work internally. Business processes use, change or create business objects and use business rules (expressing the decisions and calculations being made in those business processes). A business rule represents desired or mandatory behavior of the organization (see Paragraph 2.3). Business rules restrict or constrain the relations between business objects, which are the concepts and artifacts that are managed by the SVB.

2.2 Life Events

Life events "describe situations of human beings where public services may be required" [6]. Life events focus on a citizen perspective and can be seen as a metaphor for identifying public services related to specific situations that citizens face, rather than the execution of specific (parts of) legislation. Examples of life events are death, emigration, marriage, founding of a company, hire employees. Using life events fitted the goals of the SVB (Paragraph 1.2) with respect to

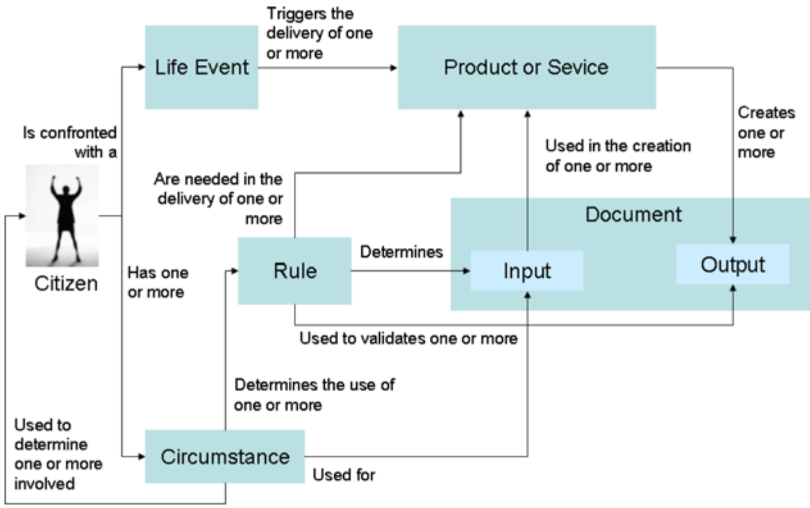

Fig. 1. Ontology guiding modeling effort

integrated customer service. A life event may trigger more than one service. To structure and guide the modeling effort, the ontology outlined in Figure 1 was used, based on [11].

As the figure shows a Citizen can be confronted with one or more life events and one or more circumstances. The circumstances are needed to determine the legislation (coded in rules) that are needed in the delivery of the product/service. Circumstances can also be used for the creation of products or services. Finally the circumstances can be used to determine other citizens, e.g. relatives, that are involved in the same life event. In other words, a life event triggers the delivery of services which are realized by means of processes which rely on business rules. The situations in which a group of citizens, mainly families, is involved (e.g., a whole family moves abroad) is special since it requires service delivery in parallel. We will discuss the consequence of this issue in more detail in Paragraph 2.5. The following is an example of the definition of the life event "moving within the Netherlands":

Example 1 (A life event):

life event -
Moving in the Netherlands

Short description -
One or more related persons living in the Netherlands, are going to live at another address within the Netherlands

Explanation -
This life event is used for moving within the Netherlands. It includes the relocation of a child or a roommate. This life event is not applicable for

temporarily residence elsewhere. Also the change of the correspondence address is excluded from this life event

Circumstances -
(a) There is a new residential address, (b) Current residence = Netherlands, (c)New residence = Netherlands, (d) Residential address = Address = Correspondence Address , (e) Residence time = undetermined, (f) Location != nursing home, prison or other institution

Related products and services [2] -
(1) AOW Pension (2) Child support (3) Anw benefit (4) TOG allowance (5) Voluntary insurance (6) AIO supplement

◇

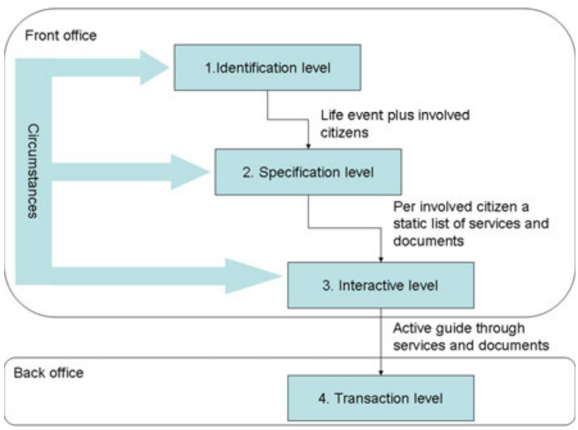

Fig. 2. Life events in relation to governing process

On the basis of this meta model life events can be modeled. When using life events in real situations, there are four different levels of abstraction from the identification of the life event that is relevant to the actual service delivery [10]. We adapted that model to our specific situation and started using this model with four abstraction levels. The adapted model with the four abstraction levels are presented in Figure 2. The relation with the governing process is explained in Paragraph 2.5. The first three levels take place in the front office. In the first level the relevant life event is selected. Since the importance of the circumstances in the identification of all the involved citizens, some circumstances are needed in this first level. After this identification the public products and services can be presented per involved citizen so they can be selected. In order to make this selection possible, additional information is needed about the services like regulation, needed formal documents and timing aspects in the delivery of the service. In the third level the circumstances of the citizen are used to determine the needed input per public services which has been selected in the specification level. In the fourth level the service is delivered through the back office. Since all

[2] The names and abbreviations of the following services refer to Dutch legislation.

circumstances are used in levels one, two and three no circumstances are needed in level four. A governing process is needed to make sure that service delivery lead times from identification until actual delivery are according to law. With these two models we were able to model and work with life events.

2.3 Business Rules

One of the basic principles in the business modeling approach of the SVB is that business rules should not be hidden in process models. As a consequence, rules regarding legislation and decision making are separated from rules governing the flow of processes (See also Paragraph 2.4) (e.g. [8,4]). It is most often the 'know' (the rules regarding legislation and decision making) that must be easily adaptable, whereas the process (the 'flow') can be kept more stable and generic. Therefore it is said that only organisations that manage to quickly adapt and (re)deploy their rules with respect to the 'know' can be truly agile.

One of the conventions in the SVB business modeling approach is to define business rules in natural language with RuleSpeak [8]. Natural language and readibility for business users and legal experts was a prerequisite; any format that would look like programming code or use variables and functions were considered unwanted. RuleSpeak was found to be appropriate, being a set of practical guidelines for declarative natural language statements, based on ORM [5] which on its turn was also the basis for SBVR [9]. RuleSpeak combines well-structuredness with readability. Rules in RuleSpeak are structured according to the ORM and SBVR basic concepts of terms and facts. Terms are collected in a business vocabulary and express the business concepts from a business domain. Facts relate these terms in a meaningful way, can be modeled using a fact model and rules can be seen as constraints on such a model. This is illustrated in the following example.

Example 2 (Illustrating terms, facts, and rules):
Suppose the vocabulary has clearly and unambiguously defined the terms `customer`*,* `age`*, and* `insurance`*. Fact types that may be crafted with these terms are:* `Customer has age` *and* `Customer applies for insurance`*. A rule using these facts could be:* `A customer may only apply for an insurance, if he is older than 18`*.* ◇

RuleSpeak gives sentence patterns and rule keywords to express the constraining or deriving rule statements that are built with facts. The rule keywords are *must* and *only*, and every business rule must contain one of these words.

Relevant legislation was thus formalized into sets of rules. We found that legislation can not be translated one-on-one to business rule statements. Instead, we carefully built a domain model and defined the business rules accordingly. Put differently, we have observed that in modeling rules from Dutch law, the structure of the law and the order of the articles is not necessarily mirrored in the (granularity of the) rules. Where the original law text gives criteria for both eligibility and duration in one article, we chose to model the rules on eligibility in a different rule group than the rules on duration. Also, RuleSpeak advises to express a computation in words, not using the symbols for mathematical operations. The SVB recorded its own guidelines allowing formulas in business rules, since Dutch law texts incorporate formulas with letters and mathematical operators.

2.4 Dynamic Process

The traditional process modeling approach results in set of fixed patterns, one for each service or product. This fixed way of modeling entails, among other things, that generic activities such as receiving and digitising messages, or sending notifications to customers are modeled multiple times.

A key driver in the new approach focussed on life events is to optimize and re-use processes whenever possible. This has implications at two levels. This implies that the way of working should be decomposed in reusable "chunks" that can be combined dynamically to handle any given life event. The Lego analogy goes a long way in this example: using the different types of Lego blocks one can create structures of practically any kind. Even more, the Lego blocks do not "know" what type of structure they are / can be part of. Indeed, the "chunks" should be defined in such a way that they are (a) independent of each other, and (b) recognisable as specific tasks to employees of the SVB.

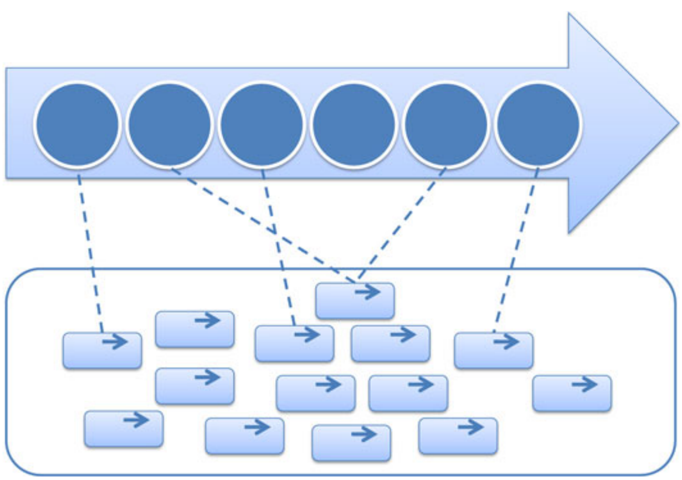

Fig. 3. Handling a life event based on reusable units of work

In the new way of working we have chosen to use a dynamic approach to processes in which generic process steps are combined and reused (more than once, if necessary) to handle the work associated with handling a life event. Figure 3 illustrates this. The figure shows that the process in which a life event is combined by reusing several generic process steps. Even more, one step is used twice.

Our approach to dynamic processes relies heavily on using a combination of process models and business rules. The use of (flow) rules to govern the (order of) execution of processes is explained in Paragraph 2.3 and 3.3.

2.5 Governing Process

Due to the dynamic process modeling approach it is necessary to include process that orchestrates the order in which process steps are executed. We have dubbed this process the "governing process", reflecting the fact that it governs the proper execution of the handling of the life event. The key goal of this process as implemented by the SVB is to ensure that a life event is handled correctly and in a timely manner. The correct handling of a life event implies that assessment of the impact of the life event must be correct in terms of which services are applicable to this particular event. The timely handling of events lies in the fact that legislation determines how long the SVB has to deal with events, taking into account certain exceptions and catering for delays when clients respond slowly/poorly to questions.

This has two consequences (1) as soon as the SVB receives notice of some life event, a governing process for this event must be initiated. This process "lives" as long as it takes to handle the life event. (2) the primary goal of the governing process is to ensure that the right process steps are executed in the right order, while keeping a constant eye on the amount of time that is still available to deal with the event. To keep track of progress, the governing process therefore routinely exchanges status information with the process steps it governs. The complicating factor in designing a governing process that implements these functions lies in the fact that a single life event may have impact on more than one person. Even more, for each person, the life event may have impact on different services (with underlying legislation, encoded in business rules as explained in Paragraph 2.3).

The basic working of the interaction between the operational level and the governing process is illustrated in Figure 4, using the process modeling notation as used in the tool BiZZdesigner [12]. This example shows the different process steps (at the bottom), the governing process (at the top) as well as the interaction points between them. As can be seen, some interaction points are reused

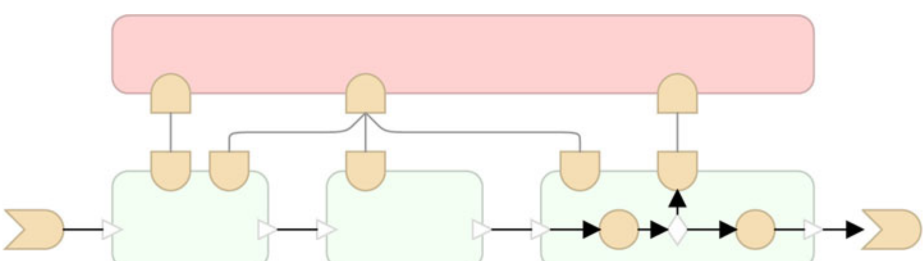

Fig. 4. Governing process

between process steps. This is the case for generic things like exchanging status information. Other interaction points are unique per process step. This is the case for e.g. starting and stopping work, sending information requests to the client et cetera. Note that the interaction mechanism allows for a “plug and play” architecture, where different types of process steps are plugged into the governing mechanism.

The governing process has three levels. All status information from the process steps flows primarily to the service level. This is where we keep track of how long weve been busy with a piece of work; how many days we have remaining etcetera. At the person level, we keep track of different tasks per person. That is, based on information of the service level (note: the service level deals with services with respect to a single person), we attempt to bundle work and interactions with the customer. The person-level reports back to the life event level. This is where we manage the handling of the entire life event for different people. Since most of the work is governed in lower levels. The following example illustrated typical actions in each of the three levels.

level	**responsibility & actions**
Service level	During the execution of a task it becomes clear that we have to send a letter to the customer to ask for more information about his situation. In this case, the timer that keeps track of how much time is left for dealing with this task has to be stopped until the answer is in.
Person level	For two different tasks (two different services) we need information about the current income of the customer. We only want to ask the customer for this information once. Therefore, the person layer keeps track of what weve asked the customer and when.
Life event level	When new information arrives about a current case (e.g., the answer to a written question), this level ensures that the new information is coupled to the ongoing handling proces(ses) as well as triggers lower-level handling processes that timers should be restarted, et cetera.

3 Issues

3.1 Issues with Life Events

One of the main issues that we have identified in dealing with life events lies in the assumption of objective observation. We assumed that it would be possible to objectively identify life events which was not the case, mostly due to the fact that the criteria for distinguishing one life event from another were subjective. Other issues that we have encountered are:

– It is not always clear to whom products and services should be delivered. Due to complex legislation it is possible that a life event affects more than one person, depending on highly specific circumstances. Whether a set of circumstances matches the set of criteria is - in some cases - a matter of opinion.

- The originally identified life event can be wrong after close examination. This issue is best illustrated with an example. A (single) person moving, need not always pertain to the life event “moving within the Netherlands”, even though all the criteria for this event are matched. It may be that the *actual* life event is a divorce.
- The integrated service delivery is not always determinable in advance. Even though service delivery was designed to be integrated as much as possible, it turned out that services may have different lead times. Due to legislation that asserts the maximum time available for handling a life event, it may therefore be the case that outcomes of all services cannot be packaged and communicated in one go.

3.2 Issues with Dynamic Modeling

The core issue that we faced with dynamic modeling follows directly from the lack of a standard notation/ approach to such a way of modeling. More precisely put, current modeling approaches assume that processes are modeled as a consistent whole and *therefore* lack notation to explicitly represent the fact that (the model of) a process can be considered a building block that is used in assembling a larger process similar to the fact that lego’s can be used to build a variety of structures.

Consider the handling of a life event whereby the SVB and its customers “take turns” in doing work. This is particularly the case when the information pertaining to the life event is not all that clear, or when information is missing. An example of was given in Paragraph 3.1 when we discussed that moving within the Netherlands may actually pertain to the life event “divorce”. Asking for more information has the effect that it is “the customers turn” to get some work done which may take a while. As new messages pertaining the same life event reach the SVB, generic process steps may have to be executed more than once in order to succesfully handle a life event.

This brief example shows that, due to the dynamic approach, process models are no longer linear. Since we had slightly amended standard modeling language, and re-interpreted particular concepts in these languages, the models *appeared* (visually) to be linear, especially since they closely resembled the traditional process models that are used in the SVB in terms of notation and presentation. Indeed, we found that stakeholders read/ interpreted these models as being linear. As a concsequence, more time than expected had to be invested in explaining our models to different groups of stakeholders.

3.3 Issues with the Cohesion of Models

A dynamic approach to modeling life events depends on the cohesion between several models (business rules, process model, underlying fact model). Communicating each of these *individual* models with respective groups of stakeholders is challenging, as is a well-known fact in any modeling approach. However, these communication issues are compounded significantly when communicating the

the *combined* set of models cohesively to different stakeholders such as (a) the business, i.e. the people who will actually perform the work in a new way, (b) management, having to decide upon the details, (c) the legal department, checking if the complexities of the law have been implemented properly, (d) the IT services department.

Since the SVB has achieved excellent results with a process atlas in the past, it seemed natural to take process models as a starting point; the assumption being that the type of process models that have been used so far are widely understood, yet formal enough to function as specifications for a software system as well. Even more, from previous research [13] we knew that approaches based on process models are easily combined with business rules. Indeed, business rules have been used extensively in order to govern execution of process models, to codify legislation etcetera. This did leave us with several challenges pertaining to process models, rules, and their combination:

- many languages are available for modeling business rules, varying in the level of formality and readability by non-technical stakeholders. In a previous pilot project, SVB had experimented with operational rules (Event-Condition-Action), which seems to fit best in event-driven situations. The readability of this form for legal experts, however, turned out to be a problem. The paradoxical situation arises that rules should be human-readable for business users, while at the same time be formal and specific in order for a machine to interpret and excute them. It was therefore decided that a two-pronged approach was to be used for the time being: specify the rules in natural language (but as formally / structured as possible) and convert them to an execution platform later. This allowed for easier / better validation by the legal department and allows for a less steep learning curve. At a later point, full-blown SBVR might be introduced to let the two steps converge into one [9].

- In combining process models and rules, we had several challenges to overcome in terms of tooling (we used different software tools for modeling processes and rules) and technique. As explained in Paragraph 2.3, we distinguished between 'know' and 'flow' type of rules, each of which may have to be executed in a single activity (of a process step). A distinction that is made, then, is the logical excecution of rules on the one hand, and the application service that may have to be called to gather the required information to be able to execute them.

  This is illustrated in Figure 5: before we execute an activity, we firstly execute the flow-rules associated with this activity. This determines, among other things, whether the activity should be executed at all. When the activity should indeed be executed, the data that are necessary for executing know-rules are retrieved using application services. The rule set also determines if additional data is missing. If this is the case, a dialogue may be necessary so that the missing data may be added.

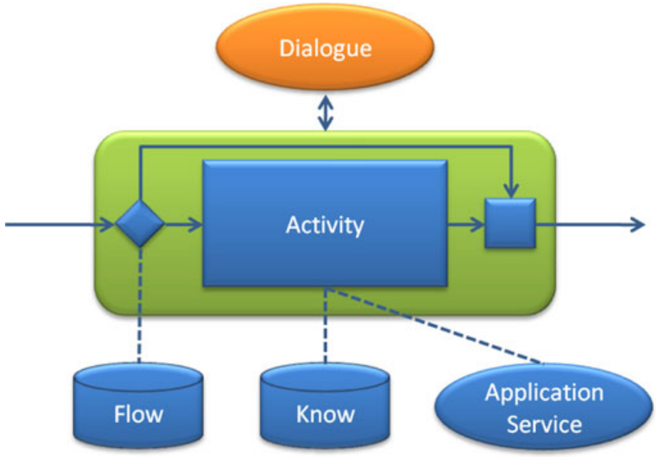

Fig. 5. Combing rules and processes

3.4 Project Organizational Issues

As described above, the total set of business models consisted of process models, business rules and data models. In a project of this size it is a challenge to organize a project team in such a way that all aspects of business modeling are represented, and more importantly that the models are integrated and coordinated.

This issue was addressed by forming a multidisciplinary team consisting of several subteams, including a business rules team, a process modeling team and an object modeling team. The teams were led by two teammanagers who were responsible for planning and the day-to-day management of the subteams. The team managers reported to one project manager who was responsible for the progress of the overall project and who managed all external stakeholders. In addition to the aspect teams, an experienced enterprise architect was involved as project architect. His role was to counsel the teams methodologically, and who to guard compliance to the referential enterprise architecture. The project architect also managed the communication from the teams back to the counsel of enterprise architects.

The project team worked in a strong iterative and incremental fashion. The work was structured into work packages of four to six weeks, depending on the relative size of a topic. This ensured that the different model types were coordinated for a formal delivery at least every four weeks. An major challenge was to manage the dependencies between the teams during these work packages. In order to facilitate close interaction between the teams we held near-daily stand-up meetings. These stand-up meetings, which were held in the morning at the start of a work day, were intended to last fifteen minutes. During these sessions each subteam talked about what tasks they have been working on since the last session, what problems they encountered and whether there were dependencies between teams that need to be addressed. Each work package followed the a Plan-Do-Check-Act quality cycle [1]:

- **Plan:** In a workshop with the different aspect teams, the high level tasks were broken down to a task level. The duration of the tasks were estimated and the tasks were assigned to teammembers.
- **Do:** The actual work during a workpackage was performed as indicated in the above paragraphs.
- **Check:** At the end of a workpackage each aspect team did a review session. During this review session we evaluated the workpackage and discussed what practices to retain and what practices to discard.
- **Act:** the suggested corrections and improvements in the process were implemented into the work process. The new principles and renewed process was documented in a project wiki.

In this way we followed a continuous improvement cycle necessary for a team that was working in an innovative environment. Although the team composition, the iterative working style and the frequent meeting, an important challenge remained. It was very hard to find a good balance between meeting and coordinating with each other, and to actually get any work done. Especially in the first few months of the project, the stand-up sessions took more than the planned fifteen minutes.

4 Conclusion

Business modeling has been a new discipline for SVB. We both adopted several best practices and developed our own methods. For event definition we used some best practices but had to solve ourselves some complicated issues. For business rule definition we used SBVR related RuleSpeak and enriched it with some specific patterns. For object modeling we mixed UML and ERD, and had to invent its relationship with rules, based on ORM. For dynamic process modeling we quickly learned that existing languages like BPMN all assume that processes are modelled from start to end. In order to introduce the dynamic aspect in our process models - i.e., using generic building blocks - we had to adapt existing approaches in terms of approach and notation.

Based on these experiences we can conclude this article with a couple of recommendations, as well as with several subjects that need further research because of the lack of theories and best practices.

4.1 Recommendations

- Use an iterative way of working with a multidisplinary team. Communication was quintessential to our project, with different stakeholders as well as within the modeling team. We found it to be a challenge to organize a project team in such a way that all aspects of business modeling are represented, and more importantly that the models are integrated and coordinated. Only by

working in a strong iterative and incremental fashion did we manage to align the efforts of different modeling-groups, and keep models consistent and focussed.
- The use of RuleSpeak resulted in a very smooth validation process with business users and legal experts, due to the natural language character of RuleSpeak and the lack of technical jargon.
- The use of proven (meta-)ontologies like the one for life events gave us a jump ahead and is a recommended practice.

4.2 Suggestions for Further Research

- There is a need for a new notation method for dynamic process modeling. The main requisite is to be able to present the process models in an understandable way for business users without losing the dynamic character.
- There is a need for a methodology that incoporates the cohesion between rules, processes and objects (or concepts). SBVR and ORM provide a formal relationship between rules and concepts (terms), but a convention for modeling the relationship between rules and process models is lacking.
- Defining life events needs more sophistication in order to enable modeling the detailed cases we were encoutering. The intuitions that were needed to assess the right life event, as mentioned in Paragraph 3.1, can be read as an invitation to analyse these intuitions and to create a life event domain knowledge model.

All in all, we experienced that using a dynamic (process) modeling approach with business rules in order to implement life events can be succesful.

References

1. Dwing, W.E.: Out of the Crisis. MIT Center for Advanced Engineering Study (1989)
2. Richardson, G. L. et al.: A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. In: MIS Quarterly, pp 385–403 (1990)
3. Stader, J., Bridge, S.: Results of the Enterprise Project. In: Proceedings of Expert Systems 1996. In: The 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems, pp. 888–893 (1996)
4. Burlton, R.T.: Business Process Management: Profiting from Process. Sams Publishing, USA (2001)
5. Halpin, T.: Information Modeling and Relational Databases - From conceptual analysis to logical design. Morgan Kaufman, Academic Press (2001)
6. Wimmer, M. and E. Tambouris: Online One-Stop Government - A working framework and requirements. In: Information Systems, pp 117-130 (2002)
7. Business Rules Group: Business Rules Manifesto - the principle of rule independence. Version 2.0 (November 2003)
8. Ross, R.G.: Principles of the Business Rule Approach. Addison-Wesley, Reading (2003)

9. Object Management Group: Semantics of Business Vocabulary and Business Rules (SBVR). Technical specification no. dtc/06-08-05 (2006)
10. Todorovski, L., Leben, A., Kunstelj, M., Cukjati, D., Vintar, M.: Methodology for Building Models of Life Events for Active Portals. In: Wimmer, M.A., et al. (eds.) Proceedings of 5th International Conference on Communication. Springer, Heidelberg (2006)
11. Trochidis, I., Tambouris, E., Tarabanis, K.: An Ontology for Modeling Life-Events. In: IEEE International Conference on Services Computing, pp. 719–720 (2007)
12. BiZZdesign:Handboek Business Process Engineering (in Dutch). Van Haren Publising (2008)
13. Gong,Y., Janssen, M., Overbeek, S.: Zuurmond: Enabling flexible processes by ECA orchestration architecture. In: Proceedings of the 3rd International Conference on Theory and Practice of Electronic Governance, pp. 19-26 (2009)

Composition of Semantic Process Fragments to Domain-Related Process Families

Claudia Reuter

Fraunhofer Institute for Software and Systems Engineering, Emil-Figge-Str. 91, 44227 Dortmund, Germany
claudia.reuter@isst.fraunhofer.de

Abstract. Efficient and effective process management is considered as key success factor for competitiveness of enterprises in an increasingly complex and closely connected environment. Today, there exists a plentitude of IT-tools that support modeling, execution, monitoring, and even flexible change of processes. Though, most process management solutions offer possibilities for reusing workflow components, development of new process models is still a cost and time consuming task. Either common process knowledge is scattered among a growing amount of process models or it is divided into unspecific components, the interrelations of which are difficult to manage. This problem becomes even worse considering potential variations of workflows. In the SPOT project, we adapted the feature modeling approach in order to represent enterprise-specific process knowledge in the form of process families. Process families consist of semantically enriched process fragments and enable the composition of business processes that conform to domain-related rules and regulations.

Keywords: Process management, semantic process fragment, compositional process modeling, feature modeling, process families.

1 Introduction

The introduction and successful deployment of business process management (BPM) and optimization techniques represents a major challenge in an increasing number of domains. E.g., with regard to cost pressure and quality requirements, healthcare providers and logistics experts strive for solutions that enable efficient management of processes. However, especially in highly dynamic markets process standardization approaches quickly reach their limits: In the healthcare domain, medical treatment processes have to focus on patients as individuals with individual needs; logistic experts have to tailor procurement and delivery processes to the specific requirements of their customers in order to stay competitive. Therefore, BPM solutions must provide the means to efficiently customize process models to the individual demands of a single case and to flexibly create new workflows based on existing process knowledge.

Although, most BPM tools already offer possibilities for reusing existing process and application components, they still lack a formal concept to develop and

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 61–75, 2010.

continuously maintain common process knowledge within an institution. Usually, the process knowledge is distributed among a plentitude of processes, sub processes, and application components. However, these fragments only represent the building blocks of process knowledge; equally important are the specification of conditions under which components can be selected and the management of common information about their dependencies and interrelations. In the context of the SPOT project, we analyzed requirements of the domains healthcare and logistics with regard to process management. Facing high flexibility demands, we developed a new approach towards creation and maintenance of domain-related process knowledge in the form of process families. This concept provides the basis for our solution of compositional process modeling, which makes it possible to efficiently derive new business processes according to common process knowledge; our approach not only contributes to the efficiency of process modeling, but ensures that all business processes conform to general rules and regulations of an enterprise.

In this paper, we will introduce Semantic Process Fragments (SPF), which represent basic components for the creation of domain-oriented process knowledge. After a discussion of related work in section 2, we briefly introduce the SPOT project and its novel approach towards compositional process modeling in section 3. Then, we present SPF type graphs as models that enable aggregation and semantic annotation of process fragments to domain-related process families (section 4). Following a presentation of fundamental concepts, we provide a formal semantics for SPF type graphs based on widely-approved feature modeling constructs. This includes a well-specified set of correctness criteria that ensure the soundness of the model. In section 5, we give an outlook on how SPF type graphs facilitate efficient development of standard and case-specific business processes using scenarios from healthcare and logistics domains as example. The paper ends with a short summary of our work (section 6).

2 Related Work

In recent years, the concept of managing potential process variations within the same model became known under the label “process configuration”. In [1], the authors describe a methodology for defining variant-rich processes for the e-business and automotive domain. Thereby, they use variation points in order to adapt a basic process model to the specific requirements of the actual case. This idea was picked up and further developed in context of the Provop approach [2]. Provop even enables the dynamic configuration of processes at runtime based on a well-defined context model. Another approach towards process configuration is proposed by Gottschalk et al., who introduce configurable workflows as common multiple of all process variations combined with a standard configuration [3]. The application of this configuration results in the basic process model representing the standard way of procedure; modifications of the standard configuration lead to variations of the process model. So far, the process modeling languages YAWL [3] and EPC [4] have been extended to support the new concepts. Although, these solutions make it possible to flexibly adapt standard business processes to specific needs, the basic process always remains the core component, from which variations can be derived. Frequently, the basic process corresponds to a standard way of procedure with regard to a goal, as e.g. the treatment of a specific diagnosis. However, related work in the context of process configuration does not address methods towards management of domain-related process knowledge, which makes it possible to flexibly create different types of process models by composing reusable process fragments.

While Provop and other configurable workflow models comply with the procedural process modeling concept, Declare represents a declarative process modeling language [5]. In this way, the order of execution of process components is no longer determined by the control flow but by specifying constraint relations between process activities. The objective of Declare is the prevention of over-specification, which often leads to very complex workflow models that are difficult to maintain. However, the BPM system Declare supports the flexible arrangement of process activities at runtime; whereas the management of domain-related process knowledge as basis for deriving standard and case-specific process models is not addressed. Although, usage of declarative languages contributes to process flexibility at runtime, they make it difficult to specify standard workflows and reduce the predictability of business processes, which is the main argument to establish BPM solutions in the first place. What is still missing is a method for capturing domain-related process knowledge that enables the efficient creation of business processes in compliance with general practices and regulations of an enterprise.

In [6], the authors also identify the need for a model comprising the entire scope of alternative options and interdependencies between business processes. The paper focus on the introduction of generic ERP modeling steps, which are independent of specific modeling languages and aim at developing such a common model. Furthermore, the approach is evaluated by using OPM (Object Process Methodology) [7] as an example. However, the solution does not define correctness criteria that dictate the way in which the overall model has to be created and ensure that only semantically correct processes can be derived from the model.

3 The SPOT Approach

During the SPOT Project (Service-based and Process-oriented Orchestration Technology)[1] we analyzed requirements from the healthcare and logistics domain on BPM solutions. According to our results, both domains are characterized by very dynamic processes that have to be flexibly and efficiently customized to the case-specific demands. However, due to complexity of current BPM solutions that rely heavily on technology, domain specialists like physicians or logistics managers lack the technical skills, which are necessary to efficiently create and change process models. Thus, they mostly depend on IT-specialists, which possess the technical skills for the development of process models, but have no knowledge how the business processes really work in practice. The SPOT approach stems from the conclusion that domain experts will only be able to modify processes if they can create the initial workflow models by themselves. Consequently, SPOT aims at changing the way, in which process models are developed. Figure 1 shows how the SPOT approach contrasts with the traditional approach towards process modeling.

[1] See http://www.spot.fraunhofer.de/ for details on the SPOT project.

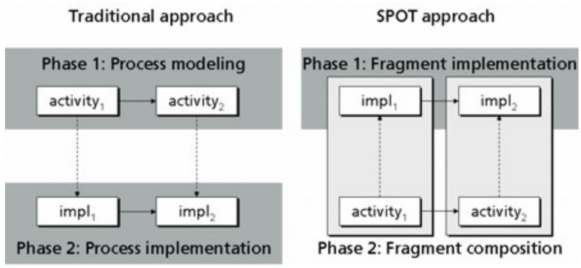

Fig. 1. SPOT approach towards compositional process modeling

Traditionally, the process modeling phase is followed by the implementation phase, which comprises the development of application components and their assignment to process activities. These two procedures must be repeated for every single business process. In SPOT, process development begins with the identification and implementation of process fragments. A process fragment is an executable component that consists of a standardized interface specification and a concrete implementation, as e.g. a web service, a program routine, or a human task interface. This work is usually done by IT-specialists, who can concentrate on their core competencies and do not have to comprehend complex business processes completely. Whereas hospitals and logistic companies often do not have detailed process descriptions at disposal, their service offerings are clearly defined, e.g. in the form of service catalogues, which can be used to identify process fragments. After that, domain experts can overtake the responsibility for the composition of the available process fragments to meaningful business processes. In this way, the two development phases, fragment implementation and fragment composition, better comply with the different competencies of IT-specialists and business professionals. Furthermore, it is possible to create a whole set of workflow models based on the same process fragments.

In order to realize this approach towards compositional process modeling, basically we need a concept that facilitates management of process knowledge in the form of reusable components and determines the way, in which these fragments can be selected and composed into process models.

4 Using Semantic Process Fragments to Build Domain-Related Process Families

In the context of development of software products, the necessity of modeling variant-rich processes is well recognized. A product family represents a set of components which address the demands of a specific market segment, as e.g. the automobile industry. The creation of product families is based on a thorough identification of solution components for a domain by considering possible variations of the resulting product. A widely-approved methodology for the development of product families is the feature modeling approach. This technique was introduced by Kang et al. [8] and since then has been refined in a number of publications [9-15]. In order to facilitate capturing of domain-related process knowledge, we adopted and extended the feature modeling approach. Thereby, each feature represents a concept that is important within the context of at least one business process. With regard to the healthcare domain, a feature can be a radiological examination or a drug therapy. By way of feature modeling, it is possible to abstract from single business processes and to concentrate on work sections and service offerings of enterprises. Then, services that play a role in several processes are classified in a bottom up manner. In contrast to traditional process modeling, dependencies and relations between features are only specified if they adhere to common rules of an institution and thus, comply with practical process knowledge.

Although, feature models are a well-established concept for modeling variability in product lines, most solution approaches still lack formal semantics [16]. E.g., existing approaches [12-15] do not provide criteria that ensure the correct usage of logical operators or the validity of combinations between operators and constraints. Therefore, we chose the feature modeling constructs that are suited for our goal and formally specified SPF type graphs and respective correctness criteria in order to appropriately design process families.

4.1 Specification of Semantic Process Fragments

Process fragments consist of a standardized interface definition and a concrete implementation. They provide the basis for the development of business processes using a process modeling language, as e.g. BPEL [17], BPMN 2.0 [18], or Workflow Nets [19]. The assignment of application components to fragment specifications is the precondition for the composition of executable process models. The interface definition of process fragments, which are distinguished by their unique identifier, comprises a name, a description, a version number, a role of the responsible agent, as well as a reference to an application component and its respective method. Furthermore, it determines which data objects are passed to the application and which objects are returned to the BPM system responsible for the process execution. IT-specialists who assign implementations to interface specifications have to ensure, that the method parameters of the implementation conform to the definitions provided by the interface. After the realization of process fragments, it is possible to set them into relation by creating a process family as *SPF type graph*. SPF type graphs are acyclic, directed graphs. The following figure illustrates their basic structure.

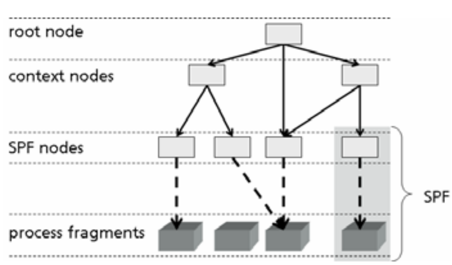

Fig. 2. Basic structure of SPF type graphs

SPF type graphs consist of exactly one root node and an optional set of further nodes. Nodes without child nodes are called *SPF nodes*. Nodes that are neither root nodes nor SPF nodes are called *context nodes*. SPF type graphs assign semantics to process fragments. In this way, they cause the creation of *Semantic Process Fragments (SPF)*, which provide the necessary information about their selection and composition options. As figure 2 shows, an SPF is a process fragment that is assigned to the SPF node of an SPF type graph. In this way, SPF type graphs have to fulfil an important condition that does not hold for feature models: They must provide a function that assigns process fragments to SPF nodes.

4.2 Basic Characteristics of SPF Type Graphs

Principally, SPF type graphs provide the hierarchical and semantic context, which facilitates selection and composition of process fragments to specific process models and ensures that processes conform to the general rules of a domain. In order to illustrate the basic concepts of SPF type graphs, the next figure presents their meta model design as UML class diagram.

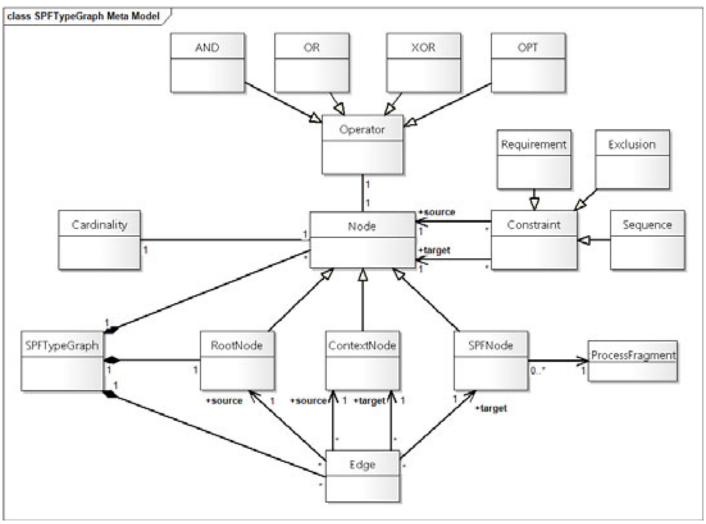

Fig. 3. Meta model of SPF type graphs as UML class diagram

In SPF type graphs, root node, context nodes, and SPF nodes are connected via directed edges. SPF nodes are the only elements that possess a relation to exactly one process fragment; though, the same fragment can be assigned to different SPF nodes. The semantic context of process fragments results from the arrangement of nodes within the graph. As we can see, objects of class type *Node* are related to a logical operator; this operator determines the amount of child nodes that can be selected for a given superior node. We distinguish the following *Operator* class types:

- AND: If a node with AND operator is chosen, all its child nodes have to be selected as well and are therefore mandatory.
- OR: Nodes with OR operator require the selection of at least one successor node.
- XOR: XOR operators only allow for choosing exactly one child node.
- OPT: OPT operators make it possible to select an arbitrary number of successor nodes or none at all. This allows for integrating optional nodes within the graph.

However, with logical operators it is only possible to express relations between successors of the same superior node. Hence, another option to influence the composition of process fragments is given by constraints, which connect two nodes along the horizontal axis. Using constraints, we can specify common rules according to practical process knowledge that should evaluate to true for any business process based on the same SPF type graph. Stemmed from our requirement analysis of healthcare and logistic related processes, we distinguish between the following types of constraints:

- Requirement: The expression "node A requires node B" means that every process model that comprises process fragments associated with node A must at least contain the fragments of B. Furthermore, it must be ensured that the execution of the process fragments of B finishes before the fragments of A are initiated.
- Sequence: The sequence constraint does not impact the selection of process fragments, but it determines their execution order within the process model. The expression "node A precedes node B" means that the process fragments of A must be performed before the fragments of B can be initiated.
- Exclusion: Process fragments associated with nodes that are connected via an exclusion constraint, may not be part of the same process model in the context of the least common sub graph.

The next figure shows how the different types of constraints are visualized in SPF type graphs.

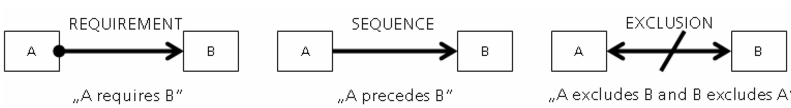

Fig. 4. Representation of possible constraint relations between nodes of SPF type graphs

Finally, it is possible to influence the structure of process models by declaring node cardinalities. In SPF type graphs, the cardinality corresponds to an integer value and indicates the maximum number of times a node can be selected. Multiple instances of nodes are called clone nodes and correspond to loop constructs within traditional process modelling languages.

4.3 Formal Semantics of SPF Type Graphs and Their Correctness Criteria

After an introduction to their basic concepts, we will now give a set-based, formal definition for SPF type graphs and their correctness criteria. An SPF type graph is a tuple STG = (V, E, V_{SPF}, r, C, VO, VC, VP):

- V is a finite nonempty set of unique node labels, including root, SPF nodes, and context nodes.
- $E \subseteq V \times V$ is a finite set of directed edges connecting two nodes with each other. The expression $(v \in V, u \in V) \in E$ indicates the existence of a directed edge leading from v to u. In this respect, we define two abbreviated notations:

$$v \rightarrow u \Leftrightarrow (v, u) \in E \text{ (u is direct successor of v)} .$$

$$v \rightarrow^* u \Leftrightarrow v \rightarrow u \lor \exists z \in V: v \rightarrow z \land z \rightarrow^* u \text{ (path from v to u)} .$$

- $V_{SPF} = \{v \in V \mid \nexists u \in V: v \rightarrow u\}$ is the set of all SPF nodes.
- $r \in V$ is the root node of the SPF type graph.
- $C \subseteq V \times V \times CT$ is a set of constraints that establish constraint relations between nodes, with CT = {REQUIREMENT, SEQUENCE, EXCLUSION} being the set of possible constraint types. E.g. the expression $(v \in V, u \in V, \text{SEQUENCE}) \in C$ means that there is a constraint relation from v to u of type SEQUENCE.
- VO: $V \rightarrow$ {AND, OR, XOR, OPT} is a function that assigns a logical operator to each node. The operator defines the rules under which the child nodes of the current node have to be selected.
- VC: $V \rightarrow \mathbb{N} \setminus 0$ is a function that assigns a cardinality to each node. The cardinality determines the upper limit of the number of clone nodes.
- VP: $V_{SPF} \rightarrow P$ is a function that assigns a process fragment to each SPF node, with P being the set of all available process fragments.

This formal specification states what types of elements may appear in SPF type graphs and in which way they can be interconnected. Based on this formalism, it is possible to define a set of correctness criteria that determine the structure of SPF type graphs. For each SPF type graph STG = (V, E, V_{SPF}, r, C, VO, VC, VP) the following conditions must hold:

Root Conditions. Each SPF type graph must have exactly one root node, which does not possess any superior nodes. If an SPF type graph solely consists of a root node, the root node simultaneously represents an SPF node, which must be related to a process fragment. The cardinality of the root node is always one.

$$r \in V \text{ (existence of the root node)} .$$

$$\nexists v \in V: v \rightarrow r \text{ (no superior node)} .$$

$$VC(r) = 1 \text{ (no clone nodes)} .$$

$$|V| = 1 \Leftrightarrow r \in V_{SPF} \text{ (root as SPF node)} .$$

SPF Node Condition. Besides the fact, that SPF nodes do not have any child nodes, each SPF node must refer to exactly one process fragment within P.

$$\forall v \in V_{SPF}\ \exists!\, p \in P: VP(v) = p .$$

Reachability Condition. There must be a path of directed edges from the root node to any other node within the SPF type graph, whether it corresponds to a context or an SPF node. Though, it is possible that a given node can be reached using different paths. In this way, the semantic context of clone nodes may also differ, depending on its route of selection.

$$\forall v \in V: v \neq r \Rightarrow \exists\ u \in V: u \rightarrow v\ .$$

Acyclic Graph Condition. With regard to their set of hierarchic, directed edges, SPF type graphs have to be acyclic.

$$\forall v, u \in V: v \rightarrow^* u \Rightarrow \neg\,(u \rightarrow^* v)\ .$$

Constraint Conditions. The definition of constraint relations is forbidden for nodes that are connected via hierarchical edges.

$$\forall v, u \in V, ct \in CT: (v, u, ct) \in C \Rightarrow \neg\,(v \rightarrow^* u \vee u \rightarrow^* v)\ .$$

More specifically, two constraints must not refer to the same scope regarding the semantic context of an SPF type graph. The scope always consists of the nodes that are connected via the constraint as well as their associated sub graphs. A constraint relation $(v, u, ct) \in C$ means that it is not possible to define further constraints between the sub graphs of v and u respectively. In order to be able to formally specify this condition, first we have to introduce an auxiliary function minNode: $2^V \rightarrow V$ that serves the purpose of identifying the least common multiple node of a given set of nodes. Let $Z \subseteq V$ be a non-empty set of nodes of an SPF type graph; then minNode can be defined in the following way:

$$\mathrm{minNode}(Z) = \{v \in V \mid v \rightarrow^* Z \setminus v \wedge \neg\,(\exists\ u \in V: u \rightarrow^* Z \wedge v \rightarrow^* u)\}\ .$$

According to this definition, it is possible that the least common superior node is part of Z itself. Now, we can define the correctness criterion that ensures that the scope of a constraint relation does not overlap with the scope of another constraint:

$$\forall v, u \in V, ct_1 \in CT: (v, u, ct_1) \in C \Rightarrow \nexists x, y \in V, ct_2 \in CT: (x, y, ct_2) \in C \setminus (v, u, ct_1) \wedge ((v \rightarrow^* x \vee u \rightarrow^* x \vee x = v \vee x = u) \wedge ((v \rightarrow^* y \vee u \rightarrow^* y \vee y = v \vee y = u) \wedge (\mathrm{minNode}(x, y) \rightarrow^* v \vee \mathrm{minNode}(x, y) \rightarrow^* u)\ .$$

The following figure presents an example that shows the importance of this condition.

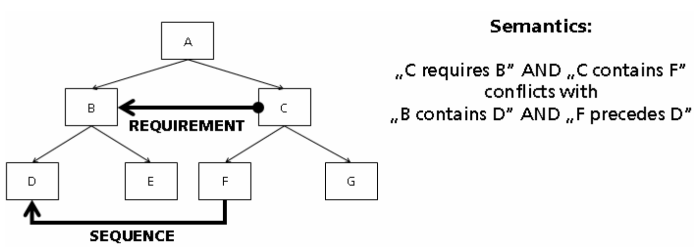

Fig. 5. Example of a conflict between constraints within the same scope

In the example above, the sequence constraint between the nodes D and F violates the requirement condition between C and B, because it is part of the same scope. This criterion also prohibits the specification of constraints, where source and target are identical as well as loop constraints.

The next condition refers to the transitivity of constraint relations. If node v requires node u and u requires z, v requires z as well; please note, that the same is true for sequence constraints.

$$\exists\, v, u, z \in V: (v, u, \text{REQUIREMENT}), (u, z, \text{REQUIREMENT}) \in C \Rightarrow \exists\ (v, z, \text{REQUIREMENT}) \in C\,.$$

Finally, it is not allowed to define requirement and exclusion constraints in such a way, that it is not possible to fulfill the conditions:

$$\exists\, v, u, z \in V: (v, u, \text{REQUIREMENT}), (v, z, \text{REQUIREMENT}) \in C \Rightarrow \nexists\, (u, z, \text{EXCLUSION}) \in C \wedge \nexists\, (z, u, \text{EXCLUSION}) \in C\,.$$

Conditions with Respect to Combinations of Logical Operators and Constraints. Besides correctness criteria that only relate to the specification of constraints within SPF type graphs, we also have to deal with problems arising from invalid combinations of logical operators and constraints. Regarding child nodes of a parent with XOR operator, it is not allowed to specify constraints at all.

$$\forall v, u \in V\ \exists z \in V: z \rightarrow^* v \wedge z \rightarrow^* u \wedge VO(z) = XOR \Rightarrow \neg(v, u, ct \in CT) \in C\,.$$

Furthermore, if two nodes are connected via exclusion constraint, it must be ensured that there is at least one option that avoids the selection of both nodes**.**

$$\forall v, u \in V: (v, u, \text{EXCLUSION}) \in C \Rightarrow \exists z \in Z: VO(z) = OPT \vee (VO(z) \neq AND \wedge (\exists\ x \in V \setminus Z: z \rightarrow x)) \text{ with } Z = \{z \in V \mid z = minNode(v, u) \vee ((z \rightarrow^* v \vee z \rightarrow^* u) \wedge minNode(v, u) \rightarrow^* z)\}\,.$$

According to this formula, Z comprises all nodes that correspond to any node that is located between v or u and their least common superior node or represents the least common superior node itself. Either one of these nodes is associated with the logical operator OPT or it must contain a child node as alternative option to v or u.

5 Compositional Process Modeling in Healthcare and Logistics

In this section, we would like to clarify the contribution of SPF type graphs to compositional process modeling by way of examples from the healthcare and logistics domain.

5.1 Treatment Processes for Diagnostics of Spinal Diseases

About three quarters of the adult population will experience back pain during their lifetime. The medical treatment can be a variable process depending on the severity level of a spinal disease and the individual situation of a patient. However, process management solutions that facilitate the coordination of the activities of the participating healthcare providers can considerably contribute to the success of the treatment. Assuming that a hospital has already begun to maintain process knowledge in the form of SPF type graphs. Given the process family for diagnostic procedures, as illustrated in figure 6, it is possible to compose new process models for the diagnostics of spinal diseases based on semantic process fragments.

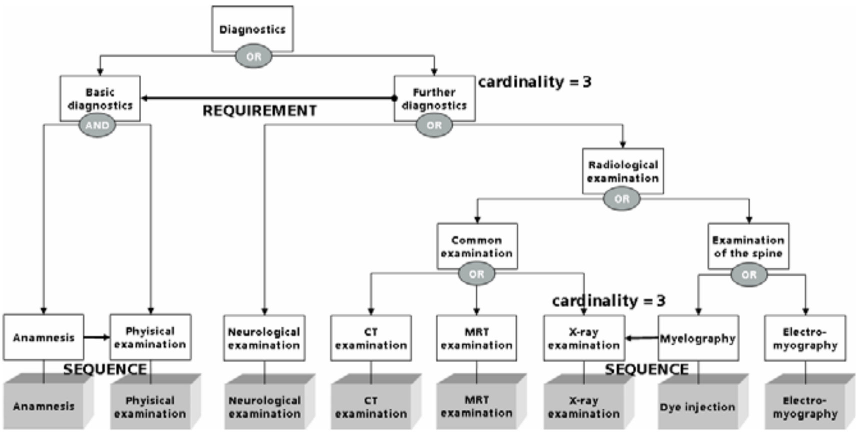

Fig. 6. Example of an SPF Type graph for diagnostic procedures

According to this graph, diagnostics can be distinguished in basic and further diagnostics. While basic diagnostics schedules the anamnesis and physical examination, further diagnostics is related to neurological and radiological examinations. Though, considering the requirement constraint, further examinations must not take place as long as the basic diagnostics is not completed. With respect to radiological examinations, we divide common examinations from examinations of the spine. Common examinations, such as X-ray, MRT, and CT, can be performed in the context of many diagnoses; whereby myelography and electromyography are used to diagnose the source of back pain. The myelography involves the injection of contrast medium into the spine, followed by X-ray examinations. As the node cardinalities show, it is possible to repeat X-ray examinations or even the whole complex of further diagnostics up to three times.

Now, new processes derived from the SPF type graph can either outline the standard way of procedure for patients that suffer from back pain or they can deal with individual treatment cases. The next figure visualizes the standard process for diagnostics of spinal diseases according to the SPF type graph.

The order of execution of the process activities is determined by the constraint relations. Due to a sequence constraint, the physical examination always follows the anamnesis. As there is no constraint between MRT and X-ray, the examinations can be performed in arbitrary order. Note that a constraint between two context nodes also adheres to the nodes of their associated sub graphs.

In standard treatment cases, physicians perform an X-ray and an MRT after completion of anamnesis and physical examination. Though, assume that we have to deal with a patient that is suspected to suffer from a recurrent lumbar disc hernia;

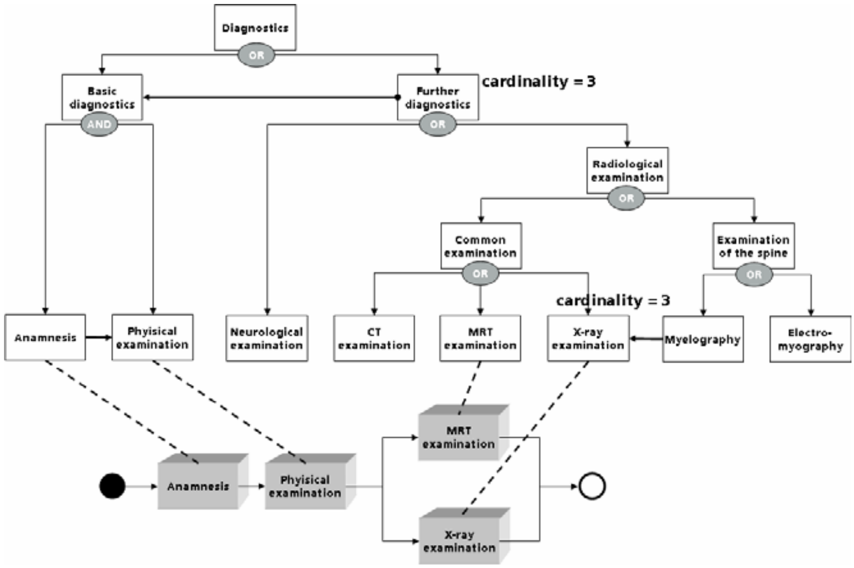

Fig. 7. Basic process for the diagnostics of spinal diseases

therefore, a myelography is indicated to distinguish a relapse from a local post-operative scar formation. A myelography is an x-ray scan that is performed after dye has been injected into the spinal fluid. Consequently, the diagnostic procedures for this patient must differ from the standard process; figure 8 illustrates the individual workflow model.

Fig. 8. Case-specific process for the diagnostic of spinal diseases

In this way, SPF type graphs provide the basis for deriving standard processes, which are executed for the majority of patients, as well as individual processes depending on the requirements of specific treatment cases.

5.2 Logistic Process for the Transportation of Goods

Goods have to be transported from their place of production to the various locations of the customers. Generally, logistic processes can vary according to the type of transfer and the transportation mode as illustrated in the following SPF type graph.

As the figure shows, there are different modes of transportation, which primarily depend on source, destination, costs, and safety. Potential carriers are air-freight, railroad, waterway, and motor-freight carriers. In some cases it is possible to directly transfer the goods from origin to destination; thereby, the process only comprises

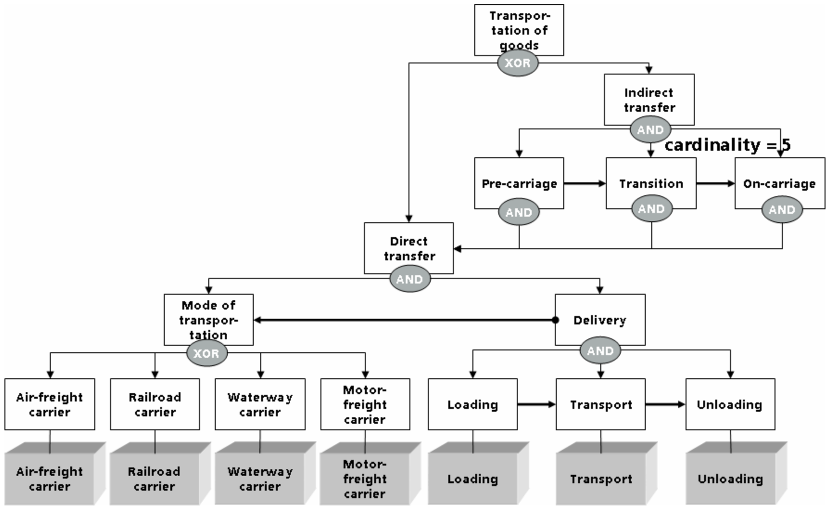

Fig. 9. Example of an SPF type graph for the transportation of goods

selection and arrangement of the transportation mode as well as the loading, transport, and unloading of the goods. Though, in case of air-freight, railroad, and waterway carrier, mostly it is also necessary to organize the pre- and on-carriage. The drive of the lorry from the loader to the terminal of departure is considered as pre-carriage; on-carriage covers the delivery of the freight from the terminal of destination to the receiving costumer. Therefore, one or more transitions occur between pre- and on-carriage. The SPF type graph defines a maximum number of five transitions.

Now, a logistics manager composes for her new customer a transport process; as mode of transportation she is going to choose air-freight carrier. In order to organize the transport to and from the airport terminals, she decides on pre- and on-carriage via motor-freight carrier. The next figure shows the resulting process model.

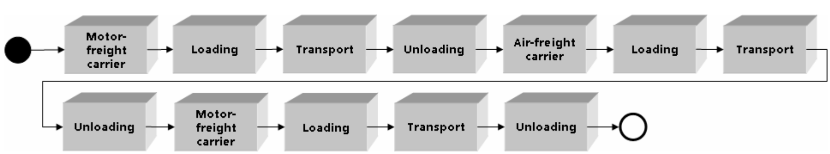

Fig. 10. Case-specific process for the indirect transfer of goods

With our approach of compositional process modeling, the SPF type graph makes it possible to efficiently tailor logistics processes to the specific needs of a customer.

6 Conclusion

The necessity to specify detailed business processes on a technical level represents a major obstacle for using BPM systems. BPM repositories for management of sub workflows and application components still lack possibilities to formally define the context of their usage and their interrelations with each other. In order to enable our approach towards compositional process modeling, we formally defined SPF type graphs as models to represent domain-oriented process families. SPF type graphs are a specific kind of feature model that allow the management of the common process knowledge within an enterprise. In contrast to other solutions in the area of process configuration, SPF type graphs do not correspond to basic processes, which contain variable elements that can be customized to the actual needs of a case. As we have shown by example of processes from the healthcare and logistic domains, it is rather possible to derive standard as well as case-specific workflows based on the same SPF type graph. Furthermore, our methodology makes process modeling easier, because IT-specialists can concentrate on the task of identifying and developing process fragments according to the service offerings of an enterprise; then, the composition of fragments to business processes can be done by domain experts. In our future work, we will formally specify criteria that ensure that fragment compositions conform to the definitions of an SPF type graph. Moreover, we will determine the rules that control the transformation of declarative process knowledge into specific process models using languages that follow the procedural modeling paradigm.

References

1. Bayer, J., Buhl, W., Giese, C., Lehner, T., Ocampo, A., Puhlmann, F., Richter, E., Schnieders, A., Weiland, J., Weske, M.: Process Family Engineering, Modeling variant-rich processes. Fraunhofer IESE Report No. 126.06/E, Version 1.0 (2005)
2. Hallerbach, A., Bauer, T., Reichert, M.: Issues in Modeling Process Variants with Provop. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 17, pp. 56–67. Springer, Heidelberg (2009)
3. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H., La Rosa, M.: Configurable Workflow Models. International Journal of Cooperative Information Systems 17(2), 223–225 (2008)
4. Rosemann, M., van der Aalst, W.M.P.: A Configurable Reference Modeling Language. Information Systems 32, 1–12 (2007)
5. Aalst, W.M.P.: Constraint-Based Workflow Models: Change Made Easy. In: Curbera, F., Leymann, F., Weske, M. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 77–94. Springer, Heidelberg (2007)
6. Soffer, P., Golany, B., Dori, D.: ERP modeling: a comprehensive approach. Information Systems 28, 673–690 (2003)
7. Dori, D.: Object Process Methodology – a Holistic Systems Paradigm. Springer, Heidelberg (2002)
8. Kang, K., Cohen, S., Hess, J., Nowak, W., Peterson, S.: Feature-oriented domain analysis (FODA) feasibility study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA (1990)
9. Kang, K., Kim, S., Lee, J., Kim, K.: FORM: A Feature-Oriented Reuse Method. Annals of Software Engineering 5, 143–168 (1998)
10. van Deursen, A., Klint, P.: Domain-Specific Language Design Requires Feature Descriptions. Journal of Computing and Information Technology 10(1), 1–17 (2002)

11. van Gurp, J., Bosch, J., Svahnberg, M.: On the Notion of Variability in Software Product Lines. In: Proceeding of the Working IEEE/IFIP Conference on Software Architecture (WICSA), pp. 45–55. IEEE Computer Society Press, Washington (2001)
12. Czarnecki, K., Helsen, S., Eisenecker, U.W.: Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice 10(1), 7–29 (2005)
13. Mannion, M.: Using First-Order Logic for Product Line Model Validation. In: Chastek, G.J. (ed.) SPLC 2002. LNCS, vol. 2379, pp. 176–187. Springer, Heidelberg (2002)
14. Schobbens, P.-Y., Heymans, P., Trigaux, J.-C.: Feature Diagrams: A Survey and a Formal Semantics. In: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 136–145 (2006)
15. Sun, J., Zhang, H., Li, Y.F., Wang, H.: Formal Semantics and Verification of Feature Modeling. In: Proceedings of the 10th IEEE International Conference on Engineering of Complex Systems (ICECCS 2005), pp. 303–312 (2002)
16. Riebisch, M., Streitferdt, D., Pashov, I.: Modeling variability for object-oriented product lines – workshop report. In: Buschmann, F., Buchmann, A., Cilia, M.A. (eds.) ECCV-WS 2003. LNCS, vol. 3013, pp. 165–178. Springer, Heidelberg (2004)
17. OASIS: Web Services Business Process Execution Language Version 2.0. OASIS Standard (April 2007), `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html` (last access of May 12, 2010)
18. OMG: Business Process Model and Notation (BPMN) FTF Beta 1 Version 2.0 (August 2009), `http://www.omg.org/spec/BPMN/2.0/Beta1/PD` (last access of May 12, 2010)
19. van der Aalst, W.M.P.: The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers 8(1), 21–66 (1998)

Assessing Collaborative Modeling Quality Based on Modeling Artifacts

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. Collaborative modeling uses and produces modeling artifacts whose quality can help us gauge the effectiveness and efficiency of the modeling process. Such artifacts include the modeling language, the modeling procedure, the products and the support tool or medium. To effectively assess the quality of any collaborative modeling process, the (inter-) dependencies of these artifacts and their effect on modeling process quality need to be analyzed. Although a number of research studies have assessed and measured the quality of collaborative processes, no formal (causal) model has been developed to assess the quality of the collaborative modeling process through a combination of modeling artifacts. This paper develops a Collaborative Modeling Process Quality (CMPQ) construct for assessing the quality of collaborative modeling. A modeling session involving 107 students was used to validate and measure the quality constructs in the model.

Keywords: Collaborative Modeling, Modeling Process Quality, Modeling Artifacts, Instrument Validation, Structural Equation Modeling.

1 Introduction

Collaborative modeling, including the modeling of enterprises [1,2,3] and/or of associated business processes [4], brings together stakeholders with varying degrees of knowledge, expertise, skills and competencies. Such collaborative modeling, which is conceptually similar to group model building [5], brings with it a number of benefits and advantages. Although such benefits and advantages have been recognized in the literature, e.g. considerable productivity and improved results [6], substantiating the success of these collaborative efforts is far from trivial.

A number of factors come into play and need to be analyzed if we are to effectively and efficiently measure and evaluate modeling process quality and determine the success of a collaborative effort [7]. First, the different stakeholders have different priorities and preferences which need to be reconciled in a group

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 76–90, 2010.

problem-solving activity, especially during evaluation of the modeling process. Second, a number of modeling artifacts are used in, and produced or amended during, the modeling process. These include the modeling language, the methods or approaches used to solve the problem, the intermediary and end-products produced and the medium or support tool that may be used to aid the collaboration. All these do impact on the success of the collaborative modeling effort and on the quality of the modeling process (especially its effectiveness and efficiency). Although the quality of each of these may be established separately, the quality of the entire modeling process is an aggregation of the quality of all these modeling artifacts.

While a number of approaches have been developed to measure and evaluate the quality of a collaborative modeling process, e.g. its successfulness [8] and users' satisfaction [9], there has not been any study that integrates the assessment of various modeling artifacts to determine the quality of a collaborative modeling process. Driven by the need to determine the efficiency and effectiveness of the modeling process, we propose an evaluation method that indeed integrates the assessments as an alternative method for determining the quality and successfulness of, and users' satisfaction with, a modeling process. To this effect, we develop and test a Collaborative Modeling Process Quality Assessment (CMPQ) construct - a causal model for assessing the quality of the modeling artifacts. We also present a validated instrument that can be used to measure the constructs in the model.

2 Collaborative Modeling Artifacts and Their Quality Constructs

Our approach is anchored on the Technology Acceptance Model (TAM) of Davis [10] and tries to assess the modelers' affective attitudes and perceptions of the quality of the modeling language, the end-products (models), the ease-of-use of a support tool and the usefulness of the modeling procedure. We therefore, identify and define the quality constructs selected to measure these perceptions for each of the modeling artifacts. It is not possible to include all the quality dimensions available in the literature for the modeling artifacts in our discussion. We, however, believe that those selected are a good representation of the modeling artifact quality. Many of the quality dimensions, of these constructs are discussed in [11]. The operationalization of the quality dimensions identified in this section will be provided in Sect. 4 and the evaluation approach will be along the lines of the Method Evaluation Model described by Moody et al. [12].

2.1 Modeling Language

Many conceptual models, which are abstract representation of real world domains, are a collection of linked graphic symbols of an underlying modeling language. To Evaluate the model for quality, one needs to first look at the adequacy (expressive power, completeness, correctness, etc.) of the modeling language. Additionally, the evaluation of the adequacy of the modeling language should also take the perspective of the participant. She should be able to understand the concepts in the language, the concepts should be easy to learn and remember, the language should have a set of signs and symbols for producing the model and it should have well-defined rules for combining signs and symbols. We follow the generic quality framework in [13] to assess the quality of the modeling language. In view of this, we define the modeling language quality construct as follows.

The Perceived Quality of the Modeling Language (PQML) is the user's affective attitude towards a modeling language in providing a syntactic or domain meta language that provides concepts in which modelers define the problem, express and communicate the solution.

Quality dimensions for the PQML construct are defined in Table 1.

Table 1. PQML Construct Quality Dimensions

Quality Dimension	Definition
Understandability	Understandability refers to how adequate the model represents concepts you recognize in view of your or someone else's domain knowledge.
Clarity	Clarity of the modeling language refers to how easily you learn and remember the concepts and notations of the modeling language through the signs, symbols, textual expressions of the modeling language.
Syntax correctness	The syntax is the common agreed communication language for agents in a collaborative modeling process and establishes a set of signs which can be exchanged and rules (syntactical rules) governing how the signs can be combined. The syntax is related to the formal relations of signs to one another.
Conceptual minimalism	Conceptual minimalism refers to the existence of primitive (basic) signs and symbols for representing data concepts of the domain as separate objects and assembling the objects to form composite abstractions. Conceptual minimalism relates to the simplicity of the modeling language.

2.2 Modeling Procedure

Any performed task is driven by set goals [14]. A goal is a result that a stakeholder strives to achieve and its awareness is accompanied by a set of perceived goal attainments [15]. This means that Stakeholders in any collaborative modeling effort strive to achieve set goals. To achieve these goals, there must be a well-defined procedure in which they formulate and define the problem and agree on how the solution will be reached. To evaluate and measure the quality of the modeling procedure, one needs to assess whether the group goal is achieved. The most prominent measure for this is effectiveness. In [16], this is viewed as "...the extent to which a result contributes to the establishment of a goal set for the collaboration process" (p.3).

Other quality constructs include e.g. the amount of time to reach the solution and to attain the goals and objectives, time to negotiate, etc. Stakeholders should also be satisfied with the negotiation, the decision and decision making process [17], the communication process and the goals and objectives set and how they are achieved through the modeling procedure. Stakeholders' commitment to supporting the goals and objectives, the collective decisions and their contribution to shared understanding is another measure of the success of a collaborative effort. In view of this discussion, we define the modeling procedure quality construct and its quality dimensions as follows.

Perceived Usefulness of the Modeling Procedure (PUMP) is the user's affective attitude towards the usefulness of the procedure used to detail the processes of how the problem is defined and how the solution is reached.

Quality dimensions for the PQML construct are defined in Table 2.

Table 2. PUMP Construct Quality Dimensions

Quality Dimension	Definition
Efficiency	Efficiency of the modeling procedure refers to the resources, e.g., time, required for reaching the solution and attaining the modeling goals and objectives; the time needed to negotiate, reach agreement and consensus.
Effectiveness	Modeling procedure effectiveness refers to how the modeling procedure enables the modelers in using communication and negotiation to get the expected outcome and thus attain their set goals. It also includes the facilitation and the way the modeling process is carried out and/or conducted, and the decision-making process.
Satisfaction	Satisfaction of the modeling procedure refers to the modelers' positive feeling about the achievement of the intended result using the modeling procedure. Intended results may include intermediary or end-results. Satisfaction can concern the way modelers communicated, negotiated, reached agreement and how they made modeling decisions.
Commitment & Shared Understanding	Commitment and shared understanding refer to the modeler's stake and promise to support the goals and objectives of the modeling process, the responsibility to abide by the modeling rules and group decisions and his/her readiness to contribute to the group's shared understanding.

2.3 End Products

The end products are the results or outcomes of a collaborative modeling process. Where the modeling language is used to generate the products, these products are the models formed. Quality constructs for measuring and assessing the quality of the modeling process outcome include product quality which may include the complexity, abstractness, clarity, correctness, completeness, consistency and understandability of the products, see, for example, [18]. In addition to product quality, modelers should be satisfied with both the process that generates the products and the outcome [19], they should be satisfied with the syntactic, semantic, pragmatic, empirical and physical quality of both the modeling language and the model. In case of models, they should be modifiable and maintainable, i.e. they should be easily changed and re-used. We use the general framework defined in [20] for the assessment of quality of models. The definition of the end-products quality construct in the context of this paper is given next.

The Perceived Quality of the End Products (PQEP) (models) is the user's affective attitude towards the outcome (including intermediary and final models) of a modeling process.

Quality dimensions for the PQML construct are defined in Table 3.

Table 3. PQEP Construct Quality Dimensions

Quality Dimension	Definition
Product Quality	Product quality refers to the accuracy of the model in depicting all the identified aspects, adequate representation of the domain concepts in the products, abstractedness, clarity and correctness.
Understandability	Understandability of the products refers to the degree to which the modelers comprehend the language concepts represented in the products, e.g., its syntax, semantics, etc., the relationship between the different concepts which are depicted by the products, and the ease with which the modelers can explain the concepts in the products even to those who never participated in the modeling process.
Modifiability and Maintainability	Modifiability and maintainability of the products refer to ease of changing the products to accommodate new changes and the degree to which the products can be kept up-to-date, and how easily they can be re-used in the re-engineering and re-structuring of the enterprise processes.
Satisfaction	Product satisfaction of the modelers refers to a positive feeling about the product's quality. This could include satisfaction with respect to the product's correctness, completeness, accuracy, consistency, clarity, understandability and/or its complexity.

2.4 Support-Tool: The Medium

The support tool or the medium is the means that supports and facilitates the collaborative modeling process. This can range from a simple white-board to a group support system (GSS) [21]. To evaluate such a support tool, a number of quality constructs are used. These include the enjoyment and/or fun derived from the use of the support tool [22]; functionality of the tool and its usability [23]. An excellent survey concerning the use of technology and its impact on the performance of groups is given in [24]. The support tool is also required to facilitate the collaboration and communication process, e.g. the negotiation process and decision making process. We define the support-tool or medium construct for assessing a collaborative modeling process as follows.

The ease of use of the medium (EOUM) (or the support tool (EOUST)) is the user's affective attitude towards a technology-based group support system (GSS) that supports the collaborative modeling process.

Quality dimensions for the PQML construct are defined in Table 4.

Table 4. EOUM Construct Quality Dimensions

Quality Dimension	Definition
Functionality	Tool functionality refers to the different functions that a tool has which support activities of the modeling process. It also refers to how the support tool executes the modeling activities and how reliable it is in executing those activities.
Usability	Usability of a tool support refers to its effectiveness and efficiency to achieve specified goals in particular environments. It is a set of attributes which bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users. Where efficiency relates to the level of effectiveness achieved to the expenditure of resources whereas effectiveness refers to the goals or sub-goals of using the support tool to the accuracy and completeness with which these goals can be achieved.
Satisfaction & Enjoyment	Satisfaction refers to perceived usability of the support tool by its users and the acceptability of the support tool to the people who use it and to other people affected by its use. It also refers to the degree of fun and enjoyment by the modelers in using the tool. Measures of satisfaction may relate to specific aspects of the system or may be measures of satisfaction with the overall support system.
Collaboration & Communication Facilitation	Collaboration and communication facilitation refers to the degree to which the support system helps modelers to collaboratively achieve the set goals and objectives. It also refers to the ability of the support system to aid the communication process and decision making process to reach agreement and consensus.

3 Modeling Experiment Set-Up

This part of the paper discusses the following: description of the modeling experiment and subjects and the modeling task.

• **Modeling Experiment and Subjects.** To assess the quality of the CMPQ construct, we conducted a modeling session. The subjects were third-year undergraduate (day and evening) students offering a Bachelor of Information Technology and Computing (BITC) degree course. The modeling experiment was conducted after an introductory course in information and system modeling using UML. A simple UML editor, embedded within the Collaborative Modeling Architecture (COMA) tool [25], was used. A total of 107 students participated in the modeling experiment. They were divided into 6 groups with an average of 17 participants and 3 or 4 participants per computer terminal. The modeling experiments were conducted on two days, each day having three groups. Each experiment lasted for not more than 70 minutes.

• **Task Description.** The modeling case that was given to the students concerned procurement of medical drugs and equipment by the Pharmacy and Medical Equipment Department of a University Teaching Hospital and distributing these to the different wards and departments of the University Teaching Hospital. The students' task was to: 1) identify the different processes, associated activities and objects; 2) develop the conceptual model using COMA's UML editor; and 3) assess the quality of the whole collaborative modeling process by filling-out the given questionnaire immediately after the modeling session. Task 1 is associated with the modeling procedure whose quality is assessed via the PUMP construct. Task 2 is concerned with the end-products (conceptual model), the modeling language (UML) and the medium or support tool (COMA) whose quality is measured, respectively, via the PQEP, PQML and EOUM constructs. Task 3 is concerned with assessing the quality of the whole collaborative modeling process via the CMPQ construct by filling-out the given questionnaire. The development, reliability and validity testing of this CMPQ Construct Measurement Instrument is described next.

4 CMPQ Construct Measurement Instrument

Evaluating and measuring the quality of the collaborative modeling process through the modeling artifacts presented in Sec. 2 requires a validated instrument. We did a literature survey for previously validated instruments narrowing the scope to only those assessing: *1) collaborative modeling quality*, *2) users' satisfaction* and *3) success of a collaborative effort* within collaborative modeling and/or group model building. This survey revealed that validated instruments presented in [15,16,17,26,27] satisfy at least one of the selection criteria given above and the first is the closest to this study. Unfortunately, being from the area of Collaboration Engineering, it looks at collaboration patterns. Many of the quality constructs presented in all these studies provide a background to the quality dimensions discussed in this paper. Although the assessment done therein, tries to get users' perception of quality, successfulness, etc, and satisfaction with product, process, system, decision and the decision making process, we propose an alternative way of performing the assessment.

Our assessment approach evaluates quality of the collaborative effort through the quality of the modeling artifacts: the modeling language, the modeling procedure, the end-products and the support tool or medium. Specifically, we look at the PQML, PUMP, PQEP and EOUM constructs to develop an integrated approach and a construct for assessing the quality of the collaborative modeling process. Although Straub [28] advises to use previously validated research instruments wherever possible, he warns that a major modification to a research instrument will negatively affect its validity and reliability. None of the validated instruments above could satisfy our needs and alteration was not possible since many of the questions on the instruments for similar constructs needed rephrasing to suit our needs. In light of the above observations, it was not possible to adopt or alter any of the existing research instruments. It is for this reason that we developed a new instrument.

4.1 Content Validation of the Instrument

We developed a new instrument to assess and measure Collaborative Modeling Process Quality(CMPQ) through four constructs: PQML, PUMP, PQEP and EOUM. The initial measurement instrument had a total of 44 quality dimensions synthesized from the literature: 10 for PQML, 10 for PUMP, 15 for PQEP and 9 for EOUM. These are shown in the second column of the table in Appendix A.

Content Validation. Content validity, which is established through literature reviews and/or expert panels or judges, measures the degree to which the selected items in the research instrument represent the content pool to which the research instrument will be generalized [29]. A panel of three content experts is considered adequate for content validation [30]. In light of this recommendation, three experts were asked for their judgement about the adequacy and representation of the constructs and their quality dimensions for the CMPQ construct. A 5-point rating scale (with 1 = Highly Appropriate and 5 = Highly Inappropriate) was used to rate the appropriateness of the quality dimensions. The mean value of each of the dimensions ranged between 1.10 and 4.33. This means that some of the quality dimensions were inappropriate.

The qualitative judgement of the experts indicated that the numbers above were too many for any construct and many were found to overlap. It was recommended to refine, merge and group many of these quality constructs. The original dimensions and their refined and merged groupings are shown in Appendix A. The groupings form a set of sixteen quality dimensions for each of the four modeling constructs defined in Sec. 2. The instrument with sixteen questions measuring the quality of the constructs, using a 7-point scale (with 1 = strongly agree and 7 = strongly disagree), is given in Appendix B.

4.2 Exploratory Factor Analysis

We carried out an Exploratory Factor Analysis (EFA) subjecting the 107 case in the data set to Principal Component Analysis (PCA) [31]. The Promax rotation method was used since the data exhibited strong correlations among the extracted factors. To identify the suitable number of factors underlying the CMPQ construct we used the three recommended steps in [32]. We dropped all factors with at most 0.4 values. This condition prevented cross-loading on more than one factor at 0.4 and above. We also applied and repeated factor analysis using 3, 4, 5 and 6 factor loadings. All factors were extracted at eigenvalue of 1. The 4-factor loadings was found to be the most suitable for the CMPQ construct and explains 70.3% of the variance. The factor loadings of the 16 quality dimensions of the four quality constructs measured through the research instrument are given in Table 5.

Table 5. Factor Analysis and Reliability Results

			Factor				Cronbach's α
Construct	**Code**	**Quality Dimension**	1	2	3	4	
PQML	ML1	Understandability	.895				.866
	ML2	Clarity	.798				
	ML3	Syntax Correctness	.886				
	ML4	Conceptual Minimalism	.787				
PUMP	MP5	Efficiency		.718			.850
	MP6	Effectiveness		.883			
	MP7	Satisfaction		.842			
	MP8	Commitment & Shared Understanding		.882			
PQEP	EP9	Product Quality			.833		.834
	EP10	Understandability			.840		
	EP11	Modifiability & Maintainability			.795		
	EP12	Satisfaction			.795		
EOUM	ST13	Functionality				.817	.833
	ST14	Usability				.944	
	ST15	Satisfaction & Enjoyment				.702	
	ST16	Collaboration Communication & Facilitation				.661	
		Eigenvalue	4.57	2.68	2.35	1.65	
		Cumulative Variance Explained (%)	28.6	45.3	60.0	70.3	

Extraction Method: Principal Component Analysis. Rotation Method: Promax with Kaiser Normalization. Rotation converged in 5 iterations.

Reliability Tests and Construct Validity. We note from the results presented that all factor loadings of the 16 items load on a single factor for each of the PQML, PUMP, PQEP and EOUM constructs. This is preliminary evidence of *uni-dimensional reliability* of the research instrument, in the Exploratory Factor Analysis (EFA) method using the Principal Component (PCA) technique, where item measures reflect only one underlying trait or concept [33]. To further check the scientific rigor of the research instrument, and to confirm our a-priori assumptions about the *reliability (internal consistency)* of the research instrument, we computed the Cronbach's alpha values [34]. The computed values, as is evident in Table 5, were all above the threshold value of 0.60 or 0.70 recommended for the EFA method using the the PCA technique [35]. This is proof that the quality dimensions of the PQML, PUMP, PQEP and EOUM constructs are related to each other within the same construct and these variables are consistent in measuring each of these constructs.

To check whether the research instrument is an effective measure of the CMPQ theoretical construct, we had to check the instrument for *Construct validity* which is established through either discriminant, convergent or factorial validity [36]. The presence of eigenvalues of or above 1, loadings of at least 0.40 and no cross-loadings above 0.40, is confirmation of discriminant, convergent and factorial validity and hence confirms construct validity for EFA method using the PCA technique [29]. As is evident from Table 5, the eigenvalues are above 1.0 and all loadings are above the threshold value and there are no cross-loadings for the 4-factor model used. Therefore the research instrument is an effective measure of the CMPQ construct.

4.3 Confirmatory Factor Analysis

In the previous section a data-driven and theory development method, EFA [37], was used to develop and identify the patterns of relationships between the PQML, PUMP, PQEP, and EOUM constructs and their quality dimensions in measuring the CMPQ construct. To further confirm the identified patterns of relationships between the constructs measuring the CMPQ construct, and test the theory of these relationships, we carried out a Confirmatory Factor Analysis on the data set [38]. CFA being a special case of Structural Equation Modeling (SEM) [39] requires special tools. We used AMOS 18.0 [40] on the data set used in EFA by applying the maximum likelihood (ML) method. We developed two models, a proposed conceptual and theoretical model (Model1) and a competing model (Model2), that included the four identified constructs: PQML, PUMP, PQEP and EOUM as first order factors in the first model and CMPQ as a second-order factor in the second model onto which the four factors in model 1 load. These models are presented in Fig. 1. The second model was to act as a competing model for the first model and is intended to corroborate the four factors in the first model [41]. Results of the CFA analysis are shown in Table 6.

As can be seen CFA results confirm the construct validity and reliability of the result instrument, since the values of GFI, NFI, AGFI are close to the threshold values and Cronbach's alpha for AMOS is above 0.70 [33,35].

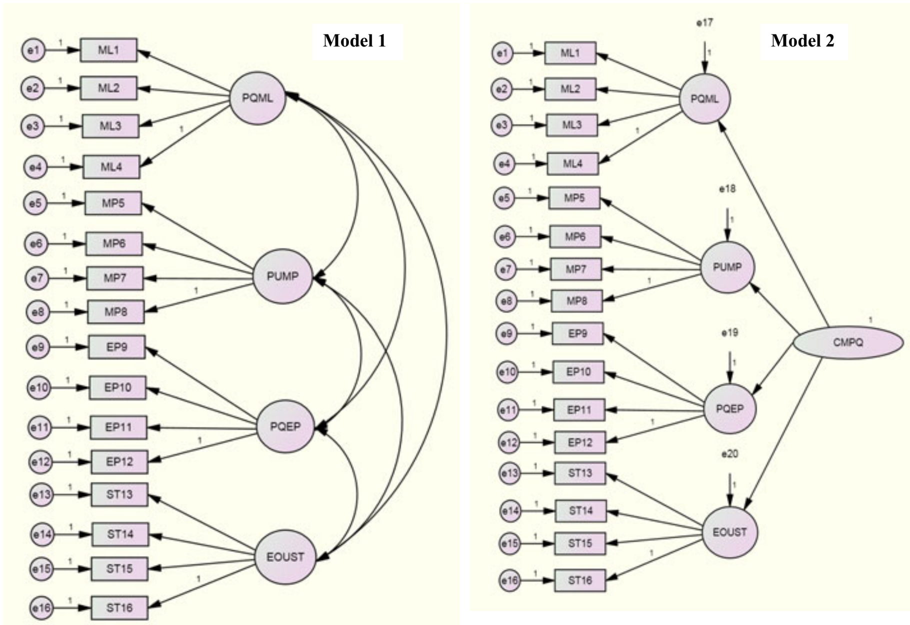

Fig. 1. CFA Models: Conceptual Model and Competing Model

Table 6. Factor Loadings and Model Fit Test Results

Construct	Code	Quality Dimension	Factor Loading		Model Fit Indices		
			Model 1	Model 2	Fit index: Threshold	Model 1	Model 2
	ML1	Understandability	0.76	0.77	χ^2 : SB	142.923	143.738
PQML	ML2	Clarity	0.75	0.75	d.f	98	100
	ML3	Syntax Correctness	0.84	0.84	p-value : $p < 0.05$	0.002	0.003
	ML4	Conceptual Minimalism	0.80	0.80	χ^2/d.f : $1 < \chi^2/\text{d.f} < 3$	1.458	1.437
					RMR : < 0.10	0.129	0.131
					GFI : > 0.90	0.863	0.861
	MP5	Efficiency	0.62	0.62			
PUMP	MP6	Effectiveness	0.94	0.94	AGFI : > 0.80	0.810	0.813
	MP7	Satisfaction	0.71	0.71	NFI : > 0.90	0.837	0.815
	MP8	Commitment & Shared Understanding	0.78	0.78	TLI : > 0.90	0.931	0.912
					CFI : > 0.90	0.942	0.923
					RMSEA : < 0.08	0.066	0.064
	EP9	Product Quality	0.78	0.76			
PQEP	EP10	Understandability	0.77	0.81	AIC : SB	218.923	215.738
	EP11	Modifiability & Maintainability	0.07	0.71	CAIC : SB	358.490	347.960
	EP12	Satisfaction	0.74	0.71			
					Key		
	ST13	Functionality	0.68	0.68	SB : Smaller is Better.		
EOUM	ST14	Usability	0.67	0.67			
	ST15	Satisfaction & Enjoyment	0.80	0.81			
	ST16	Collaboration Communication & Facilitation	0.81	0.80			

5 Discussion

The first observation about the results of CFA is that the (standardized) factor loadings of the the conceptual model (Model 1) and the competing model (Model 2) are close. In fact they are the same for the PQML and PUMP constructs while slight differences are noticed for the PQEP and EOUM. This closeness of the results indicates that the Model used in the EFA was a good conceptual model. To determine the possibility of Model 2 being preferred to Model 1, we compare the model fit indices of both models to determine which ones are near or better than the threshold values (see [29,33,36,41] for these threshold values).

Comparison of the fit indices indicates that the values are close. Model 1 has better fit indices than model 2 for the following indices: chi-square value (χ^2), degrees of freedom (d.f), probability value (p-value), chi-square to degree of freedom ratio ($\chi^2/d.f$), root-mean square residual (RMR), goodness-of-fit index (GFI), Normed fit index (NFI), Tucker-Lewis index (TLI) and comparative fit index (CFI). Model 2 has better fit indices than Model 1 for the following indices: adjusted goodness-of-fit index (AGFI), root-mean square-error for approximation (RMSEA), Akaike information criterion (AIC) and consistent Akaike information criterion(CAIC). The fit index values of both models for RMR, GFI and NFI are below the threshold values. Since the AIC value of Model 2 is better than Model 1, Model 2 is the most parsimonious model [39] and this means is preferred to Model 1. However, looking at the number of fit indices above the threshold values for Model 1 which is higher than model 2, it is equally plausible to conclude that Model 1 is better. We believe reaching a convincing conclusion requires further analysis using different sets of data for the EFA and CFA. We, however, believe that Model 1 could be used if (inter-)dependencies between the quality constructs PQML, PUMP, PQEP and EOUM are of interest. Model 2 could be used if particular explanatory relationships (latent regressions) [39] are postulated among the quality constructs rather than analysing only the (inter-)relationships among the quality constructs as is the case for model 1.

Theoretical and Practical Implications. One of the theoretical implications of this research is that a conceptual domain for the CMPQ construct has been defined based on the modeling artifacts used in and produced during the modeling process together with their quality dimensions. Rather than assessing the quality of the modeling process by defining quality dimensions directly for the CMPQ construct, these could be defined for the PQML, PUMP, PQEP and EOUM and the quality assessed via these constructs. This approach has been operationalized by applying the EFA and CFA methods which have, respectively, produced and confirmed the existence of measurable quality indicators for the four quality constructs. The practical implication of the study is that the developed research instrument offers a means of assessing and measuring the quality of the CMPQ construct. This can be used by collaborative modelers and facilitators to assess their perceived quality, usefulness or ease of use of not only the modeling process and the outcomes, but also the modeling language and the support tool or medium.

Limitations of the Study. The limitation of the study is that we used the same data set for both EFA and CFA. To explicitly determine which of the models presented is better, we need to have two separate data sets. Due to the problem of getting the required minimum sample size this was not possible. Moreover, splitting the one data set into two data sets: one for EFA and the other for CFA was not possible since this would drastically reduce the sample size. Use of students to evaluate the modeling process using the questionnaire could not guarantee that the results were the best we had hoped for. Using professional modelers could be the best but these are hard to come by!

6 Conclusion and Future Research

This paper set out to develop a quality construct for assessing the quality of collaborative modeling through four constructs: PQML, PUMP, PQEP and EOUM. It has been shown through known statistical techniques: Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) that the approach is sound and the research instrument passes the validity and reliability tests. The contribution of this paper is thus two-fold. First, it develops a method of assessing collaborative modeling quality based on modeling artifacts used in, and developed during the collaborative modeling effort. The approach developed assesses users' perceived quality, perceived usefulness and ease of use of the modeling artifacts. Second, a validated instrument for measuring the developed constructs and assessing the quality of the CMPQ construct is presented. We hope this research will stimulate further academic interest in the study of assessing the quality of the collaborative modeling process via the modeling artifacts that are used and produced during the modeling process. Our future research is to further develop and test the theory about the (inter-)dependencies and explanatory relationships between different constructs in relation to the CMPQ construct.

References

1. Barjis, J., Kolfschoten, G.L., Verbraeck, A.: Collaborative Enterprise Modeling. In: Proper, E., Harmsen, F.,, J.L. (eds.) PRET 2009, LNBIP, pp. 50–62. Springer, Heidelberg (2009)
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 2007, pp. 97–106. Tapir Academic Press, Trondheim (2007)
3. 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)
4. Rittgen, P.: Collaborative Modeling of Business Processes-A Comparative Case Study. In: Proceedings of the 24th Annual ACM Symposium on Applied Computing, Waikiki Beach, Honolulu, Hawaii, pp. 225–230. ACM, New York (2009)
5. Vennix, J.A.M.: Group Model Building: Facilitating Team Learning Using System Dynamics. Wiley, Chichester (1996)

6. Vreede, G.J., de Briggs, R.O.: Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Task. In: Proceedings of the 38th HICSS Conference, IEEE computer Society Press, Los Alamitos (2005)
7. Sedera, W., Rosemann, M., Doebeli, G.: A Process Modelling Success Model: Insights From A Case Study. In: 11th ECIS Conference, Naples, Italy, pp. 1–11 (2003)
8. Davison, R.: An Instrument for Measuring Meeting Success: Revalidation and Modification. Information and Management 36, 321–328 (1999)
9. Briggs, R.O., Reinig, B.A., de Vreede, G.J.: Meeting Satisfaction for Technology Supported Groups: An Empirical Validation of a Goal-Attainment Model. Small Group Research 37 (2006)
10. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. MIS Quarterly 13(3), 319–340 (1989)
11. Ssebuggwawo, D., Stijn Hoppenbrouwers, S.J.B.A., Proper, H.A.: Evaluating Modeling Sessions Using the Analytic Hierarchy Process. In: Persson, A., Stirna, J. (eds.) PoEM 2009, LNBIP, vol. 39, pp. 69–83. Springer, Heidelberg (2009)
12. Moody, D., Sindre, G., Brasethvik, T., Sølvberg, A.: Evaluating the Quality of Process Models: Empirical Analysis of a Quality Framework. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, p. 380. Springer, Heidelberg (2002)
13. Nysetvold, A.G., Krogstie, J.: Assessing Business Process Modeling Languages Using a Generic Quality Framework. In: Halpin, T., Krogstie, J., Siau, K. (eds.) Proceedings of the EMMSAD 2005 held in conjuction with CAiSE 2005, Porto, Lisboa, pp. 545–556. Idea Group, USA (2005)
14. Locke, E.A., Latham, G.P.: A Theory of Goal Setting and Task Performance. Prentice Hall, Eaglewood Cliffs (1990)
15. Briggs, R.O., de Vreede, G.J., Reinig, B.A.: A Theory and Measurement of Meeting Satisfaction. In: Proccedings of the 26th HICCS Conference (HICCS 2003), p. 25c . IEEE Computer Society, Los Alamitos (2003)
16. Duivenvoorde, G.P.J., Kolfschoten, G.L., Briggs, R.O., de Vreede, G.J.: Towards an Instrument to Measure the Successfulness of Collaborative Effort from the Participant Perspective. In: Proceedings of the 42nd HICCS Conference (HICCS 2009), pp. 1–9. IEEE Computer Society, Los Alamitos (2009)
17. Paul, S., Seetharaman, P., Ramamurthy, K.: User Satisfaction with System, Decision Process and Outcome in GDSS-Based Meeting: An Experimental Investigation. In: Proceedings of the 37th HICSS Conference (HICSS 2004), pp. 37–46. IEEE Computer Society, Washington (2004)
18. Dean, D.L., Orwig, R.E., Vogel, D.R.: Facilitation Methods for Collaborative Modeling Tools. Group Decision and Negotiation 9, 109–127 (2000)
19. Reinig, B.A: Towards Understanding of Satisfaction with the Process and Outcomes of Teamwork. Journal of MIS 19, 65–83 (2003)
20. Krogstie, J.: A Semiotic Approach to Quality in Requirements Specifications. In: Proceedings of the IFIP TC8 /WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems, Montreal, Canada, pp. 231–249. Kluwer B.V., The Netherlands (2001)
21. Dean, D.L., Orwig, R.E., Vogel, D.R.: Technological Support for Group Process Modeling. Journal of MIS 11(3), 43–64 (1994)
22. de Vreede, G.J.: Collaborative Business Engineering with Animated Electronic Meetings. Journal of MIS 14(3), 141–164 (1997)
23. Fjermestad, J., Hiltz, S.R.: A Decsriptive Evaluation of Group Support Systems and Case Field Studies. Journal of MIS 17, 115–159 (2001)

24. Fjermestad, J., Hiltz, S.R.: An Assessment of Group Support Systems. Experimental Research: Methodology and Results. Journal of MIS 15, 7–149 (1999)
25. Rittgen, P.: Collaborative Modelling Architecture (COMA), http://www.coma.nu/COMA_Tool.pdf (accessed on: 09/08/2010)
26. den Hengst, M., Dean, D.L., Kolfschoten, G.L., Chakrapani, A.: Assessing the Quality of Collaborative Processes. In: Proceedings of the 39th Annual HICCS Conference (HICCS 2006), vol. 1, p. 16b . IEEE Computer Society, Los Alamitos (2006)
27. Green, S.G., Taber, T.D.: The Effect of Three Social Decision Schemes on Decision Group Processes. Organizational Behaviour and Human Performance 25(1), 97–106 (1980)
28. Straub, D.W.: Validating Instruments in MIS Research. MIS Quarterly 13, 47–169 (1989)
29. Straub, D., Boudreau, M.-C., Gefen, D.: Validation Guidelines for IS Positivist Research. Communication of the AIS 13, 380–427 (2004)
30. Lynn, M.: Determination and Quantification of Content Validity. Nursing Research 35(6), 382–385 (1986)
31. Joliffe, I.T.: Principal Component Analysis, 2nd edn. Springer Series in Statistics. Springer, Heidelberg (2002)
32. Tabachnick, B.G., Fidell, L.S.: Using Multivariate Statistics, 4th edn. Harper Collins, New York (2001)
33. Gefen, D., Straub, D., Boudreau, M.-C.: Structural Equation Modeling and Regression Guidelines for Research Practice. Communication of the AIS 4(7), 1–78 (2000)
34. Cortina, J.M.: What is Coefficient Alpha? An Examination of Theory and Application. Journal of Applied Psychology 78, 98–104 (1993)
35. Nunnally, J.C.: Psychometric Theory 2nd edn, McGraw-Hill, New York (1978)
36. Hair, J.F., Tatham, R.L., Anderson, R.E., Black, W.: Multivariate Data Analysis, 5th edn. Prentice-Hall, Upper Saddle River (1998)
37. Brown, T.A.: Confirmatory Factor Analysis for Applied Research. Guilford Press, New York (2006)
38. Albright, J.J., Hun, M.P.: Confirmatory Factor Analysis Using Amos, LISREL, Mplus, and SAS/STAT CALIS. Technical Working Paper. The University Information Technology Services (UITS) Center for Statistical and Mathematical Computing, Indiana University, http://www.indiana.edu/all/cfa/cfa.pdf (accessed on: 17/08/2010)
39. Raykov, T., Marcoulides, G.A.: A First Course in Structural Equation Modeling, 2nd edn. Psychology Group Taylor & Francis Group, New York (2006)
40. Arbuckle, J.L.: AMOS 18: User's Guide. SPSS Inc., Chicago (2009)
41. Tojib, D.R., Sugiant, L., Sendjay, F.: User Satisfaction with Business-to-Employee Portals: Conceptualization and Scale Development. EJIS 17, 649–667 (2008)

A Original Quality Dimensions and Their Groupings

Construct	Original Quality Dimensions	Quality Dimension Groups
PQML	construct deficit, construct overload, construct redundancy, construct excess; expressive power, directness, systematicity; syntax, semantic & pragmatic clarity; modeling primitive adequacy	Understandability, Clarity, Syntax correctness, Conceptual minimalism
PUMP	Efficiency; effectiveness; ease of application, in-out-description adequacy, process & relation description adequacy, method compatibility, interaction & collaboration adequacy, communication & negotiation adequacy; rule & goal commitment, shared understanding	Efficiency, Effectiveness, Satisfaction, Commitment & Shared Understanding
PQEP	correctness, completeness, propriety, clarity, consistency, orthogonality, generality, syntax adherence adequacy, semantics adequacy, pragmatics adequacy; user-comprehensibility; Modifiability, re-usability, flexibility; user satisfaction	Product Quality, Understandability, Modifiability & Maintainability, Satisfaction
EOUM	tool functionality, performance & reliability; efficiency, effectiveness; satisfaction; synchronicity, negotiation/argumentation adequacy, commenting/proposition adequacy, planning/agenda setting adequacy	Functionality, Usability, Satisfaction & Enjoyment, Collaboration & Communication Facilitation

B Research Instrument

Code	Measurement Item	1	2	3	4	5	6	7
	Perceived Quality of Modeling Language (PQML)							
ML1	The modeling language was easy to understand.							
ML2	The modeling language was easy to learn and remember.							
ML3	There are expressions that are not allowed by the modeling language.							
ML4	It was easy to represent all concepts using the available signs and symbols of the modeling language							
	Perceived Usefulness of Modeling Procedure (PUMP)							
MP5	We took a lot of time to negotiate, reach agreement and consensus and at times failed to make important decisions.							
MP6	The modeling procedure enabled us to reach the solution and attain the modeling goal in less time.							
MP7	I was satisfied with the way we communicated/negotiated, reached consensus and agreement and how we made the modeling decisions to obtain the end results.							
MP8	I was in full support of the goals and objectives, had a stake in achieving the goals and objectives of the of the modeling session and contributed to shared understanding.							
	Perceived Quality of End-Product (PQEP)							
EP9	The intermediary and end-products were accurate in depicting all the identified aspects of the domain and only essential details were represented.							
EP10	When I look at the final models I understand the rules and concepts represented and model is easy to understand and explain to those that never participated in the modeling session.							
EP11	It is easy to modify the model to accommodate new changes, to re-use and restructure the model.							
EP12	I am satisfied with the quality of the intermediary and end-products.							
	Ease of Use of Medium (EOUM)							
ST13	The modeling tool functionally supported and aided the modeling process and was reliable in executing the modeling process .							
ST14	The modeling tool was efficient and effective in helping me realize the goals of the modeling process							
ST15	I was satisfied with the functionality offered by the modeling tool, it was fun to use the modeling tool and I enjoyed using it.							
ST16	The modeling tool helped the group to collaboratively develop the model, reach agreement /consensus and our communication was very much facilitated by the modeling tool.							

Scale: 1 = Strongly agree, 4 = Neutral, 7 = Strongly disagre.

Patient Care across Health Care Institutions: An Enterprise Modelling Approach

Sobah Abbas Petersen[1,2], Grete Bach[1], and Astrid Brevik Svarlein[1]

[1] KITH AS, Sukkerhuset, Sverresgt 15, Trondheim, Norway
[2] Dept. of Computer and Information Sciences, Norwegian University of Science & Technology, Trondheim, Norway
{Sobah.Petersen,Grete.Bach,Astrid.Svarlein}@kith.no

Abstract. This paper presents a modelling exercise conducted in the health sector, to identify functional requirements for Electronic Patient Records. The model is based on a holistic modelling approach, Active Knowledge Modelling and is conducted by a multi-disciplinary team with domain and modelling experiences. The main aim of the paper is to share our experience and to highlight the advantages from a holistic approach to modelling. The paper describes the needs of the stakeholders that drove the modelling activity and the design decisions that influenced the model. The model supported a common understanding among the model creators and the data providers. In addition, it helped identify new requirements for Electronic Patient Records that will enable better continuity in care in the health sector.

Keywords: Enterprise Modelling, Active Knowledge Modelling, Model Design, Health Information Systems, Electronic Patient Records.

1 Introduction

Health Information Systems or Medical Informatics is one of the fastest growing areas of Information and Communication Technology (ICT). A number of IT systems are used in the health sector to support both patient care as well as administrative and financial services. One of the central applications of ICT is the Electronic Patient Journals or Electronic Patient Records (EPR), which is an electronic means of documenting and storing patient related information that can be shared among health care professionals. EPR are an essential step to providing an up-to-date and coherent view of a patient's medical history to health care providers.

Norway was among the first to start using EPR and there is much to suggest that Norway may have been the first with almost full coverage of EPR in both the GP services and specialist health care services [1]. However, smooth electronic collaboration and sharing of clinical information across health care institutions is still limited to a few services. The current versions of the commercially available EPR systems are designed to replicate the paper-based patient journals and thus lack the support for health care processes.

The EPR systems in use today are products that are developed by commercial vendors. Different health care institutions use different systems; thus, EPR share a

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 91–105, 2010.

similar interoperability problem as other IT systems in sharing and exchanging information. The lack of interoperability causes problems in the exchange of information across different health care institutions. Interoperability has been defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [2]. This definition of interoperability identifies interoperability at the technical as well as semantic levels. Often, interoperability problems are resolved at the technical level with little or no concern for the actual work processes that these systems support. The notion of information modelling is often used where the focus is on the actual data exchanged. Furthermore, when the work processes span over several enterprises, e.g. different health care institutions as in the case of patient care, there is a need to consider different aspects of the problem such as the processes where the information exchanges across the systems occur. The European Interoperability Framework takes the concept of interoperability further to the organizational level [3]. Similarly Gao and Krogstie also propose the organizational perspective in additional to the technical and operative contexts in modelling [4]. There is a need for EPR to be interoperable at the process and enterprise levels, in order to support health care processes to enable continuity of care across health care institutions.

The aim of this paper is to describe a modelling exercise that was conducted to identify functional requirements for EPR. The contents of the paper are focused towards modellers, particularly in the health informatics domain. The main contribution to the modelling community is the added benefits achieved by taking an enterprise modelling approach rather than an information modelling approach.

The main aim of the model that was developed is to support the identification of functional requirements for EPR that can support planning and implementation of continuity of care processes, in which patient-related information and biological material are exchanged across health care institutions. The request for investigations or laboratory tests is such a process and one that has been identified as needing interoperability among supporting IT systems [2].

The main aim of the modelling exercise is to go beyond information modelling to see the ”context” of the information or the processes where the information exchanges occur. The context is particularly important in complex domains such as health care because there are several actors involved, a number of different resources are needed, the work affects people; it is not only information that is exchanged between processes and institutions, but also physical material and the quality and timeliness of the processes are important. This exercise aimed at understanding the flow of information within a process by analyzing the process and work flow in a holistic way. The modelling exercise and the model contributed to the understanding of the processes in order to identify areas of improvement. They also helped focus on the work processes and the patient rather than the information exchange itself.

This paper describes the case that was examined and the process of creating a model of a health care process. The process of gathering information for the model, the ideas and design decisions behind the model and the factors that affected the design of the model are discussed. The main focus of the paper is on the modelling exercise rather than the description of the contents of the model. In addition to supporting the identification of functional requirements for EPR, the model also helped identify additional roles that can be played by today’s IT systems that are used in the health sector and the desired functionality in existing systems. The experiences from the modelling exercise and the benefits of it and the model are discussed in detail.

The rest of this paper is organized as follows: Section 2 describes our modelling approach, Section 3 describes the case, Section 4 describes our methodology for creating the model and the design of the model, Section 5 provides an analysis of the model that was created, Section 6 describes the evaluation of the model and Section 7 provides a summary of the paper.

2 Background and Modelling Approach

The concept of Enterprise Modelling has been around for sometime; e.g. Vernadat presented the perspectives of production and manufacturing in [5] and a more IT perspective was presented by Fox and Gruninger in [6].

Modelling work is often conducted by IT people for developing IT support, using UML [7]. Thus, there has often been a tendency to reduce the problem and thus, the analysis and solutions to IT applications [8]. Business process modelling is often used to analyze a situation and to propose an improved process. Business process modelling languages such as BPMN [9] have influenced this work. However, even with the added value provided by process modelling methods and supporting technologies, the focus may often be on the implementation.

Enterprise Modelling calls for the analysis of a much larger scope of contents and a multi-dimensional analysis, taking into account the processes, products resources, information elements and possibly others [8]. The concept of Active Knowledge Models (AKM), introduced by Lillehagen, advocates the analysis of several dimensions or aspects of the model and the power of visualization of the model contents [10]. AKM also considers the analysis of several aspects and how they depend on or influence one another. In addition to some of the concepts considered in IT-focussed modelling such as goals (e.g. [11]) or process-oriented concepts such as actions and decisions (e.g. [12]), AKM considers the product (information-based or physical) as important as the process modelled. This calls for a greater analysis of the product.

Several studies conducted in the health care sector for the standardization of information focus on the information exchange and model the information, primarily using UML. Ideas of enterprise modelling have been used by Staccini et al., where they have applied IDEF0/SADT techniques to identify requirements for ICT, (e.g. [13],) and to map care processes to ICT-based services (e.g. [14]). In [15], Jun at al. reviewed different methods that are used for modelling health care processes and identified the strengths and weaknesses of the different methods. They also highlighted which methods are most applicable when. Some of the methods or diagrams presented by them include stakeholder diagrams, information diagrams, process diagrams, swimlane activity diagrams and state transition diagrams. One of their conclusions was that there was no one technique for capturing the essential elements in all the diagrams and combining them. However, the importance of considering a multitude of aspects was confirmed and emphasized and is aligned with the AKM thinking.

In addition to experience in Enterprise Modelling, the AKM methodology encourages an insight into the domain modelled; in fact experience shows that for a successful model, knowledge about the domain is essential. Thus, for our work the modelling team is composed of people with both the domain and modelling experience.

The design of the model discussed in this paper is based primarily on our experience in modelling and the aims of the model rather than the validation of a specific modelling approach or a theory. The main influence in the modelling approach is from the ideas in AKM [10], a holistic approach to modelling. The Business Process Visual Architect from Visual Paradigm [16], based on the BPMN [9], was used to create the model. Thus, the limitations of the functionality of the tool with respect to the AKM methodology hindered the explicit representation and visualization of some of the concepts considered in the model.

3 Case Description

The work described in this paper is a part of the ELINS-2 project, which is a project within a series of projects that are focused on electronic information in the health sector. The long term vision of these projects is to enable the health care systems to plan and implement continuity of care through more effective electronic communication and collaboration between the actors and better knowledge. ELINS-2 focuses on electronic information flow in and across health care institutions, in particular, when two or more institutions are involved, such as the General Practitioner (GP) and the hospital. One of the main aims of this project is to achieve EPR systems that also provide support to the health care processes.

The health care process studied in this paper is the situation in which a health care professional, such as a GP, requires a laboratory test conducted for a patient – a Request for Investigation. A request is made by the doctor and a laboratory form and biological samples are sent to a laboratory in another health care institution. The laboratory test results are sent back to the doctor or the institution that requested for the investigation. An electronic request for investigation, generated directly from the EPR is still in its early stages; in particular, when the receipt of the request and the results of the investigation have to be conveyed back to the origin of the request.

A simplified version of the request for investigation process can be considered as follows: a health care professional would like an investigation, e.g. a laboratory test, to be conducted. Once a request form is created, it is sent to the relevant health care institution for processing; i.e. for the actual samples to be collected from the patient and the analyses to be conducted at the laboratory. Once the results of the required tests are available, the results are conveyed back to the requestor of the investigation.

The process of requesting for an investigation is illustrated in Fig. 1. The person making the request decides upon the test that she wants to be conducted and decides who will conduct the tests before she writes the request form. After the data collection workshop (described in section 4.2), it became apparent that there are situations when the health care professional making the request has to search for specific information. For example, if the condition that the doctor would like investigated is a rare disease or if there are specific requirements or constraints that must be complied with in conducting the test. In such situations, health care professionals sometimes searched on the internet or called around asking for information. Similarly, laboratory personnel also had the same problem in obtaining relevant information. This not only meant that they searched on the internet and various databases, but also numerous telephone conversations took place to share information among health care professionals.

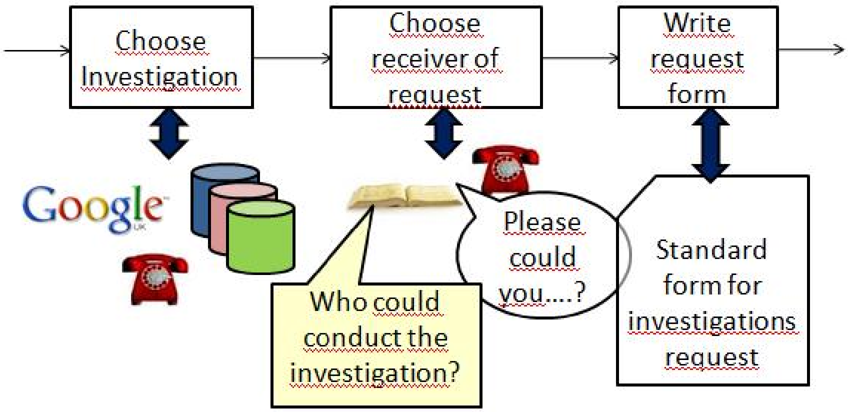

Fig. 1. Current situation when creating a request for an investigation

Currently, the forms used to request for an investigation are standardized and are intended to cover the common investigations that are conducted. If a doctor needs to request for an investigation that is not included in the standard laboratory form, there is no electronic means of including this as a part of the form.

4 Method

The project consisted of five participants, three of whom had a background in the health sector. The creation of the model was a collaborative process among three of the project participants. This was a multi-disciplinary group, where two were experienced health workers (a mid-wife and a bio-engineer) with IT and modeling experience and one with experience in Enterprise Modelling. The data for the model was collected at an interactive workshop with health care professionals. The following steps were followed in creating the model:

- Preparation for data collection and the workshop
- Conduct workshop
- Synthesize data from the workshop
- Create preliminary model and present in a follow-up workshop
- Refine model

4.1 Data Collection

A workshop was held to collect data for the model. Effort was made to include participants that represented all the stakeholders of the process and to represent health care professionals that were involved in all aspects of the request for investigation process. Similarly, effort was made to represent as many of the different hospitals and health care institutions across the country. The participants included a doctor, a nurse, a radiographer and laboratory personnel. Nine participants joined the workshop, which lasted four hours. The workshop consisted of plenary sessions at the beginning and at the end, and a session for interactive information gathering.

At the beginning, the participants were asked to introduce one another as an initiative to facilitate interaction among them. A fictitious scenario of a patient and the request for investigation process was presented to set the scene for the workshop. Five topics, which are also the main processes in the request for investigation, that were identified during the preparations for the workshop were also presented:

1. Preparing the request
2. Taking samples
3. Sending the samples
4. Sending test results
5. Receiving test results

The “Evaluation Café” method was used during the workshop to drive the conversation as well as to ensure contributions from all participants [17]. The Evaluation Café method is a method for group facilitation that allows all participants to contribute their views and have an impact on the outcome of the work. It is a fast, result-driven means to collect information and points of view. It is also a means to initiate a collaborative dialogue and for sharing knowledge and experiences.

The workshop area was set up as a café’ where the participants could move from one table to another and join the discussions. This is illustrated in Fig. 2. The tables were covered in paper tablecloths that the participants could write or draw on. These tablecloths also served as a means of gathering data and as a part of the documentation of the workshop contents. Each table had a facilitator and a topic for discussion. Each participant had 30 minutes at each table, where s/he met a different group every time to

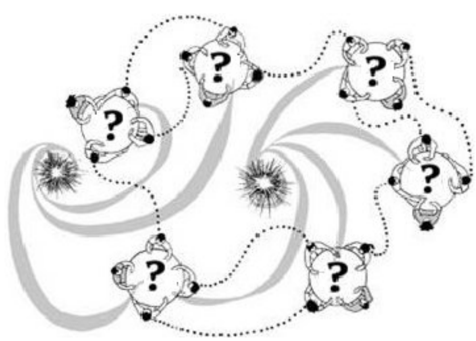

Fig. 2. Illustration of the Evaluation Café workshop method (taken from [17])

discuss one of the topics. The facilitator ensured that the conversation continued to be relevant to the topic under discussion and that each participant was able to contribute their views. The facilitators also made notes during the conversations, which they used in synthesizing the data gathered during the workshop.

At the end of the Evaluation Café session, the participants gathered in a plenary session where a summary of the workshop was presented.

4.2 Synthesize Data

After the workshop, the facilitators summarized the information gathered during the workshop and structured them as follows:

- Current situation – a description of how things were done today.
- Desired situation – ideas for how health care professionals would like things to happen in the future.
- Challenges or hindrances – some of the challenges faced by health care professionals with the current practice.

The information gathered during the workshop was used as the basis for designing and creating our model. Prior to creating the model, the project team discussed the contents and some of the ideas for the model.

4.3 Modelling Process

The model is a result of a collaborative process where the strengths of the three members of the group were used. The modeling process, once the data was gathered from the workshop, is illustrated in Fig. 3. The members with a background in health care started by modelling some of the specific sub-processes, while the member with the experience in Enterprise modeling focused on achieving a holistic model and on how the different topics discussed in the workshop could be combined in a single model, as a generic Request for Investigation process. The specific models were combined into a single model.

A follow-up workshop was conducted with the same participants as the first workshop. The main purpose of this workshop was to verify if we had understood the

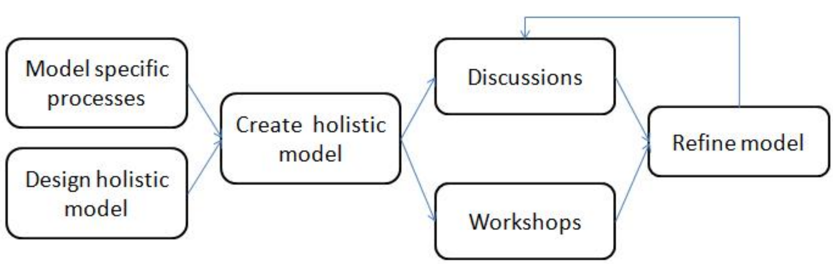

Fig. 3. Modelling Process

process correctly and that the process that we had modelled was not in conflict with the way things could be done. Note that since our task was to model a normative process, it did not reflect the current situation, rather the desired situation.

Based on the feedback from the workshop participants and further discussions among the group, the model was refined.

5 Design of the Model

Our task was to design a "normative" process model; i.e. the aim was not to model the current situation (or as-is); rather to model a process that illustrates the desired process, yet one that can be realized in the near future.

The design of the model was influenced by the outcome of the workshop and the information that was gathered from the workshop. An interesting observation was that the practice in the different health institutions varied. This is perhaps natural as the different institutions had different kinds of patients and their practices were often influenced by the regional differences in terms of geography and availability and accessibility of resources. However, from the perspective of national standardization of practice such that a national health IT infrastructure could support it, this is not only interesting, but also a challenge.

The design was driven by the needs of the health care professionals expressed during the data gathering workshop. These needs may be summarized as follows:

- **Easy access to relevant information.** Health care professionals have a need for easy access to information that are relevant for their work, such as descriptions about specific laboratory tests, overview of where specific health services could be obtained and contact details for them or experiences of other health care professionals that could benefit others. There was also a need for information that was reliable in terms of its quality and source.
- **Status of the request for investigation.** There was a need, in particular from doctors who would be requesting for an investigation most of the time, for the possibility to obtain and follow-up the status of a request that they had made. There was a need to be able to know where in the process as well as the physical location of the request at any point in time and to know as fast as possible when the test results were available. Ideally, access to this information was desired from the EPR. In particular, in situations when the request does not proceed as smoothly as expected, often it was difficult to trace the request. For health care professional, the possibility to have the status of the requests was an indicator of the quality of the service.
- **Documentation of the patient care.** There was a need for better documentation of the patient care process, including processes such as the request for an investigation. Health care professionals would benefit if they were able to obtain an overview of all patient related information from the EPR. Thus, there was need to identify when and what information was either accessed and extracted from the EPR or updated in the EPR at any point in the process.

Based on the above needs, the following design decisions were made for the model:

- To explore the idea of accessing common information sources, wherever possible, rather than transferring information from point to point (or in addition to this, if necessary). For example, make available databases (or health registers as they are often referred to) that can be accessed and shared by a range of health care professionals.
- To explore the idea of a traceability log for the request for investigation, which could provide the status of the request and be accessible to all stakeholders in the process.
- To focus on the information that is required and used in the process, its roles, the type of information (e.g. static or dynamic), its format and where it is created, accessed from and stored.
- To focus on what flows between the sub-processes to identify the requirements from the sub-processes. This comes from the fact that physical material such as blood samples flow between processes and they have to be coordinated with their information counterparts.

6 Analysis of the Model

This section describes the model that was created and how the design decisions that were presented in the previous section influenced the model. The model that was created is fairly large and therefore, it is not possible to include a figure containing the complete model. However, we will attempt to describe important elements in the model, as a direct consequence of our design decisions, using illustrations.

6.1 Access to Common Information Sources

We have identified several points in the process where access to information is necessary for the work. For example, when the doctor prepares the request, she may access a source that provides information about different tests that could be conducted. Similarly, when the nurse or the laboratory staff take samples from the patient and conduct the tests, they may also require access to the same information. Rather than all the different parties searching on the internet or elsewhere for the relevant information, it is desirable that this is available in a (national) database that contains information that are correct, updated and reliable. In contrast to the scenario illustrated in Fig. 1**,** the idea of commonly accessible databases with reliable information not only ensures that all parties access correct information, but it also saves time, avoids duplication and redundancy of information and contributes towards an increase in the quality of the service provided. This idea is illustrated in Fig. 4, where the doctor making the request and the laboratory personnel conducting the tests are able to access the same "Analyses Database" containing information about laboratory analyses. Similarly, making the request form available through the EPR provides information about the test done for a patient for all care providers.

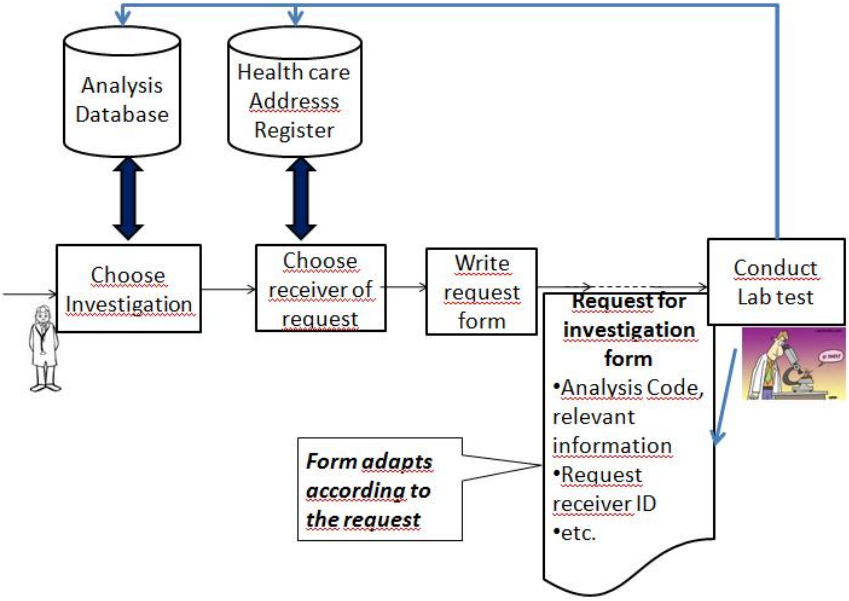

Fig. 4. Desired situation when requesting for an investigation

6.2 Traceability Log

The idea of a traceability log is introduced to keep track of the status of the request for investigation. The main idea here is that all requests for information are registered in a system (either regional or national), where each request has a unique ID, allowing it to be traceable for health care professionals. This will enable all parties involved to access the form as well as obtain the status of a particular request at any point in time. The access to this information is desired from the EPR. A part of the model that was created is shown in Fig. 5, where the main processes for creating a request form and taking samples are shown. This is to highlight that the traceability log and the request or the laboratory form are accessed at several points in the process. The data or information elements are modelled on the top of the model or between two main processes, placed above one another for. The dotted lines linking the processes and the data elements represent access to these elements. The access is not only to obtain status information, but also to access some content from the data element or to add to the contents of the element, e.g. to and from the EPR. It can be seen that the EPR, the laboratory form and the traceability log are accessed at several point in the process.

While the traceability log allows tracking a request and obtaining a status, it also means that several actors along the process have a responsibility to update the log. Based on this need, we are able to identify several functional requirements to the traceability log in order to support successful use of the concept for health care professionals and to support continuity in care for the patients.

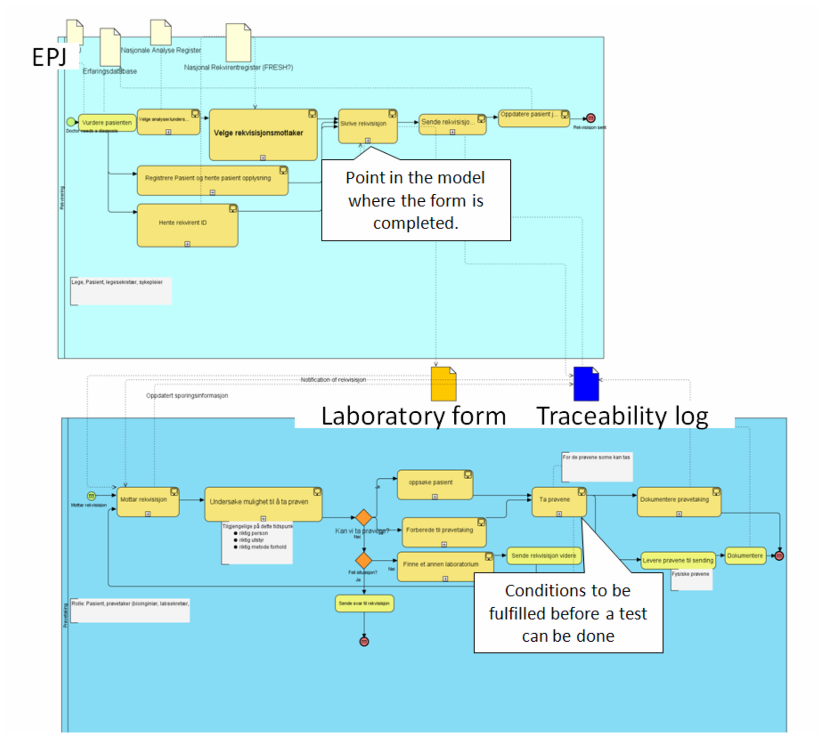

Fig. 5. Overview of a part of the model, highlighting the traceability log (*Note that due to the large model, the text is unreadable. Hence, additional text elements are inserted to highlight the important elements in the model.*)

6.3 Role of Information

One of the main roles of ICT is to support the flow of information. However, by focusing on the role of information in processes provides additional insight into the needs for supporting information flow and documentation of patient related information. For example, being able to distinguish between clinical information about a patient and the administrative information helped us identify when the EPR played a role and what role it played in the process.

An understanding of the information aspects of the model helped us identify when specific information is required or generated. For example, in the top part of the process model shown in Fig. 5, several sub-processes are shown before a request form is completed. These sub-processes describe how a request for investigation form is put together by several bits of information such as the address of the recipient, clinical and administrative information related to the patient, information about the tests to be done. These sub-processes are linked to data elements where these information could be obtained from. Similarly, some of the links indicate points where information is added to the data element, most important of which is the documentation of the patient history in he EPR that continues throughout the process.

By focussing on the information itself enabled us to identify information that is static (at least from the perspective of the request process), e.g. address, dynamic, e.g. patient history or the status of request. Taking this further, we were able to identify that some dynamic information such as the status of the request have a lifetime where it is dynamic and the interest is mostly the log, whereas the clinical information about a patient is constantly updated with a variety of information. The lifetime of this is intended to last as long as the patient lives. The nature of these two types of information and the purpose they serve in the process sets requirements on them and the ICT support that is required. In addition to the status, static and dynamic information, we were also able to identify points where awareness information could be used to let the relevant actors know about certain things, in particular when their attention is required as soon as possible. For example, when a possible laboratory cannot be done, the doctor needs a notification as soon as possible to take remedial action, so that the best possible care is extended to the patient.

Most importantly, by examining the nature of the information through an entire process allowed us to identify the evolution of the EPR that is desirable and the flexibility that is required.

6.4 What Flows between Sub-processes?

It is sometimes important to identify what flows between two sub-process, e.g. information (paper or digital) or material (physical). This is particularly important in the request for investigation process, where the flow of the laboratory form and the physical samples have to be coordinated across health care institutions. This is illustrated in Fig. 6, where a request for investigation is sent along with blood samples from one health institution to another.

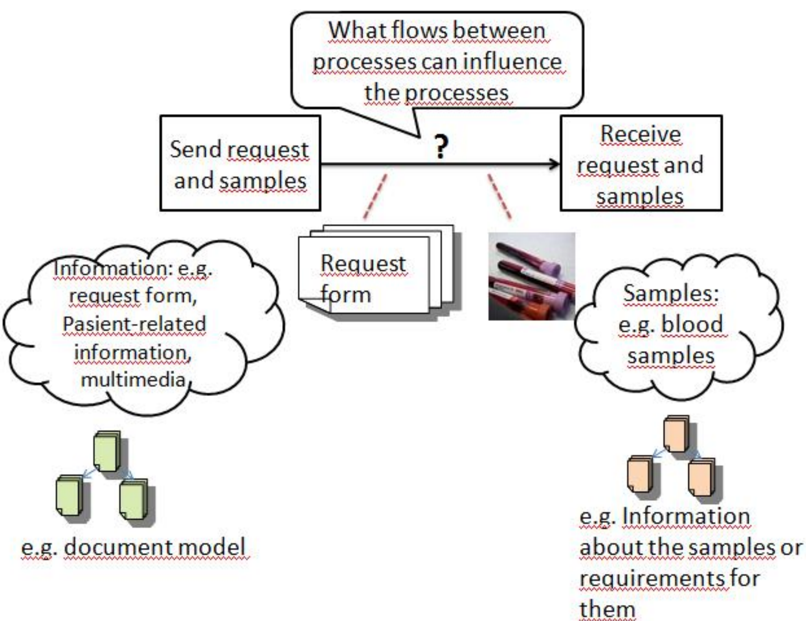

Fig. 6. What flows between sub-processes?

This poses challenges not only in the coordination and tracking of the request, but it may also influence the sub-processes themselves. For example, if the blood samples have to be handled in a special way (information that may be available from the Analyses Database), then it may affect the procedures followed by the laboratory staff and the resource planning they have to do. It may also have requirements and/or constraints on the procedures, competences and facilities for conducting the test. See the point marked "conditions to be fulfilled before a test can be done", in Fig. 5. Thus, taking a holistic approach to the model, it allowed us to identify points where the design of the sub-processes in the model may be affected and how they may be affected.

7 Evaluation

The aim of the model was to support the identification of functional requirements for EPR that provide support for health care processes. Thus, the evaluation of the model was done in two ways: (i) by examining if the model fulfilled its aim and (ii) from the feedback from the follow-up workshop.

We believe that the model does fulfill its aim as it has facilitated us to identify several functional requirements for EPR. Some of the requirements identified are related to the structuring of the information in the EPR as well as role-based access to its contents. The model not only facilitated the identification of new requirements for EPR, but also for other health information systems that currently exist or are desired by health care professional. An example of such a new system would be the traceability log.

None of the workshop participants were used to Enterprise or Process models and thus found the model very complicated to understand or relate to. However, when the rationale behind the model was explained and an overview of the contents was presented, positive feedback was received. The participants were in favour of the ideas adopted for tracing of the request and liked the fact that this would enable all actors at any point in time to obtain the status of the request. They also like the fact that such information was seen as essential information that should be included in an EPR. There was unanimous support that this was a possible way to increase the quality of the care provided for patients.

The main reason why the participants could not understand the model was the fact that the model was presented using the modelling tool, BPVA, assuming familiarity with the notation and large process models. This is perhaps a mistake made by modellers in many occasions. This highlighted the importance of visualization of models and the need for functionality for different means of visualizing models.

8 Summary

One of the main contributions of the modelling exercise was the support for a better understanding of the request for investigation process and a holistic view of the complete process from the beginning to the end. This assisted understanding the actors, the resources, in particular, the information resources and the communication that took place in the process. The model also facilitated a common understanding among the model creators and acted as a basis for discussion of the normative process and the role of the information elements and the role of EPR and ICT in the process. The model also acted as a means of conveying our understanding of the process to the participants of the workshop which facilitated a deeper understanding of the process among everyone. Describing the process at a sufficient level of detail facilitated the refinement of the model.

The model also facilitated the thinking process, specially with respect to detail and allowed us to identify issues and grey areas. For example, a number of legal issues were identified with respect to the privacy of personal data and responsibility. In addition to this, ambiguities in the routines at the health institutions were identified.

By applying a holistic approach to modelling and using ideas from AKM, the model enabled us to identify several functional requirements for future versions of EPR, which will not only serve the purpose of documentation, but will support the work processes of health care professionals and enable better collaboration and coordination in health care. The approach used enabled us to go beyond the scope of the model initially envisaged to obtain a better understanding of not only the request for investigation process, but also other similar processes where information and the patient may move across health care institutions. Our next modelling exercise will be to examine the referral process, where one doctor or an institution refers a patient to another doctor. We plan to use the same design ideas for the model and we feel that the experience has prepared us well for the work ahead.

Acknowledgments. This work has been supported by the ELINS-2 project with financial support from National ICT which is the institution that coordinates the ICT initiative in the specialized national health services. The authors would like to thank Jim Yang for his valuable comments to an earlier draft of this paper, Thorill Antonsen and Anna-Catharina Hegstad for their contributions to this work and the workshop participants for sharing their experiences.

References

1. ERP Monitor (2008), http://www.helsedirektoratet.no/vp/multimedia/archive/00275/EPR_Monitor__Englis_275699a.pdf
2. Benson, T.: Principles of Health Interoperability HL7 and SNOMED - The Book. Springer, Heidelberg (2010)
3. European Interoperability Framework for Pan-European eGovernment Services, Version 1.0 (2004), http://ec.europa.eu/idabc/servlets/Doc?id=19529
4. Gao, S., Krogstie, J.: A Combined Framework for Development of Business Process Support Systems. In: Persson, A., Stirna, J. (eds.) The Practice of Enterprise Modeling, PoEM 2009, pp. 115–129. Springer, Heidelberg (2009)
5. Vernadat, F.: Enterprise Modeling and Integration: Principles and Applications. Springer, Heidelberg (1996)
6. Fox, M.S., Gruninger, M.: Enterprise Modelling. AI Magazine, 109–121 (Fall 1998)
7. OMG: UML Resource Page, http://www.uml.org/

8. Jørgensen, H.D.: Enterprise Modeling - What we have learned and what we have not. In: Persson, A., Stirna, J. (eds.) The Practice of Enterperise Modeling, PoEM 2009, pp. 3–7. Springer, Heidelberg (2009)
9. OMG: BPMN Information Homepage, `http://www.bpmn.org/`
10. Lilleagen, F., Krogstie, J.: Active Knowledge Modelling for Enterprises. Springer, Heidelberg (2008)
11. Horkoff, J., Yu, E.: Evaluating Goal Achievement in Enterprise Modelling - An Interactive Procedure and Experiences. In: Persson, A., Stirna, J. (eds.) The Practice of Enterperise Modeling, PoEM 2009. LNBIP, vol. 39, pp. 145–160. Springer, Heidelberg (2009)
12. Shahzad, K., Zdravkovic, J.: A Goal-Oriented Approach for Business Process Improvement Using Process Warehouse Data. In: Persson, A., Stirna, J. (eds.) The Practice of Enterperise Modeling, PoEM 2009, pp. 84–98. Springer, Heidelberg (2009)
13. Staccini, P., et al.: Modelling Health Care Processes for Eliciting User Requirements: a way to link a quality paradigm and clinical information system design. International Journal of Medical Informatics 64, 129–142 (2001)
14. Staccini, P., et al.: Mapping Care Processes within a Hospital: from theory to a web-based proposal merging enterprise modelling and ISO normative principles. International Journal of Medical Informatics 74, 335–344 (2005)
15. Jun, G.T., et al.: Health care process modelling: which method when? International Journal for Quality in Health Care 21(3), 214–224 (2009)
16. Visual Paradigm, `http://www.visual-paradigm.com/product/bpva/`
17. Weitzenegger, K.: How to run an Evaluation Café ?, `http://www.weitzenegger.de/cafe/index.htm`

The Practice of Competence Modelling

Thomas Albertsen[1], Kurt Sandkuhl[1,2], Ulf Seigerroth[1], and Vladimir Tarasov[1]

[1] School of Engineering at Jönköping University,
P.O. Box 1026, 55111 Jönköping, Sweden
[2] Rostock University, Institute of Computer Science, 18 055 Rostock, Germany
{Thomas.Albertsen,Kurt.Sandkuhl,Vladimir.Tarasov,
Ulf.Seigerroth}@jth.hj.se

Abstract. A clear understanding of the organizational competences of an enterprise and the underlying individual competences and the competence development needs has become more and more important for many industrial areas as a foundation for competence supply processes and adjustment to changing market conditions. Competence modelling, i.e. the use of enterprise modelling techniques for capturing existing and describing desired organisational and individual competences in enterprises, offers important contributions to this. In the last years, the authors of the paper have performed a number of competence modelling cases, which revealed different characteristics and resulted in lessons learned. This paper presents an examination of different characteristics of competence modelling cases, and recommendations and lessons learned from these cases for the practice of competence modelling.

Keywords: Enterprise Modelling, Competence Modelling.

1 Introduction

Application of enterprise modelling can have many different purposes. Two examples of this are capturing and understanding the "as is" processes and organisational structures in an enterprise, and developing and specifying the "to be" situation as support for process improvement of organizational change processes. In such "as is" or "to be" enterprise models, organization structures and roles often are included, but detailed competence and skill descriptions for the roles are not common practice, since this aspect usually does not have priority in improvement processes. However, a clear understanding of the organizational competences of an enterprise and the underlying individual competences and the competence needs has become more and more important as a foundation for competence supply processes and adjustment to changing market conditions. Examples of relevant industrial trends are flexible supply networks, virtual supplier organizations or network-based sourcing [1], which change traditional long-term sourcing strategies to more flexible composition of teams for given projects.

Competence modelling, i.e. the use of enterprise modelling techniques for capturing and describing organisational and individual competences in enterprises, offers important contributions. Research on competence modelling so far resulted in

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 106–120, 2010.

languages and techniques for competence modelling (see section 2), but did not investigate practices. Which are the differences between enterprise modelling practices and competence modelling practices? In the last years, the authors of this paper have performed a number of competence modelling cases, which revealed different characteristics and resulted in experiences. The contributions of this paper are (1) an examination of different characteristics of competence modelling cases, (2) recommendations and lessons learned from these cases for the practice of competence modelling.

The paper is structured as follows: Section 2 will briefly summarize related work in relation to competence modelling. The cases of competence modelling forming the basis for the recommendations and experiences are presented in section 3. Section 4 discusses lessons learned from these cases. Section 5 summarizes recommendations for the practices of competence modelling. Section 6 summarizes the work and presents an outlook to future work.

2 Background to Competence Modelling

2.1 Definitions of Competence

The concept of competence is a rather complex one. In fact there is a large number of definitions of competence and there is none that is adopted by the whole community which spans over several research fields. This makes it important to define concepts as clear as possible. According to Coi [2] there is a distinction between the concepts of *competency* and *competence* where competence consists of three underlying dimensions (competency, context, and proficiency level). Competency represents a skill, e.g. piloting, the context represents the domain in which the skill is performed in, e.g. small civil aircraft, and the proficiency level represents the level at which the competence is mastered, e.g. expert. This makes the definition of competence more reusable since all three dimensions can be separated and reused. Thus, competence can be defined as *"effective performance within a domain/context at different levels of proficiency",* which was originally made by [3] .

There have been efforts made in order to standardise competence models, primarily from three different organisations. HR-XML (developed by the HR-XML Consortium [4]) is a library of XML schemas with focus on modelling of a wide range of information related to human resource tasks. Using such schemas it is possible to define profiles in order to use competency definitions. It specifies data sets like job requirement profiles or personal competency profiles. The former describes competences that a person is required to have. The latter describes competencies that a person has. Such profiles are composed of evidences referring to competency definitions (e.g., IEEE RCD).

The IEEE Reusable Competency Definition [5] is constructed to make it possible to define reusable competency definitions. These definitions are placed in central repositories where they can be reused by communities of competence modellers.

Simple Reusable Competency Map [6, 7] models relationships between competences using directed acyclic graphs. This is an improvement over the IEEE RCD. A map can contain information about dependencies/equivalences among competencies, including composition of simple competences into more complex ones.

2.2 Competence Modelling Approaches

Several models have been developed to provide a systematic evaluation of competences in enterprises. Concerning the evaluation of individual competences, most approaches follow a similar way: (1) the results of socio-human research are used to identify the main models linked to competencies; (2) they provide a formal and qualitative model of the concept of competence; and (3) a mathematical and/or quantitative model is proposed to generate a systematic evaluation of competence levels.

One approach by Harzallah [8] suggests the CRAI model (competence, resource, aspect, individual) associated with axioms based on set theory. The approach aims at describing formal competence in order to provide a mapping between required and acquired competence in an enterprise reengineering context. Competences are characterized by sets of knowledge, know-how and behavior associated to a context and linked to individual actors. Based on a classical evaluation of these characteristics, a mathematical aggregation is suggested to provide a quantitative evaluation of competences.

Another approach is using fuzzy logic for the evaluation of competencies [9]. An aggregated competence indicator is constructed by a fuzzy aggregation of several evaluation criteria through analysis of a work situation using formal tools. The choice of fuzzy methods is motivated by the need to clarify the reasoning of an expert in charge of competency management so that this activity can be at least partially automated. This approach is quite complex and technical but the main point is that it is impossible to define a competency independently of the work situation. This supports the view of the previously described approach where context is an important part of the competency description.

In OntoProPer [10] profiles are described by flat vectors containing weighted skills, which are expressed as labels. Weights represent importance if applied to requirements or skill level if applied to acquired skills. The system itself mainly focuses on profile matching and introduces an automated way of building and maintaining profiles based on ontologies. However, the authors also realizes that maintaining profiles from employees manually is a time consuming task, therefore metadata that is structured according to an ontology and contained in various documents is used. The crawled metadata constitute the foundation for inferences, which derive additional skill data to supplement the explicit skill data in the database.

Very few results concerning evaluation of collective competences have been reported to date. Indeed, socio-human research on collective competence is still recent and the results are not advanced enough to look for industrial engineering-oriented formalization. However, promising trends have emerged: One type of models aims at enabling decision support system to configure groups of actors (teams). Individual competences of actors are mapped against the competence requirements. The configuration of groups of actors aims at a good performance, but the collective competence as such is not evaluated.

2.3 Integrating Enterprise Models with Competence Models

Enterprise modeling is an important tool for strategic planning of any enterprise today. It consists of the process of building models of the whole or a part of an enterprise with process models, data models, resource models, etc. It is based on the knowledge about the enterprise, previous models, reference models and/or domain ontologies. The term "enterprise model" is used in industry to denote differing enterprise representations, with no real standardized definition. Due to the complexity of enterprises, a vast number of differing enterprise modeling approaches have been pursued across industry and academia. Enterprise modeling constructs can focus upon manufacturing operations and/or business operations; however, a common thread in enterprise modeling is an inclusion of assessment of information technology.

Although this is a well known research field and there are several approaches and modeling languages available, competence modelling has not really been included in the enterprise models. To address this issue the Unified Enterprise Competence Modelling Language (UECML) has been developed [11]. UECML has integrated the concept of competence in the language at three levels; the competence itself, individual competence, and aggregated competence of a group of individuals. UECML quite unique to explicitly integrate human competence in enterprise models.

3 Cases of Competence Modelling in Practice

The research work presented in this paper is motivated by a number of real-world cases, three of them were selected for brief presentation in this section.

3.1 Team Formation at An Automotive Supplier

The first competence modelling case is taken from automotive supplier industries and connected to the EU FP6-project MAPPER [12]. MAPPER aimed at capture reusable organisational knowledge to support product innovation in a networked manufacturing organization. At the end of the project, one of the industrial partners, a first tier automotive supplier from Sweden, decided to continue knowledge modelling for certain enterprise purposes. One of the resulting activities aimed at specifically support collaboration set-up and dynamic knowledge sharing with the suppliers of key technologies for economic and ecologic product innovation. In this context, the existing enterprise knowledge models from the MAPPER project, which described the process of innovation and nine of its core tasks, like “establish material specification” or “develop new test method”, were extended and complemented.

The competence modelling performed aimed at documenting the skills and competences needed for the different roles and tasks in future joint projects involving different partners. The modelling focused on competence aspects like required educational background, occupational competences, competences regarding specific design concepts and solutions, and skills of test methods and procedures. The models from MAPPER already included processes, roles, and required resources. The competence modelling extended these models by adding the above mentioned competence aspects for every role. Fig. 1 below shows a small part of the competence model developed. In the upper part, the model of the process for country-specific testing is shown. The lower left section under the process shows the roles involved in the processes; the arrows between role and task indicate the tasks the roles are involved in. For one of the roles, “test specialist (Germany)”, part of the individual

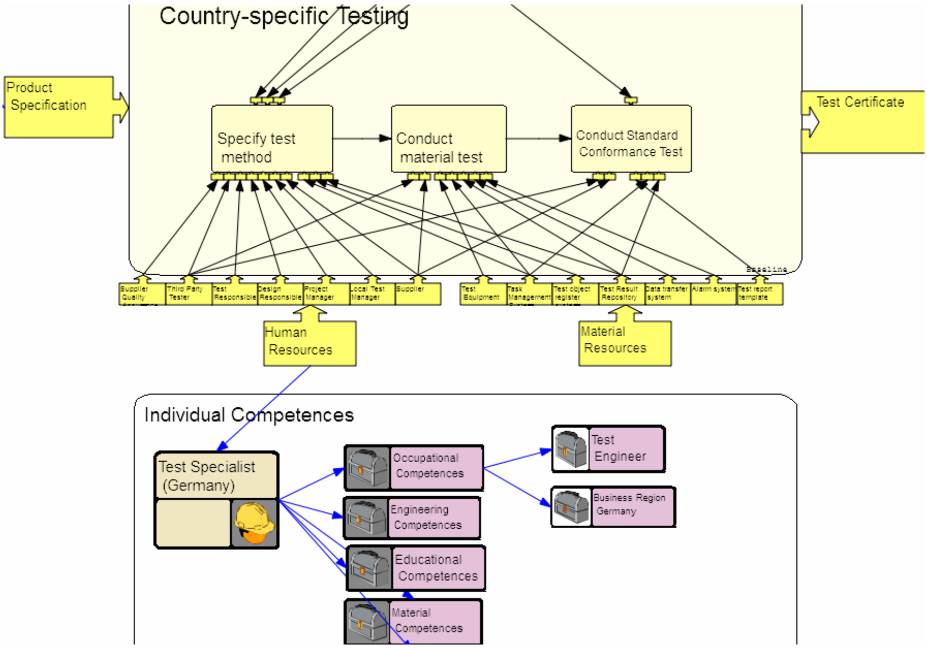

Fig. 1. Excerpt of competence model developed in automotive case

competences are shown (test engineer and business region Germany) etc. The refinements of the different competence areas are not included in Fig. 1 to maintain the understandability.

Since the MAPPER project used the enterprise knowledge modelling method C3S3P [13], we continued with this method during the competence modelling activity. More concrete, an additional iteration of the solution modelling phase was performed. In participatory modelling sessions with representatives from the first tier automotive supplier and the key sub-supplier required for the planned innovation, the activities to be performed by the different roles guided the identification of the competences aspects and skills required. The internal competence catalogues of the two companies formed a useful starting point and structural frame during the modelling, but proved to be not detailed enough and partly outdated.

During the modelling sessions, the results were documented with pen and paper, on a whiteboard or on plastic with sticky notes. After the modelling session, the tool expert and the modelling facilitator documented the modelling results in the METIS[1] tool with the MEAF[2] language. At the beginning of the following modelling session, a walkthrough of the model was used to inform all participants and validate the model.

The competence model served as input for a computerized tool for team formation, i.e. to identify suitable employees in the enterprise for participating in a planned

[1] METIS has been renamed to Troux Architect. Information is available at `http://www.troux.com`

[2] MEAF is the abbreviation for METIS Enterprise Architecture Framework.

project. However, this part of the project was not performed due to cost cutting in the economic crisis 2008/2009. Instead the model was used for documenting best practices, competence development and recruitment, and for training new employees.

3.2 Competence Demand Modelling in Higher Education

This case is one of the deliverables in the project *Competence Modelling and Competence Matching* (KoMo) where the case has served as a test bench for developing a conceptualization of how to perform and represent competence demand modelling. KoMo is a research project that was funded by The Swedish Armed Forces, in particular by the human resource management (FörbePers) and the Command and Control Systems unit (LedSysP). The tasks that we have addressed in this case are structuring, modeling and matching of individual and organizational competences for the master thesis course in higher education. This case has addressed capturing competence demands for different tasks through enterprise models. The case therefore is the base to: (i) describe an approach for competence capture through enterprise models, (ii) present a framework with constructs and relations for modelling and capturing task oriented competence demands, and (iii) describe how context will affect competence modeling.

The final thesis course case was modeled in order to try out an idea and approach for modeling organizational competence demand based on enterprise models. This means that this case had a clear activity oriented approach where the logic between activities in a process was the foundation for addressing competence demand modeling. The competence demand in this case is not defined from the view of a position or role in an organization, which is a traditional way of viewing competence demand. Instead it is defined from the viewpoint that a task has to be performed, in our case giving a final thesis course. A consequence of this view of competence demand is a focus on the organizational competence rather than on the individual competence. In this case we have not focused on finding one person for a certain job, instead what we have tried to find an organization or organizational unit that can perform a task. The size of the organization or unit is decided based on the task and its purpose. An additional consequence is that the context of task is not always the same, in fact it is very common that the context is different every time the course is given, creating different variants of the competence demand. This is something that has proven to be important to capture in order to make it possible to store this knowledge and to create a repository of competence demands. This could make it possible to externalize and reuse knowledge in the organization, i.e. a dimension of knowledge management based on competence demand modeling.

This case was mainly chosen because we as authors of this paper have expert knowledge and extensive experiences from the area (the thesis process in one of the master programs at our university). In the work with this case we have represented different roles: Domain experts regarding the thesis case, Domain experts regarding the KoMo project, Competence modeling experts, Enterprise modeling experts, and Method development experts. The approach for developing this case example has mainly been based on two major activities; modeling seminars and individual work. During the modeling seminars all actors above were involved in the process. The modeling seminars were mainly performed as interactive discussions where we gradually, in argumentative dialogue, developed the example model that captures task oriented competence demands for the thesis process. We performed four modeling seminars and we have conducted individual work between the seminars as input for the next seminar. The individual work has mainly consisted of transforming and formalizing the evolving models into Troux Architect-models. The individual work has also consisted of finding ways to express different principles and relations in Troux Architect-models.

Without giving an exhaustive description of the constituents of the conceptual structure that were developed for this case, activity oriented competence demand modeling, Fig. 2 below at least depicts the main components and their relations.

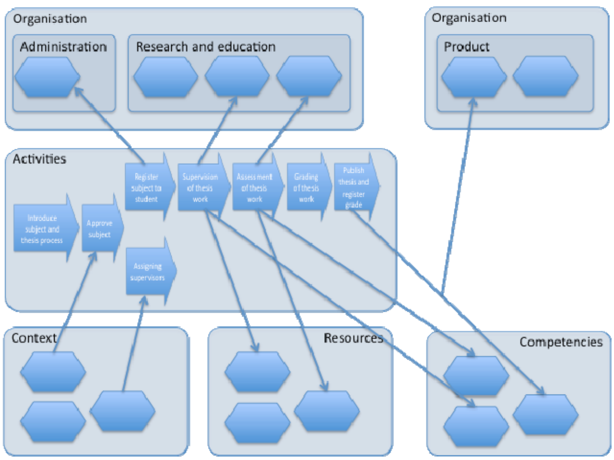

Fig. 2. The overall principle for activity oriented competence demand model

The core of an activity oriented competence demand model is of course the process with its activities. With this as a base we have also identified five other perspectives that together constitutes the whole competence demand model; Organization, Product, Competencies, Context, and Resources. Each of these perspectives represents different views of the competence demand. The activities for a certain task are described in the process view, showing the actual process with its interlinked activities, the competence perspectives provide the descriptions of the different competencies needed for a certain task. The relations between activities and competence will express what competencies are needed for performing a certain activity within a task.

3.3 Modelling Competence of Individuals

The third case is the project ICT-Support for formation of business relationships with developing countries based on immigrant competence (the SPIDER project), which was supported by the Swedish International Development Agency. In this project we aimed at supporting Swedish and Vietnamese companies searching for business partners. The Vietnamese diaspora in Sweden was supposed to act as a business mediator and help with overcoming different obstacles in the business relationships establishment. The competence modelling in this project was thus focused on capturing individual competences of diaspora members such as personal abilities, and knowledge about the required business/industrial sector and the countries.

The modelling process consisted of the following steps. The first one was study of existing approaches to structuring and modelling of individual competences as well as analysis of the case with identification of abilities and skills relevant for the domain. Then we created a semi-structured questionnaire and conducted interviews with diaspora members. After analysis of the interview and structuring of the competences, we constructed a competence model and created personal profiles. The last step was creation of formalization (machine-readable representation) of the model and profiles. The modelling activities were carried out by two modelling experts and one domain expert. The formalization was performed by one knowledge engineer.

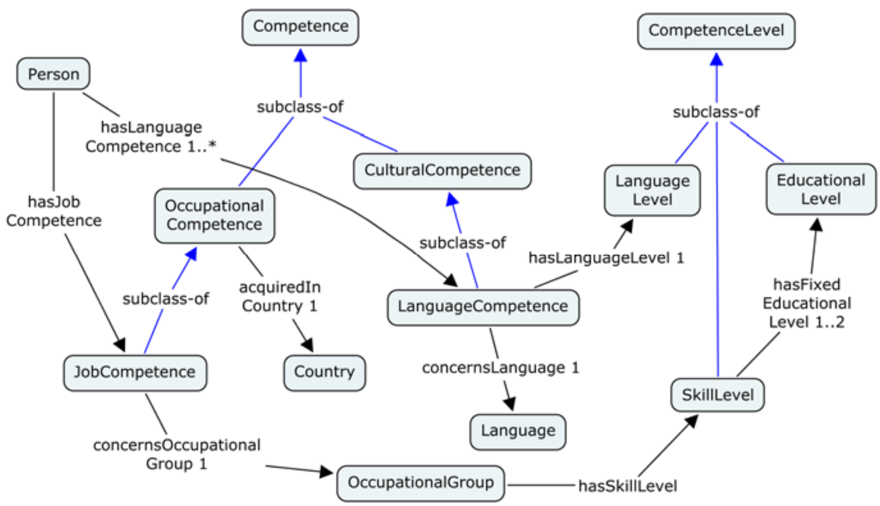

Fig. 3. A fragment of the competence model from the SPIDER project

The core area of modelling was qualification and skills of individuals. The created model includes three major parts: general competences like ability to inform or plan, cultural competences, e.g. command of English, and occupational competences. The latter represents educational background and work experience. All abilities and skills were graded against a scale. A fragment of the model is shown in Fig. 3 above. Competences of each diaspora member were described as a profile created based on the individual competence model (an example is shown in Fig. 4 below).

The model and profiles were implemented as an ontology [14]. During the ontology development process we initially followed Noy & McGuinness's methodology and used the Protégé frame-based ontology language. Recently, the initial competence model has been redesigned to take into account the experience accumulated after the project end. We added the conceptualization steps from the

METHONTOLOGY methodology. The model implementation was also reengineered and the ontology language was changed to OWL (Web Ontology Language) because of good tool support. The ontology was intended to use as a basis for development of an add-in for the existing web applications in Sweden and Vietnam to support companies by suggesting diaspora members who could be of help. Several experiments were made but the applications were not fully completed due to organisational changes and lack of technical competence related to this technology on the Vietnamese side. However, the competence model was used as a basis for competence modelling in another area – collaborative engineering design [15].

4 Lessons Learned and Recommendations for CM

4.1 Comparison of the CM Cases

Table 1 compares the three cases, including the focus of modelling, method, modelling language and application of the results.

Table 1. Comparison of the competence modelling cases

Criteria/Aspect	Automotive Supplier	Higher Education	Migrant competences
CM purpose	Team Formation	Perform task	Competences of individuals
Core area	Product knowledge	Task/activity oriented competence demand	Qualifications and skills
Method / process	C3S3P	Participative modeling	Noy & McGuinness's meth., METHONTOLOGY
Modelling language/ notation used	Metis Enterprise Architecture Framework (MEAF)	Metis Enterprise Architecture Framework (MEAF)	Protégé-Frames, OWL
Application of the results	Documenting best practices, plan competence development, teaching employees	Documentation of action oriented competence modelling, teaching in master course	Use of the model in several other projects as the basis for further modelling

4.2 Lessons Learned

Team Formation at an Automotive Supplier. A lesson learned from the various competence modeling cases is the importance of understanding the core area of the competence modeling as early in the process as possible, i.e. the decisive competence perspective is the activity to be performed, the individual's competence or the product knowledge. The core area affects the further modeling process, the stakeholders to be involved and to some extent other prerequisites of the modeling process. When product knowledge is in focus, like in the automotive case (see section 3.1), understanding product composition, dependencies and commonalities between components, relevant technical parameters and other engineering aspects are of importance and ask for the involvement of product managers and technical experts in the modeling process. Furthermore, the modeling can benefit from knowledge in product data management and product lifecycle management and of knowledge about the industrial domain under consideration. With individual competence as core area, the consistency of the competence model with established enterprise-internal or international competence description "standards" gains importance. This requires an introduction of relevant standards for the modeling team and the involvement of experts in this area. An activity focus often begins like a conventional process modeling, but requires a strict perspective on activities as representation of organizational competences rather than on activities as what is ongoing in an enterprise. In practice, additional validation of the enterprise models and sometimes individual competence focus as continuation of activity focus can be the consequences.

Experience and knowledge in enterprise modeling is an excellent basis. Enterprise modeling requires an understanding of the domain under consideration, which can be achieved by including domain experts in the modeling team. Competence modeling requires also an understanding of different competence modeling traditions, which can occur during the modeling exercise, as experienced in the cases discussed in section 3. In the automotive case, a catalogue with core competences and skills for the automotive supplier under consideration existed. This catalogue had been developed at the Swedish location of the supplier during a period of more than a decade and was updated or changed whenever a new recruiting was planned which showed shortcomings in the catalogue. Compared to this "local" competence structure, other larger enterprises require the use of catalogues defined company-wide, often including different countries of operation. In the thesis case (section 3.2) there is a local formal requirement at the university that the examiner, course manager and tutor should have a PhD degree. This personnel category carries a lot of knowledge and experiences about the thesis process and therefore they need to be involved in the competence demand modeling process. In the case for the Swedish development agency (Section 3.3) the compatibility with international standards was important, aiming at categorizing internationally accepted qualifications and occupation descriptions. As a consequence, the competence modeling team should include at least one member with expert knowledge in how to capture and express competences.

When establishing a competence modeling project, management support and acceptance in the organization is of high importance – like in enterprise modeling projects. But competence modeling requires from our perspective additional efforts in preparing the organizational setting since both, line management, division and human resource management have to be involved. Competence supply, selection and preparation of qualification measures, recruitment and promotion processes, and description of the tasks and responsibilities of a role usually are located in the human resource department. Even though organizational competences often are the responsibility of the line management (or not allocated to any specific role in the organization at all), the human resource department will be able to provide valuable contributions to competence modeling projects.

We observed in many projects that international standards or company internal standards for structuring and representing competences are seen as starting point and as a means to guarantee compatibility. However, these standards were not always of value for the project but sometimes turned out to be rather a "curse" than an asset.

International standards offer from our experience a rather high level categorization of competences, since they form the common denominator for various education traditions worldwide. The attempt to use the standard as basis and add refinement levels might lead to the problem that some competences have be considered as refinement in different categories, which will lead to ambiguities in the categorization. Similarly, company-internal standards can turn out to be inconsistent, outdated, incomplete or all of the before. Cleansing and improvement of such standards can cause more efforts than the actual modeling project.

Competence Demand Modelling in Higher Education. A lesson that can be drawn from this case is the need for both domain experts and modeling experts during the modeling process. The domain expert(s) must, naturally, have sound knowledge and experiences from the domain that is in focus for competence modeling. It is also important that there is coverage of domain knowledge for the whole process. The domain experts must therefore cover the whole process with all tasks and hierarchical representation if this is relevant for tasks in focus. There is also a need for the modeling expert(s) to expand their situational knowledge in relation to the domain in focus. There is some kind of threshold for the modeling expert to climb over in order to be able to ask generative question about the competence demand in relation to a certain task.

The next lesson is that the competence model is modular; parts of the model are optional, although all of them contribute with valuable information. The only parts that cannot be left out, which also are the core in task/activity oriented competence demand modelling, are *the task*, *competence* and *organization*. Without these focal areas it is not possibility to express the needed competencies for a certain task. With these three modules we can express what competencies are necessary to fulfill a task. This is the minimal way of doing activity oriented competence modeling, the benefit is that it is fast and almost immediately gives a task related competence model.

Context is an important perspective to be covered during activity oriented competence modeling. The main reason for this is that there always are a number of situational aspects that will affect the competence demand even if the formal task(s) is the same. It is therefore important to try to characterize the context in terms of different frame conditions. This is also a way to identify different scenarios for a certain task where the competence might differ depending on the frame conditions. The context and frame conditions need to be addressed during the modeling session but they will not be really validated until the task is really performed.

Finally we have formulated some procedural recommendations for the activity oriented competence demand modeling process. The modeling should be an incremental and iterative process where different activities will be generative interdependent to each other. The first modeling task will be to capture the process. What are the activities that will constitute the task, how are the activities interlinked, and what are the main results in the process. The next step should be to address what competences that are needed to perform these activities and where these competencies are located both existing ones and competencies that don’t exist in the current situation. These competencies need to be related to the organizational dimension, which usually is done through roles. It is important to note that the relation between roles and competencies is expressed through the activities and not directly between each other. This is important since we can define and evaluate roles, existing and needed ones, based on requirements that are needed to be able to perform an activity. This is also a way to define requirements on how to organize and to set up a team. These three parts (competence – task – organization) define the core of competence modeling. In order to be able to meet situated dimensions the next step is to address the context. The frame conditions that are identified in the context will be an important source for refinement of the other three core perspectives; competence, task, and organization. Finally, there is a need to deal with the recourses that are needed to perform a task, both existing resources and demand for new or changed ones.

Modelling Competence of Individuals. The first lesson learned from the modeling of individual competences (section 3.3) is that abilities/skills assessment is difficult. In most cases it is not enough to state that a person has certain ability, it is necessary to model the level of this ability. In our project we used combination of self-evaluation and assessment through indirect interview questions. But this approach turned out to be error-prone and inaccurate. More elaborated methods for competence assessment are needed and this calls for contribution of professional psychologists.

The second lesson is that a clear and distinct phase of design is necessary during development of a competence model. The design documentation helps very much in revisions of the competence model, especially when the model is implemented in a computerized language to be part of an application. That is why we utilized the conceptualization steps from the METHONTOLOGY methodology.

One more lesson is that if a competence model is supposed to be a basis for a software application, it is better to use such an implementation language, which has more extensive tool support. Protégé-Frames was flexible for implementation of competence models but the lack of tool support made retrieval of needed competence profiles difficult. When we switched to the Web Ontology Language for implementation of the competence model, we found that existing tools like ontology classifiers (e.g. Pellet) and SPARQL processers simplify search for needed competences significantly. For example, if a person is needed who has experience of repairing cars in Sweden and speaks some Swedish, then it is enough to classify the ontology (model implementation) and run this SPARQL query:

```
SELECT ?person_id
WHERE {
 ?person :hasJobCompetence ?job_comp .
 ?job_comp :concernsOccupationalGroup :OccupationalGroup7231 .
 ?job_comp :acquiredInCountry :Country-Sweden .
 ?person :hasLanguageCompetence
                     [:concernsLanguage :Language-Swedish] .
 ?person :hasPersonID ?person_id .
}
```

The answer found will be “Person C”. The corresponding part of person C’s competence profile is shown in Fig. 4 below (experience of repairing cars is represented by occupational group 7231, Motor vehicle mechanics and repairers).

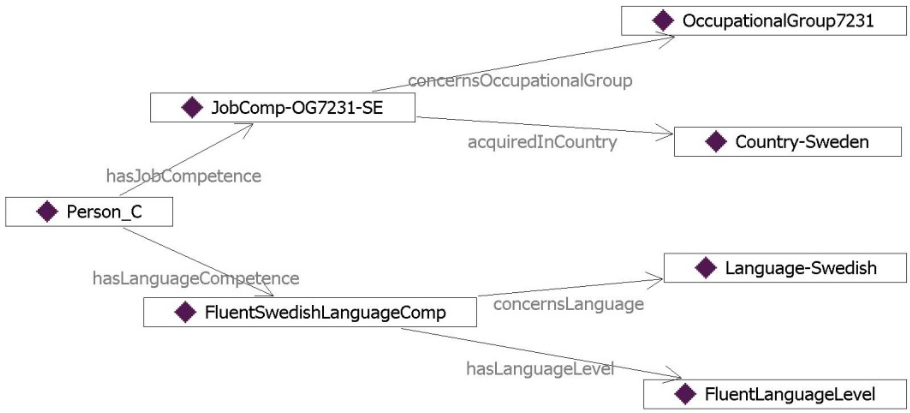

Fig. 4. A fragment of person C's competence profile (the SPIDER project)

The last lesson from this project is that if a competence model is intended to be part of a software application, it is necessary to make sure that needed technical skills exist on site. The lack of such skills was one of the reasons for incomplete use of the model in the SPIDER project. Use of competence models in applications requires advanced technical skills. If such skills are missing, it is necessary to arrange for relevant training for developers.

5 Summary of Recommendations

Recommendation 1: Identify the core area of the competence modeling as early in the modeling process as possible: Activity, product knowledge or individual competence focus?

As a rule of thumb, we recommend the following. If organizational competences are the core interest of the project: chose an activity based perspective first and keep an open mind for individual competences. If the purpose of the project is strongly related to products or their lifecycle, select a product knowledge focus first and try to understand the activities to be performed. If the human resources are of specific interest for the project, start with individual focus of the project and observe the importance of product knowledge in case of manufacturing enterprises.

Recommendation 2: Investigate what ways of structuring and describing competences exist in the scope of the competence modeling project

We recommend involving the human resource department or the responsible manager for this issue. Questions to investigate are ways of describing and structuring competences (local scope, global definition, integration of standards, etc.), existing task and responsibility descriptions for roles, and competence development plans and their basis.

Recommendation 3: Check quality and suitability of competence catalogues or classifications before deciding to apply them

As a simple initial check, we recommend to express the competences for one selected role or activity with the standard under consideration. If such standards shall be applied, the motivation for this requirement has to be analyzed. A requirement regarding compatibility between the competence model to be developed and the existing standard can turn out as severe constraint for the projects, where the effects have to be analyzed. Standard as starting point, without compatibility constraints, can be beneficial for the modeling project.

Recommendation 4: If assessment of individual abilities/skills is necessary, find out reliable methods for doing this beforehand

Assessment of abilities/skills through self-evaluation and interviews is error-prone and inaccurate. More reliable measures are needed that can be created with the help of professional psychologists and human resource departments.

Recommendation 5: Include both domain experts and modeling experts in the modeling process

The domain expert has knowledge and experiences from the domain that are needed for competence modeling. The modeling expert can through this expand their situational knowledge in relation to the domain in focus.

Recommendation 6: Include in a competence model the task, competence, organization, and context.

As soon as a competence model is modular, other parts of the model are optional. The first three mentioned modules are necessary to express what competencies are necessary to fulfill the task. A context can represent a number of situational aspects that will affect the competence demand even if the formal task(s) is the same.

Recommendation 7: Conduct modeling in an incremental and iterative way

The modeling should include several steps addressing the process with activities, competences that are needed to perform these activities, relation between roles and competencies, and recourses that are needed to accomplish a task.

Recommendation 8: When a competence model needs to be implemented in machine-readable form, it is advisable to follow a distinct design phase, choose an implementation language that has good tool support, and make sure that technical specialists skilled in the chosen technology exist on place

The design documentation is needed for revisions of the competence model. If a competence model is supposed to be a basis for a software application, extensive tool support of the implementation language simplifies application development. Lack of relevant technical skills will be an obstacle for application deployment.

6 Conclusions

This paper describes three cases of competence modelling with different core areas: product knowledge, task/activity oriented competence demand, and qualifications and skills of individuals. The cases are analysed and lessons learned are presented. Based on this, several practical recommendation are formulated that can simplify competence model development. These recommendations can be applied to other cases of competence modelling.

Acknowledgements. Some parts of the presented research were financed by the Swedish Armed Forces in the project "Competence modelling and matching (KOMO)", by the European Commission in EU-FP6 project MAPPER and by SIDA in the project "ICT-support for formation of business relationships with developing countries".

References

1. Tapscott, D., Ticoll, D., Lowy, A.: Digital Capital. Harvard Business School Press, Boston (2000)
2. Coi, J.L.D., et al.: A Model For Competence Gap Analysis ∗. Work (2005)
3. Cheetham, G., Chivers, G.E.: Professions, Competence and Informal Learning. Edgard Elgar Publishing Limited (2005)
4. HR-XML. HR-XML Measurable Competencies, `http://www.hr-xml.org` (cited 2009)
5. RCD, I.: IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions (2008)
6. SRCM: Proposed Draft Standard for Learning Technology — Simple Reusable Competency Map (2006)
7. Committee, L.T.S.: Proposed Draft Standard for Learning Technology — Simple Reusable Competency Map (2006)
8. Berio, G., Harzallah, M.: Towards an integrating architecture for competence management. Computers in Industry 58, 199–209 (2007)
9. Pepiot, G., et al.: A fuzzy approach for the evaluation of competences. International Journal of Production Economics 112, 336–353 (2008)
10. Sure, Y., Maedche, A., Staab, S.: Leveraging Corporate Skill Knowledge — From ProPer to OntoProPer 1 Introduction 2 Architecture of our case study. Architecture: 1-12
11. Pepiot, G., et al.: UECML: Unified Enterprise Competence Modelling Language. Computers in Industry 58, 130–142 (2007)
12. Johnsen, S., et al.: Model-based Adaptive Product and Process Engineering. In: Rabe, M., Mihók, P. (eds.) New Technologies for the Intelligent Design and Operation of Manufacturing Networks, Fraunhofer IRB Verlag, Stuttgart (2007)
13. Sandkuhl, K., Lillehagen, F.: The Early Phases of Enterprise Knowledge Modeling: Practices and Experiences from Scaffolding and Scoping. In: IFIP PoEM 2008. LNBIP, Springer, Stockholm (2008)
14. Tarassov, V., Sandkuhl, K., Henoch, B.: Using ontologies for representation of individual and enterprise competence models. In: Bellot, P., Duong, V., Bui, M. (eds.) Proceedings of the Fourth IEEE International Conference on Computer Sciences Research, Innovation and Vision for the Future, RIVF 2006, pp. 205–212. IEEE, Ho-Chi-Minh City (2006)
15. Tarasov, V., Lundqvist, M.: Modeling collaborative design competence with ontologies. International Journal of e-Collaboration: Special Issue on the State of the Art and Future Challenges on Collaborative Design 3(4), 46–62 (2007)

Modeling Network-Based Defence: Success and Failure of an Enterprise Modeling Endeavour

Thomas Albertsen[1], Kurt Sandkuhl[1,2], Ulf Seigerroth[1], and Vladimir Tarasov[1]

[1] School of Engineering at Jönköping University,
P.O. Box 1026, 55111 Jönköping, Sweden
[2] Rostock University, Institute of Computer Science, 18 055 Rostock, Germany
{Thomas.Albertsen,Kurt.Sandkuhl,Vladimir.Tarasov,
Ulf.Seigerroth}@jth.hj.se

Abstract. Research projects have an inherent risk of failure, and learning how to cope with the risk is an important task for everyone involved. In order to do so it is necessary to share the knowledge of the experiences done during and after the project. This paper investigates a recently completed enterprise modeling research project and contributes with lessons learned and recommendations for future enterprise modeling projects.

Keywords: Enterprise Modeling, Competence Modeling, Experience report.

1 Introduction

In the last five years, the practices of enterprise modeling received more and more attention, which includes the use and development of methods, application of specific notations, combination of approaches from different modeling disciplines, participatory modeling approaches, and many more aspects (see [1] and [2] for an overview). In this context, lessons learned from problematic or failed enterprise modeling projects can be considered a contribution to the body of knowledge in the field.

The purpose of this paper is to investigate an enterprise modeling project recently completed in order to understand reasons for problems experienced in the project and potential strategies to avoid them. Many challenges occurring are well known as risks in project management. Lessons learned in this context are why the contingency plans did work. Other challenges originated from the type of the project and knowledge about them might be valuable fur future projects in this area. The contributions of this paper are experiences and lessons learned from an enterprise modeling case.

The paper is structured as follows. Section 2 introduces the background for this work including the enterprise modeling method and tools used. Section 3 describes the project and case in much detail, which originates from network-based defence. Section 4 discusses lessons learned from the project and derives recommendation for future projects. Section 5 summarizes the work and gives and outlook to future activities.

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 121–129, 2010.

2 Background to Enterprise Modeling

'Modeling' refers to a systematic set of actions taken in order "to describe a set of abstract or concrete phenomena in a structured and, eventually, in a formal way. Describing, modeling, and models is a key technique to support human understanding, reasoning, and communication" [1]. Modeling can be used in various domains like mathematics, geology, economics, climate, etc. When applied to enterprise - an organization created for business ventures [6] - it is called Enterprise Modeling (EM) which describes enterprise objectives, activities, information resources, processes, actors, products, requirements, and the like as well as relationships between those entities [2].

The outcomes of Enterprise modeling activity are enterprise models. They are the "representations of the pertinent aspects of an organization's structure and operation" [7]. Presley [7] also define enterprise model as "a symbolic representation of the enterprise and the things that it deals with. It contains representations of individual facts, objects, and relationships that occur within the enterprise" Enterprise models can be both descriptive and definitional with the aim of achieving model-driven enterprise design, analysis, and operation. "From a design perspective, an enterprise model should provide the language used to explicitly define an enterprise. From an operations perspective, the enterprise model must be able to represent what is planned, what might happen, and what has happened. It must supply the information and knowledge necessary to support the operations of the enterprise, whether they are performed by hand or machine. It must be able to provide answers to questions commonly asked in the performance of tasks" [2].

Use and advantages

Once the enterprise models have been created, they can have a variety of usages as well as many benefits for the company.

According to an earlier study made by [7], some examples among many uses of enterprise models are:

- Insight: by abstracting away the complexity of the overall organization, enterprise models can help to improve the understanding and the organization's workings.
- Communication: they can allow all members of the organization to see views of the enterprise based on a common picture.
- Enactment: they can be of great worth in designing and implementing the organization's processes.

Other examples of the use of enterprise models mentioned by [3]:

- Changes of organizational structure: to better suit to relevant business activities.
- Help for management: to gain complete view of the business organization.
- Business process reengineering: in the meaning of efficiency, etc.

As mentioned before, enterprise modeling can have many advantages for each employee in the company and for the entire organization.

Bubenko et al. [1] assert that one of the advantages of enterprise modeling is the effect on the participants. While modeling, the participants can get better understanding about the organization, its main goals, the different processes, how the processes are performed… The participants can also improve their capability to find solutions to problems in a participative way and by consensus of all the participants. Therefore, enterprise modeling enhances communication between the actors and it facilitates the process of organizational learning.

Another advantage of using EM is that it could help to convey semantic unification. It may happen that people in the company use different terms to express the same thing or they can use the same term to express things that are completely different. Enterprise modeling will offer a mutually agreed language to the different actors.

3 The Network-Based Defence Case

3.1 Project Background

Over the last decade the Swedish military has changed from a stable organization with the clear task of defending the Swedish borders, to a much smaller organization with diverse tasks including providing a combat ready taskforce organized under European command, the Nordic Battle Group (NBG). This task force can be deployed within a 600km radius from Brussels on peace keeping missions. The military has also incorporated the network based defence paradigm where all units should concurrently provide data to support a complete view of the situation.

To support competence supply for this leaner organization, the KoMo-project was initiated as cooperation between the Swedish military and the School of Engineering (JTH) of Jönköping University. The project started in May 2006 and had a runtime until the end of 2009. It was financed by the Swedish military and had the purpose of supporting competence supply within the Swedish armed forces by *"developing methods, concepts and techniques for structuring, modeling and matching of competences on individual and organizational level"*. The need for this has become evident within the Swedish military organization since the focus of the military changed rapidly over the last years.

The organisation of the project changed over the time. From the beginning the project had two project owners, the human resource department FörbePers, and the research and development department for human resource information systems LedSystP. At the start-up of the project LedsystP was the main project owner and receiver of the results. However, the organisation of the Swedish military is dynamic and this department was eliminated during reorganisation after two years. The personnel involved in LedSysP was transferred to other departments and the responsibility for the project was moved to FörbePers.

3.2 Project Phases

The project can be divided into two major phases. The first year was focused on quality assurance of an information system under development. The last two years of the project had the purpose of creating concepts and methods for modeling competence demand within the organization.

The evaluated information system was created to support competence supply within the military and also to give future guidelines for development of information systems within the military. This quality assurance had two goals: to make sure that the system was fulfilling the given requirements and to illuminate the present State-of-the-Art within the research community to support the development of the system.

The modeling part of the project had a wide focus in the beginning since there were no clear requirements from the military regarding what the exact purpose of the research should be. The focus was adjusted with the project development. The first intentions were to focus on individual competence and relations between individuals to support formation of teams, but this approach was not well received by some of the stakeholders from the military side. The second approach was to formulate a method for expressing competence demand using enterprise models. This was done using domain experts from the military.

The competence demand approach was to model a part of the NBG from the top of the organization down to a single unit. This was done stepwise starting from the top. The overall organisation in which NBG resides was modelled with a domain expert from the Swedish Army strategic planning department. From JTH two modeling experts where present to facilitate the modeling.

The second step of this approach was to model a unit in the NBG. The purpose of this modeling was to capture the unit's organisation and processes to express competence demand. The unit in question was the an airborne infantry unit and the domain expert was the Commanding officer. The modeling was done by two modeling experts from JTH using interview guidelines to support the modeling process.

After the modeling session the models where created in the software Troux Architect using the Metis Framework.

The results of the two major phases of the project where delivered in several work packages. The first phase produced two separate reports which were well received by LedSysP. The second phase of the project resulted in one big final report which was presented to the stakeholders from FörbePers and the project was finalized.

3.3 Success or Failure?

When evaluating the project results regarding "success or failure?", different perspectives are possible: customer and researcher.

The customer expressed satisfaction with the project results, stated that most of the original goals were reached and that the overall evaluation was positive. The project contributed to the quality assurance of the competence matching platform developed in the first project phase, new ideas and concepts for competence modeling and supply were provided to the military organisation, the competence models developed delivered a proof concept for the new ideas, and all deliverables were provided in time and good quality. However, the basis for this judgement on the customer side is an "adjusted" ambition level in the sense that the customer is aware of the own deficits during the project and accepts that some of the individual goals were not reached since the human resources were not available on customer side.

From a researcher's perspective, the project somehow seems to be unfinished, since the collected data motivating the work could not be validated and the new concepts developed had to be applied in a separate case, which was initiated for this purpose and taken from the own university. Furthermore, the project finished with substantial delay and exceeded the planned budget significantly. All in all, this constitutes a mixture of success and failure.

4 Case Analysis and Lessons Learned

4.1 Typical Project Risks and Unsuitable Contingency Plans

This section will summarize and discuss problems experienced in the project, which basically are a consequence of well-known project risks addressed in the contingency plans, but which still could not be fully avoided. The discussion of these problems will focus on the reasons why the contingency plans were not successful.

Key personnel not available
A typical risk for R&D projects with a runtime of several years is that key personnel might not be available at the required point in time or that they leave the project before it is completed. In the NBD project, this risk became reality on both customer and academic side. At Jönköping University, the PhD candidate who was supposed to be a main player in the project was sick during longer periods of the project, which created a challenge regarding the planning of research work: on the one hand, the NBD project was supposed to be the main source of empirical data for the PhD student and the context for validating new concepts. To deliver the project results without the PhD student would have been feasible, but would have created problems for the schedule of the overall PhD project. We managed to tackle this challenge by shifting the dates for deliverables to a later point in time and by involving additional researchers for certain parts of the project.

At the customer side, the selected colonel with experience in team formation for peace keeping missions was assigned to a task force in Afghanistan. This happened at a point in time, when the main part of the data collection already was completed and only clarifications and additional details were needed. However, the details turned out to be much more complex and important than originally expected. The contingency action was to perform the additional interviews with the colonel by using a well-established video link between Sweden and Afghanistan. After some severe military incidents in Afghanistan, this turned out to be not feasible. When returning from his mission, the project partner on the military side wasn't able to locate the colonel in question in Sweden. When located again, he was on another peace keeping mission, which was completed long after the project's end. Thus, the project had to accept the incomplete empirical data and create a second case.

Our conclusion from this endeavour is simple and consistent with project management recommendations: it is crucial with involvement of at least two domain experts (in our case the colonel with NBD experience) for project-critical subjects.

Organisational changes
Many project management handbooks point out that changes in the customer's organisation structure constitute a project risk, since the support for the project might be affected, and changes in business objectives might be an additional risk. From the

very beginning, the NBD project had two project owners in the military organization sharing the funding and the responsibility (see also section 3). One of these owners was a development project, i.e. it was known that this project would finish some months before the end of the NBD project, the other one an organisation unit. Thus, all steering board and follow-up meetings of the project made sure that always both partners would be involved in order to avoid problems of the shared ownership. However, it turned out that the project owner had a more clearly defined interest in the results of the NBD project and that the organisation unit was keen to learn about new competence modeling opportunities, but did not have so precise objectives. The concrete problem showing when the institutional owner took over was that personnel resources had been allocated to the project, but the individuals supposed to work in the project had not been identified and informed. A new staffing phase began on the customer side, which required time and caused delays.

Our recommendation from this experience: if several project owners are involved, make sure that in addition to the overall (joint) project objectives also owner-specific objectives exist that are clearly expressed. Furthermore, explicitly plan the hand-over process from one owner to the next in the project plan.

Shift in scope

Many R&D projects experience during their runtime a “scope creep”, which often is originated by the knowledge gathered in the project and the necessity to adapt the project plan and the objectives to the new knowledge level. Such an adaptation often is positive for the project as such, since it indicates progress in the actual R&D area, which is welcomed by both customer and developer. However, the challenges caused by such situations with respect to staffing of the project, agreement on changed objectives and preparation of new time plans are comparable with establishing a new project. In the NBD case, the scope creep was basically driven by the researchers since we realized that the progress for the customer would be much bigger when focusing on modeling organisational competences rather than staying on team level. The responsible persons at the customer not only understood and accepted the change, but also supported it actively, since it was in line with other developments in the organisation. The scope shift actually was not perceived as problematic for the project.

Our recommendation is to treat substantial changes in project objective or plans openly with the customer and as thoroughly as a project set-up.

Project delay due to administrative overhead in the military domain

In the list of domain-specific project risks, many textbooks include for military projects that administration overheads can cause delays. Unfortunately, the NBD project confirms this statement to some extent: it was not sufficient for the project owner to identify a suitable person in the military organisation with experience in team formation for military missions, the availability of this person also had to be checked and he had to be released from his duties in the chain of command. The situation would probably have been similar in industrial organisation with several departments.

Table 1. Summary of problems in NBD project

Problem experienced	Original contingency plan	Actual action
Key personnel not available	Staffing of the project with two persons for project-critical tasks	Problem occurred on customer and researcher side; staffing not performed as planned
Organisational changes lead to turbulences in project	Establish support for the project on top level at the customer side in order to have fall-back level in place	Shift project ownership from two to one of the partners
Shift in scope	Establish change request procedure for agreement on new requests	Re-plan project after scope shift
Project delays due to administrative overhead in particular in military projects	Include time buffers in project planning for parts depending on military partner	Time buffers were too small

Table 1 summarizes the problems that occurred, the original contingency plan, and the actual way to manage the situation.

4.2 Challenges Originating from Innovative Modeling Projects

In addition to the project risks introduced in section 4.1, we also collected experiences related to the nature of the modeling project. As described in section 3, the network-based defence project was aiming at exploring innovative ways in competence supply, including new ways to model organisational competence demand and to capture team competences. The innovative character encompassed both, the modeling approach and the thinking and practice in the military domain. Capturing organizational competence demand with enterprise models in order to use this for information searching and team composition was new for the project team involved. Using enterprise modeling experiences as a basis created a good level of confidence that the project team would be able to deliver the expected results, but potential changes regarding modeling approach or required additional elements in the modeling notation were not known at the beginning.

For the military application area, the idea of functional teams representing a specific organizational competence "module" was far from common practice. On the strategic level, this idea was developed and to some extent already accepted, but in the organization in general, it was widely unknown.

This "double innovation" situation was a project risk as such, but this was realized only afterwards since the research side did not understand the conflict potential attached to the functional team idea, which probably was similar on the military side regarding low maturity of the competence modeling idea. On project management and steering board level, an agreement was made what modeling steps to use, how to follow-up and validate the results, and how to disseminate conclusions from the project. During the first follow-up session, when the problems regarding acceptance began to surface, the management level strongly recommended to continue the planned way of working because of the strategic value of the project.

At the end, both innovations survived the feasibility test, but both need additional evaluation. Our conclusion for future projects is to avoid to simultaneously have modeling innovation and domain innovation in a project, where domain innovation may not be well accepted at all organizational levels. Although, in general, modeling innovation and domain innovation can well co-exist together, in some projects it may lead to unnecessary complexity and decreased level of control. It is therefore advisable to first investigate if domain innovation is welcomed or at least accepted at different organizational levels. If it is not, these aspects should be investigated sequentially. Furthermore, we will spend more time on understanding the domain under consideration in situations where domain innovations are possible in order to better grasp the consequences of these innovations.

Another experience during the project, which created problems and is related to the innovation character of the project, was that the possibility that the project in the long run might lead to changes quite instantly created reluctance on the employee side to participate in project related activities. It is well-known that many organizational change projects fail due to missing acceptance of the employees or due to insufficient management support. But it was a new experience for the project team that only mentioning potential organizational innovations already created this reaction. More concrete, during the modeling session with the domain expert, the idea of training and forming smaller “functional” units which provide specific well-defined competences instead of conventional large units caused a long discussion about this issue. Afterwards, the actual modeling purpose was difficult to put back into focus and to complete the modeling exercise became a challenge.

5 Conclusions

In this paper we have discussed success and failure of a project concerned with enterprise modeling of network-based defence. The project aimed at creating concepts and methods for modeling competence demand within an organization. During the project runtime we encountered typical problems of project management as well as more specific difficulties originating from the innovative character of the modeling project.

The standard project management problems were lack of key personnel, organisational changes, shift in scope, and project delays. The actions taken were in line with usual project management recommendations. The lesson learned is that despite the fact that these are well-known project risks and we addressed them in contingency plans in the project, we could not completely eliminate the problems.

The challenges due to the innovative character of the project are more interesting. The problem was that innovations were required in the modeling approach as well as the thinking and practice in the military domain. The lesson learned is that it is better to avoid this kind of “double innovation” in a project if domain innovation is not well accepted at different organizational levels. In such a case, modeling innovation and domain innovation should be investigated sequentially. This will allow for higher level of control and decreased complexity. One more problem was that the project on the long run might lead to changes. The next lesson learned is that even possibility of potential organizational innovations can create negative reaction of the employees and reluctance to contribute to the project.

The case analysis and lessons learned presented in this paper are relevant for modeling projects in the military domain. However, we think that the results can be also applied to industrial organizations with several departments and busy key personnel.

Acknowledgements

Some parts of the research presented were financed by the Swedish Armed Forces with the project “Competence modeling and matching (KOMO)”.

References

1. Bubenko, J., Persson, A., Stirna, J.: Proc. 1. IFIP working conference on the Practice of Enterprise Modeling, PoEM, Stockholm, Sweden. LNBIP, vol. 28. Springer, Heidelberg (November 2008)
2. Fox, M., Gruninger, M.: Enterprise Modeling. The AI Magazine, 109–121 (1998)
3. Madarász, L., Raček, M., Timko, M., Raček, M.: Enterprise modeling and its application in company management systems (2004)
4. Stirna, J., Persson, A. (eds.): Proc. 2. IFIP working conference of the Practice of Enterprise Modeling, PoEM, Stockholm, Sweden. LNBIP, vol. 39. Springer, Heidelberg (November 2009)
5. Wordnet (2010), `http://wordnetweb.princeton.edu/perl/webwn?s=enterprise`
6. Wolverton, M.: Exploiting enterprise models for the automatic distribution of corporate information. In: Proceedings of the Sixth international Conference on information and Knowledge Management, CIKM 1997, Las Vegas, Nevada, United States, November 10-14, pp. 341–347. ACM, New York (1997)
7. Presley, A., Huff, B., Liles, D.: A Comprehensive Enterprise Model for Small Manufacturers. In: Proceedings of the Second Annual Industrial Engineering Research Conference. Los Angeles, CA (1993)

Interactive Goal Model Analysis Applied – Systematic Procedures versus Ad hoc Analysis

Jennifer Horkoff[1], Eric Yu[2], and Arup Ghose[1]

[1] Department of Computer Science, University of Toronto, Canada
jenhork@cs.utoronto.ca, arup.ghose@utoronto.ca
[2] Faculty of Information, University of Toronto, Canada
yu@ischool.utoronto.ca

Abstract. Intentional modeling, capturing the goals of stakeholders, has been proposed as a means of early system elicitation and design for an enterprise, focusing on social and strategic requirements. It is often assumed that more utility can be gained from goal models by applying explicit analysis over models, but little work has been devoted to understand how or why this occurs. In this work we test existing hypotheses concerning interactive goal model analysis via multiple case studies. Previous results have indicated that such analysis increases model iteration, prompts further elicitation, and improves domain knowledge. Results of the new studies do not provide strong evidence to support these claims, showing that such benefits, when they occur, can occur both with systematic and ad-hoc model analysis. However, the results reveal other benefits of systematic analysis, such as a more consistent interpretation of the model, more complete analysis, and the importance of training.

Keywords: Goal Modeling, Model Analysis, Empirical Studies.

1 Introduction

Goal modeling has been proposed as a tool for system analysis and design, increasing awareness of the social-driven goals which motivate system design or redesign in an enterprise, aiming to increase the success of the system in practice. Intentional modeling was included as the first sub-model in the EKD (Enterprise Knowledge Development) method aimed to capture and make decisions over aspects of an enterprise [1]. Goal modeling techniques have been elaborated in software requirements engineering frameworks (e.g., [2] [3]). Certain goal modeling frameworks allow users to capture qualitative, imprecise goals (softgoals), difficult to quantify in early analysis, as well as interactions and dependencies between various stakeholders and systems within an enterprise [4].

Studies have shown the benefits of intentional modeling as part of systems analysis, e.g., [5]. Further work has argued that more utility can be gained from goal models by applying systematic analysis over model constructs, guiding users to use the models to answer questions about the domain. A wide variety of analysis procedures have been introduced, e.g. [6] [7]. One class of analysis procedures allows placement of values to intentions in the model reflecting an initial question, then

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 130–144, 2010.

guides propagation of those values, including ways to deal with conflicting values, ways to backtrack over conflicts, and how to draw overall conclusions. For example, in a model that connects information systems solutions such as CRM (Customer Relationship Management) or predictive analytics to enterprise goals, one might ask "Will adopting CRM improve enterprise agility?" (forward analysis, from means to ends). Conversely, if we want to achieve enterprise agility, "What solutions (among the ones included in the model) should one adopt?" (backward analysis).

Most of the research on analysis procedures focuses on the analytical power and mechanisms of the various analysis procedures, typically demonstrating utility by providing a single example application, often in the context of an industrial project. To our knowledge, little work has been done to study how modelers analyze goal models – to compare ad hoc analysis (without a systematic procedure) with the application of proposed procedures. Without a systematic analysis procedure, the modeler/analyst may be examining the model in an ad hoc manner, possibly mixing forward and backward propagation of values, or assigning values to model intentions without following a predetermined systematic process.

Previous work by the authors provided goal model analysis procedures specifically suited to early stages of enterprise system analysis, supporting qualitative analysis over imprecise concepts, and encouraging iteration over models and elicitation over the domain [8]. Specifically, the work outlined several hypotheses concerning the benefits of qualitative, interactive analysis for agent-goal models:

- **Analysis:** aids in finding non-obvious answers to domain analysis questions,
- **Model Iteration:** prompts improvements in the model,
- **Elicitation:** leads to further elicitation of information in the domain, and
- **Domain Knowledge:** leads to a better understanding of the domain.

An exploratory experiment was conducted to test if these benefits were specific to systematic analysis, or a product of any detailed examination of the model, even if ad hoc. Result did not produce a strong conclusion one way or another, although the experiment suffered from a small number of participants and flaws in the design.

In this work, we designed and administered two types of case studies to further test the hypothesis concerning interactive analysis suggested by previous work. Following our earlier work, we use i* as the goal modeling framework in these studies. Due to the great number of confounding variables, we chose to use case studies as the research method, rather than experiments producing statistically significant data. Specifically, we conducted ten case studies using subjects with some experience in i* modeling. Half of the participants analyzed models using no explicit procedure (ad-hoc analysis) while the other half used implementations of the forward ([8]) and backward ([9]) interactive analysis procedures.

Previous work hypothesized that interactive analysis provokes useful group discussions [8]. In order to gain some insight into analysis by individuals versus analysis in a group, we administered a separate multi-session case study involving a project team designing tool support for modeling "back of the envelope" calculations.

Qualitative and quantitative analysis of results are used to compare treatments in both studies, to gather evidence to support or deny the hypotheses, and to gain an understanding of the benefits of and barriers to systematic goal model analysis, helping to guide the application of goal analysis for systems within an enterprise.

The paper is organized as follows. Section 2 provides background on goal modeling and interactive analysis. Section 3 describes the study design. Section 4 presents results and analysis. Section 5 contains discussion, including threats to validity. Section 6 summarizes related work, while Section 7 provides conclusions.

2 Background: Goal Modeling and Interactive Analysis

2.1 The i* Framework

The i* Framework facilitates exploration of an enterprise emphasizing social aspects by providing a graphical depiction of system actors, intentions, dependencies, responsibilities, and alternatives [4]. An example i* model with a legend is shown in Fig. 1. The social aspect of i* is represented by *actors*, including *agents* and *roles*, and the associations between them. Actors depend upon each other for the accomplishment of *tasks*, the provision of *resources*, the satisfaction of *goals* and *softgoals*. *Softgoals* are goals without clear-cut criteria for satisfaction. Actors have *boundaries* containing the *intentions* of an actor: desired goals and softgoals, tasks to be performed, and resources available. The relationships between intentions inside an actor are depicted with *Decomposition* links, showing the elements which are necessary in order to accomplish a task; *Means-Ends* links, showing the alternative tasks which can accomplish a goal; and *Contribution* links, showing the effects of softgoals, goals, and tasks on softgoals. Positive/negative contributions representing evidence which is sufficient enough to satisfy/deny a softgoal are represented by *Make*/*Break* links, respectively. Contributions with positive/negative evidence that is not sufficient to satisfy/deny a softgoal are represented by *Help*/*Hurt* links.

Analysis labels are used to represent the degree of satisfaction or denial of an intention. Following [2], the *(Partially) Satisfied* label represents the presence of evidence which is *(insufficient)* sufficient to satisfy an intention. *Partially denied* and *denied* have the same definition with respect to negative evidence. *Conflict* indicates the presence of positive and negative evidence of roughly the same strength. *Unknown* represents the presence of evidence with an unknown effect.

2.2 Forward Interactive i* Analysis

The forward analysis procedure starts with an analysis question of the form “How effective is an alternative with respect to goals in the model?” The analysis starts by assigning qualitative evaluation labels to intentions related to the analysis question. These values are propagated along links in the forward direction (i* links are directed) using defined rules. The nature of a *Dependency* indicates that if the element depended upon (*dependee*) is satisfied then the element depended for (*dependum*) and element depending on (*depender*) will be satisfied. *Decomposition* links depict the intentions necessary to accomplish a task, indicating the use of an AND relationship, selecting the "minimum" value amongst intentions in the relation, ordered from satisfied to denied. Similarly, *Means-Ends* links depict the alternative tasks which are able to satisfy a goal, indicating an OR relationship, taking the maximum value.

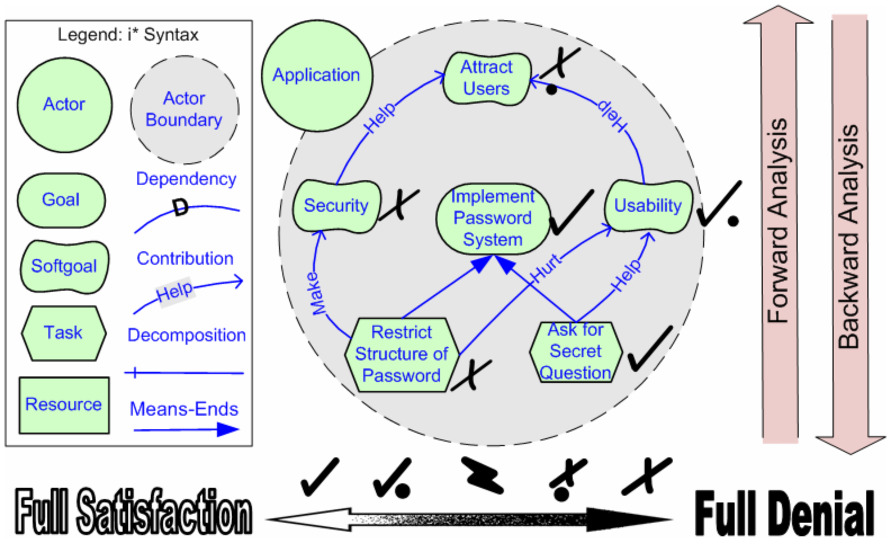

Fig. 1. i* Model Example of a Simple Application with Legend

The procedure adopts *Contribution* link propagation rules from the NFR procedure [1]. Positive values (✓, ✓•) propagated through positive links (Make, Help) produce positive values, weakened with the latter link. Positive values propagated through negative links (Break, Hurt) propagate full or partial negative values (✗, •✗). Links in i* are symmetric: negative values propagated through positive links produce negative values, and negative values propagated through negative links produce positive values.

The procedure prompts for interactive input when human judgment is needed to combine multiple incoming conflicting or partial values to determine the resulting label for a softgoal. Human judgment may be as simple as promoting partial values to a full value, or may involve combining many sources of conflicting evidence.

Once the procedure has finished interactive propagation, the final analysis values for the intentions of each actor are examined in light of the original question. By looking at the degree of satisfaction or denial of key intentions, an assessment is made as to whether the alternative would work in the domain. More information concerning the procedure can be found in [8].

2.3 Backward Interactive i* Analysis

Backward analysis allows users to ask questions of the form "Is it possible for a set of intentions to be satisfied? If so, how?". The procedure uses the same propagation rules as in the forward procedure, but now propagates evidence both forward and backward. The backward propagation is implemented via a formalization of i* expressed in conjunctive normal form (CNF) and passed to a SAT solver. Human judgment is needed for intentions which have conflicting analysis values assigned. In the backward procedure judgment takes the form "I want Intention X to have the value V. Give me a combination of values for the contributing intentions that would result in the target value". Users are shown a list of contributing intentions and their associated links, and then are expected to choose a value for each contributing intention. The user can say "No Combination" if no combination of values would produce the target value. The procedure is iterative in that it repeatedly calls the SAT solver until a satisfying assignment is found and no more human judgment is needed. Each iteration involves more human judgment questions. When judgments produce conflicting results, the procedure "backtracks", re-asking the last round of questions involved in the conflict. Currently, when conflicts occur the user is provided with a list of intentions involved in the conflict (derived via a SAT solver which provides a resolution proof). A solution may not be found if the model is conflicting and does not require human judgment, or if the user cannot enter a human judgment which produces a solution. More information can be found in [9].

3 Case Study Design

We designed and administered ten case studies involving individuals and one multi-session case study with a group of participants, all applying interactive analysis over i* models. In the first type of study, our unit of analysis was the individual participants, while in the second it was the group as a whole. As our aim was for interesting qualitative and quantitative findings without statistical significance, changes were made to the procedure under analysis and to the case study designs at various points. We describe the initial and modified study designs in the following. Study design choices and threats to validity are discussed in Section 5.

3.1 Individual Case Studies

Overview. The studies were administered in two rounds. In the first round, six participants were provided i* refresher training and instructions for the study. They were given an introductory sheet describing the model domain, introduced to the three subject models and twelve analysis questions, then given time to answer the questions over the models. In the second round, four participants were given i* refresher training and study instructions, then spent about 25 minutes creating an i* model about life as a student, and then followed an analysis methodology which guided the application of various questions over the model. In both rounds, half of the subjects used the systematic analysis procedure while the other half answered the questions using ad-hoc analysis. The subjects using systematic i* analysis received an additional round of training for the forward and backward procedures (15 minutes). All participants were told that they could make changes to any model at any point, but that they should not feel obligated to do so. The study involved a "think-aloud" protocol, with one of the authors present to observe the progress and answer questions. Participants were encouraged to ask questions about the model if they had them. Results were recorded via audio recording, screen capture and saving versions of the models. All participants used the i* drawing implementation in the OpenOME tool [10]. Every participant was asked a series of follow-up questions concerning their experience. The total time for each study in both rounds was two hours or less.

Participants. Participants were recruited via a call for participation to students who had learned about i* in one or more system analysis courses, or to students involved in i*-related tool or research projects. Selection was purposive rather than random, we wanted subjects with some knowledge of i* but who were not very familiar with goal model analysis of any form. The resulting participants were students at either the graduate or undergraduate level in Computer Science, Information Systems or Health Informatics. The students had previously created anywhere from one to ten i* models of varying detail, all within the last year. Participants had from none to ten years of industry experience, mostly in technical-related fields. Subjects were paid $40 regardless of the time taken or the results, and results were not made available to anyone who had an influence on course evaluation.

Training. The first two participants of Round 1 were given an i* refresher handout, similar to an expanded version of Section 2.1. The subject using the systematic analysis procedure was given a similar handout describing the forward and backward procedures. After these initial runs of the study, it was apparent that the subject's i* knowledge was not particularly strong. The time devoted to reading the refresher and training documents was not significant. The study was revised such that the facilitator gave a ten-minute i* refresher lesson, and for the participants using systematic analysis, a 10-15 minute instruction session.

Model Domain. In Round 1, subjects were asked to analyze models from the ICSE Greening domain. The models were the result of a study which analyzed the possibilities for "greening" the ICSE'09 conference, conducted via the construction of several medium to large i* models, focusing on the tradeoffs between greening and non-greening goals for the conference chairs [11]. Three models were used from this study, containing between 36 and 79 intentions, 50 and 130 links, and 5 and 15 actors. These were the same models used in the exploratory study described in [8].

The results of the first round of the study performed with six participants showed minimal model changes or elicitation questions, as well as participant difficulties in understanding the models. The decision was made to revise the study and instead allow participants to make their own models over a domain they were familiar with – student life. In the second round, the four participants were provided with some leading questions, (e.g., Who is involved? What do the actors want to achieve?), and then spent 25 minutes creating a model describing their student experiences. P1 to P6 used the ICSE Greening Models, while P7 to P10 used their own student models.

Analysis Questions. In the first round of study, twelve analysis questions over the three models were presented to the participants, four per model, two each aimed at forward and backward analysis. The questions were aimed to represent interesting questions over the domain. For example "If every task of the Sustainability Chair and Local Chair is performed, will goals related to sustainability be sufficiently satisfied?" (forward question) and "Is it possible for both sustainability and successful conference to both be at least partially satisfied? If so, how?" (backward question).

Results from the Group Case study, described in the next section, indicated that it was challenging to motivate modelers to analyze their own models, and that it was sometimes difficult for modelers to come up with interesting analysis questions. As a result, a suggested methodology for model analysis was created using our experiences in evaluating our models in practice. We summarize the methodology in Fig. 2.

1. Alternative Effects (Forward Analysis)
   Identify all leaf intentions in the model (no incoming links)
   a) Implement as much as possible: Evaluate the situation where all leaves are satisfied.
   b) Implement as little as possible: Evaluate the situation where all leaves are denied.
   c) Reasonable Implementation Alternatives: Evaluate likely alternatives in the domain.
2. Achievement Possibilities (Backward Analysis)
   Identify all roots in the model (no outgoing links)
   a) Maximum targets: Evaluate the situation where all roots must be fully satisfied. Is this possible? How?
   b) Minimum targets: Evaluate the lowest permissible values for the roots. Is this possible? How?
   c) Iteration over minimum targets: If a solution was found in b), try gradually increasing the targets in order to find maximum targets which still allow a solution.
3. Domain-Driven Analysis (Mixed)
   a) Use the model to answer interesting domain-driven questions, if possible. If the model cannot answer these questions, can it be expanded to do so, or is it a limitation of the notation?

Fig. 2. Suggested Analysis Methodology

Generally, the first two sections were meant to act as "sanity checks" in the model, checking that it produced sensible answers for a variety of questions, while the second part was intended to support more useful analysis in the domain. Round 2 participants were asked to use this methodology to analyze the student life model they had created. The same methodology was used for all participants, as it did not explicitly reference the forward or backward analysis implementations.

3.2 Group Case Study

A second study was conducted involving a group of four graduate students and a professor who were in the process of designing and implementing a tool (Inflo) to support modeling and discussion of "back of the envelope" calculations. The participants wanted their tool to support informed debate over subjects, such as carbon footprint calculations, containing references to easily understandable models which themselves contain clear references to information sources.

Three two-hour modeling and analysis sessions occurred. Each session had one of the authors present as an i* expert and modeler, and anywhere from two to four of the participant group members. Most of the time in these sessions was devoted to constructing and discussing a large i* model representing the tool, its users, and their goals. During each session, some time was devoted to applying both the forward and backward analysis procedures, letting the participants make decisions over the human judgments posed by the procedures. In this study, the author/facilitator played more of a participatory role, drawing the model and administering the analysis with constant feedback and input from the participants. The first session concluded with a survey concerning the participant's experience with the analysis procedures, while the second and third sessions had audio and/or video recording.

3.3 Data Analysis Methods

The studies produced approximately 24 hours of audio and video, many versions of models, and pages of observer notes. Quantitative data was collected by counting how many and what type of changes to the model were made, (e.g., change a link type, add an intention), and how many domain-related questions were asked for each type of question for each participant (e.g., "What do they mean by collaborate?").

Qualitative data was coded as per the study hypothesis described in the introduction section, allowing for extra fields to capture additional interesting observations. The process of finding results not related to our initial hypotheses was similar to Grounded Theory [12], where qualitative data was grouped according to relevant categories or codes relating to potentially interesting observations or theories. Analysis of further subjects potentially added more evidence to these categories, or produced new categories. What resulted was a list of interesting observations or theories with an associated list of qualitative support classified by participants.

4 Results and Analysis

4.1 Initial Hypotheses

Analysis aids in finding non-obvious answers to domain analysis questions. Results for this hypothesis were mixed. Some participants gave explicit answers to the questions, some referred to analysis labels in the model as answers to the question, while yet others had difficultly producing answers to the questions. One participant was not sure when they were done answering a question. Ideally, participants would be able to interpret the results of the question in the model in the context of the domain; however, only some participants were able to do so. Similarly, participants often had difficulty in translating questions into initial labels in the model.

These results point to a difficulty in mapping the model to the domain, both in starting the analysis and in translating the results back to the world. Presumably, this is a skill which comes with modeling experience. It is interesting to note that these difficulties seemed more prevalent in Round 1 where participants were analyzing large models created by others. It seems that knowledge of i* and the domain may have a significant effect on the ability to apply and interpret analysis.

Model Iteration: prompts improvements in the model. Counts of the number of changes made for each participant are shown in the 3rd and 4th columns of Table 1. Generally, few changes were made with the exception of P1 who redrew much of the model at the start of the study independent of the analysis questions. We omit detailed data on types of changes; however, some specific examples include changing decomposition to contribution links and adding or renaming tasks. Note that a few changes were suggested by the participants but not made, and are not included in the counts. The number of changes was not significant for most participants, and there are more changes made with ad-hoc than systematic analysis. There is no notable difference between participants analyzing their own or others models. For the five participants who made changes, we asked if those changes were helpful, four said yes, while one said it depends on whether changes would be helpful to domain experts.

Table 1. Number of Model Change and Questions Asked for each Participant

		# Model Changes		# Questions Asked		
Treatment	Partic.	Forward Questions	Backward Questions	Forward Questions	Backward Questions	Round
Ad-hoc	P1	59	10	10	1	1
	P4	0	0	1	0	
	P5	5	13	6	6	
	P7	2	5	0	0	2
	P9	0	5	0	0	
Systematic	P2	0	0	2	3	1
	P3	0	0	2	0	
	P6	0	3	5	1	
	P8	0	0	2	2	2
	P10	0	0	0	1	

Elicitation: leads to further elicitation of information in the domain. The number of domain-related questions asked by each participant is shown in the 5th and 6th columns of Table 1. Again, we see no interesting differences between groups.

We can try to understand the results for model changes and elicitation, and why they differ from the results found in previous studies ([8]), by examining the reasoning behind these hypotheses. Previous studies have claimed that it is the interactive nature of the analysis that prompts for changes to the model and drives elicitation. We can expand on this claim, by considering the differences between a goal model representing a domain and the participant's mental model of the domain. An i* model can be considered an incomplete representation of the mental model of its creator. When human judgment is needed in a model, the evaluator is asked to use their mental model of the world to supplement the contents of the physical (explicitly expressed) model. The hypotheses rely on differences between the mental model of the participant and the explicit i* model, especially if they were not the creator of the model. Although such differences could be discovered at any point, they may become particularly apparent when answering human judgment questions.

When these differences are discovered, they may prompt changes to the model, or may cause inquiries concerning the domain. For an example of the former, in the Inflo case study, when asked "Is it possible to make (Inflo) models at least partially trustworthy?" one of the participants decided that validation of a model was not relevant to trustworthy, and the link was removed. The model did not match that participant's mental model of the domain. In other cases, missing elements, inaccurate contributions, or questions concerning the meaning of elements could arise. For example, a human judgment concerning Make conference participation fun made one participant make changes to the model to make the conference more fun and sustainable, renaming task and changing a link from a *hurt* to a *help*.

Because a small number of changes were made to the model, and a modest amount of questions were discovered, we can hypothesize that either the evaluation did not typically reveal differences between the mental model of the evaluator and the explicit model, or these differences existed, but were not used to modify the model. We can find several examples where the evaluator seemed to ignore the structure of the model and answer human judgment questions using only their mental model. For example, in one case, in the forward judgment for the softgoal "Make conference participation fun" the three contributing intentions all contributed partially denied. The participant decided the value was unknown because "I'm not sure how any of these directly related to fun". It seems this would lead to a conclusion that the model is incomplete or inconsistent with the mental model of the participant, and thus needs to be changed, but no changes were made. In another type of example the participants treated the model and judgment situations as an oracle, deferring to the explicit model, "it's telling me that it's weakly satisfied".

A tentative conclusion is that correcting the model and producing questions relies on more extensive knowledge of the syntax, and may require explicit training in detecting differences between physical and mental models. Further studies could continue to test these hypotheses, in different situations, for example with an experienced modeler or in an industrial setting.

Domain Knowledge: leads to a better understanding of the domain. At the end of every individual study, we asked: do you feel that you have a better understanding of the model and the domain after this exercise? Seven out of ten participants said yes. One participant who did not say yes was commenting on the complexity and learning curve associated with i*, another complained that they were already very familiar with being a student, and didn't learn anything further, and the last said that they learned more about the model, but not about being a student. Selection of complex models and a familiar domain seemed to hinder this potential benefit. Analysis was helpful for both systematic and ad-hoc approaches. Participants provided specific comments concerning evaluation: analysis brings out the flaws in the model, and it was helpful for understanding the effects of goals and relations and in choosing between alternatives.

Promote Discussion in Group Setting: Application of systematic evaluation in a group setting did produce several situations where human judgment caused discussion among participants. For example, the participants discussed whether getting feedback was really necessary in order to make models trustworthy after this contribution appeared in a backward judgment situation for Make models trustworthy. In other examples, the group had discussions about the exact meaning of goals appearing in judgments situations, for example "what is meant by Flexibility?" This revealed that different participants thought it meant slightly different things. To be fair, not all judgment situations provoked discussion; more experience is needed to determine how to maximize this positive effect.

4.2 Additional Findings

In addition to findings supporting or denying our initial hypotheses, our qualitative analysis produced other categories of findings, resulting in new tentative hypotheses.

Model Interpretation Consistency. When examining the differences between ad-hoc and systematic analysis, we can see that some participants using ad-hoc analysis made use of the analysis labels and performed some form of label propagation (2/5), while others explained the answer to the question over the model without propagating (1/5), while some participants did both in the same study (2/5). The i* training received by all participants contained an explanation of evaluation labels.

Because the i* Framework was defined in such a way as to leave room for interpretation of its symbols and syntax, by creating systematic procedures we extend the definition of the language, making its meaning more precise. It could be argued that the interpretation used by the analysis procedures is not the best/most obvious; however, what is more important is that i* users and evaluators make consistent and similar interpretations of the model. Thus we are interested in whether or not the participants are consistent with each other (and themselves). Collected evidence shows a variety of interpretations of the model expressed via the propagation of evaluation labels, showing that ad-hoc propagation can be inconsistent among evaluators. For example, one participant interpreted the AND decomposition intentions as having to be at least weakly satisfied for the parent to be satisfied (in the procedure they would have to all be satisfied). In the same model, the participant decided that one intention in another AND decomposition was necessary for the satisfaction of the parent, but the other was optional. In several other cases propagation was consistent with the rules of our procedure. Future studies could ask participants to explicitly propagate in order to collect further examples.

Coverage of Model Analysis. Further analysis of the difference between ad-hoc and systematic analysis revealed significant differences in the coverage of analysis across the model. Subjects who used ad-hoc analysis considered the effects of far fewer intentions and actors in the models. For example, one of the participants who did propagation without systematic analysis ignored the links between the actor under analysis and another actor entirely. Several participants when propagating manually forward or backward only propagated one level or one link jump without continuing to consider the affects of other factors in the model. When participants did not propagate at all they often missed the effects of various links or intentions in their verbal analysis. For example, when considering the satisfaction of goals related to Attendee experience without propagation, a participant only looked at contributions from the Sustainability Chair and did not acknowledging positive effects from goals within the Local Chair.

Although use of the explicit analysis procedures increased the coverage of the analysis, it did not ensure complete coverage. Depending on the choices for initial values, the propagation results often did not cover the entire model. Most participants did not see any problems with such incomplete propagation. If propagation is to be complete more often, more training concerning the selection of initial values and the interpretation of analysis results is needed.

Model Completeness and Analysis. Several participants made interesting comments about the relationship between model completeness and the effectiveness of model analysis. In the Inflo case study, the participants felt that analysis was not useful until the model reached a sufficient level of completeness. One individual participant thought that the study should urge people to make a more complete model before analysis. Another participant said that the model would have been much better if there had been more time to work on it, yet this participant finished creating the model before time was up. For this participant, the analysis revealed that model was incomplete. Another participant, when applying analysis, noticed that the model had no negative links. We can conclude that analysis may be more useful for answering domain questions when the model is complete, but that analyzing over an incomplete model has the potential to reveal its incompleteness.

5 Discussion

5.1 Study Design Selection

Several study design choices were available, the most applicable of which being controlled experiments, action research, or case studies. An experiment would have required the isolation of as many control variables as possible in order to convince the reader that the results in terms of dependent variables (for eg., model changes, questions asked) followed from the manipulation of independent variables (using or not using the procedure, analyzing your own or others models). In the case of goal model analysis, many variables exist which are difficult to control, including: the participants experience with i* and other goal modeling frameworks, their experience with goal model analysis, their experience and openness to modeling in general, their industry experience, and the nature and subject matter of their education. Given that we want to use participants with some i* experience, the second barrier to the application of experiments is finding enough participants to produce statistically significant results. Despite the popularity of i* in research [13], in practice it is not widely used, and a large pool of i* users is not available.

Action research was a further alternative, similar to the types of case studies performed in most work which introduces goal model analysis procedures (for eg. [6] [7]). The forward interactive procedure used in this study has already been applied in one such large-scale study, producing results which led to the formation of the initial hypotheses [8]. Although future studies of this type are useful, we believed it would be advantageous to collect evidence from multiple cases, in an effort to collect a greater quantity of qualitative data. Case studies are useful in that they can provide evidence not only to confirm the existence of hypotheses, but also to explain why such phenomena occur, particularly useful in cases with many confounding variables.

5.2 Threats to Validity

Several threats to the validity of our studies exist.

Construct Validity. We used several measures to test our hypotheses. To test analysis capabilities we looked at how participants were able to use the model to answer questions, whether they could apply some default questions to the models, and whether they could create and analyze their own questions over their own models. However, it was challenging to measure the difficulty participants had in performing these tasks. Often it was hard to determine if the participants were able to take analysis results and use them to draw conclusions over the domain.

To measure model iteration, we counted changes made to the model, or in some cases suggested changes. However, it is difficult to know if these changes are always beneficial. To measure elicitation, we collected questions asked over the model domain during the study. However, classification of questions versus comments can be subjective, and not all domain questions asked over the model would realistically lead to further elicitation. We used a follow-up question to measure improvements in domain knowledge. However, it is difficult to isolate whether analysis was the source of improved understanding and not simply reading or creating the models.

All other exploratory hypotheses are measured using the collecting of qualitative data. This collection can be subjective, although we battled this subjectivity to some degree by having more than one person involved in the data analysis, and by performing systematic classification of qualitative observations.

Internal Validity. We must show that the design of our study adequately tests the initial hypotheses. The extra analysis training given to participants using explicit analysis may have affected the results, although these participants didn't make any more model changes or ask any more questions. Although the study facilitator tried to encourage honest opinions, the presence of one of the authors in all study sessions may have influenced the results. The think-aloud protocol may have affected participant actions, avoiding actions they could not justify. Some of the participants were not comfortable with the think-aloud protocol, and were quiet, making it hard to understand the motivations behind their actions. It is possible that the choices of model domains influenced results, with the domains being too unfamiliar or familiar.

External Validity. As our study used upper-year undergraduate or graduate students as participants, it is possible that results may not generalize to other groups with less technical background. As our studies used the i* framework and interactive analysis procedures, it is questionable whether the results generalize to other goal modeling frameworks or analysis procedures. We believe that results are applicable to frameworks which have a syntax similar to i* (Tropos, GRL). However, it is unlikely that results will generalize to fully-automated analysis procedures.

Reliability. The study was administered by someone with expert knowledge of i* and i* analysis. If the experiment was repeated with someone with less i* or analysis knowledge, the quality of the training or of questions answered in the study may differ, and so may the results. The researcher in question is the creator of the analysis portions of the OpenOME Tool and the Analysis Methodology in question. Some of the potential bias was avoided by having each participant either use or not use the procedures, avoiding an unintentional promotion of one over the other. Every effort was made to avoid influencing the participants during the study; however, it is difficult to avoid all bias or potential effects in such cases.

6 Related Work

We can find examples of studies applying intentional modeling in repeated case studies or experiments. In [1], Stirna and Persson describe multiple participatory cases to illustrate guidelines for participatory Enterprise Modeling (EM). Related work uses two studies to derive conclusions and recommendations about participatory EM and tool support [14]. An interesting conclusion of this work is that EM modeling requires an EM expert. Our findings concerning the need for more extensive i* and analysis training reflect this finding; however, we believe it is too restrictive to say that i* and associated analysis should only be used with an expert present. Existing work shows that even i* novices who misuse the notation benefit from its use [15], and our individual participants generally increased their knowledge of the domain through modeling and analysis.

Work in [16] investigates the role of NFR catalogues in creating i* modeling of a software project using a controlled team experiment. Another study tested the effects of patterns on i* modeling using both a case study in practice and an exploratory experiment in a classroom setting [17]. Similar work in [18] evaluated patterns developed in EKD via workshop experiments involving experienced professionals.

7 Conclusions

In this study we applied interactive i* analysis in ten studies with individual participants and one group study with the aim of testing existing hypotheses concerning the benefits of analysis, and discovering new knowledge about interactive goal model analysis. Despite the small participant sample size, the results are interesting, and not as anticipated. The results can be summarized as follows:

- **Analysis:** Both systematic and ad hoc analysis can be useful for answering analysis questions over the domain, although training is needed to apply initial analysis values and interpret the results.
- **Model Iteration:** Both systematic and ad hoc analysis prompted small amounts of iteration over the model.
- **Elicitation:** Both systematic and ad hoc analysis prompted a small number of questions over the domain. The iteration and elicitation effects observed in previous studies may require explicit training in adjusting the model to match the analysts mental model, and using the model to reveal gaps in knowledge.
- **Domain Knowledge:** Both systematic and ad hoc analysis lead to a better understanding of the domain.
- **Model Interpretation Consistency:** Ad hoc analysis will often use interpretations of the model which are inconsistent within one analysis and amongst modelers. Use of systematic analysis promotes a consistent interpretation of the model.
- **Coverage of Model Analysis:** Systematic analysis increases the coverage of intentions and actors considered in answering analysis questions.
- **Model Completeness and Analysis:** A certain level of model completeness may be necessary for effective analysis. In some cases analysis may reveal the incompleteness of the models.

Study results can guide the application of intentional modeling and analysis in an enterprise setting, illustrating the potential benefits of ad-hoc vs. systematic analysis and emphasizing the role of training or the presence of an experienced facilitator.

Acknowledgments. Financial support has been provided by the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship Program.

References

1. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
2. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Norwell (2000)
3. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-Directed Requirements Acquisition. Science of Computer Programming 20, 3–50 (1993)
4. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE 1997), pp. 226–235. IEEE Press, New York (1997)
5. Maiden, N.A.M., Jones, S.V., Manning, S., Greenwood, J., Renou, L.: Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 367–383. Springer, Heidelberg (2004)
6. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Simple and Minimum-Cost Satisfiability for Goal Models. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 20–35. Springer, Heidelberg (2004)
7. Franch, X.: On the Quantitative Analysis of Agent-Oriented Models. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 495–509. Springer, Heidelberg (2006)
8. Horkoff, J., Yu, E.: Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. In: Persson, A., Stirna, J. (eds.) PoEM 2009, LNBIP, vol. 39, pp. 145–160 (2009)
9. Horkoff, J., Yu, E.: Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. In: Parsons, J. (ed.) ER 2010. LNCS, vol. 6412, pp. 59–75. Springer, Heidelberg (2010)
10. OpenOME, `https://se.cs.toronto.edu/trac/ome/wiki`
11. Cabot, J., Easterbrook, S., Horkoff, J., Mazon, J., Lessard, L., Liaskos, S.: Integrating Sustainability in Decision-Making Processes: A Modelling Strategy. In: ICSE 2009 New Ideas and Emerging Results (NIER 2009).
12. Seaman, C.B.: Qualitative Methods in Empirical Studies of Software Engineering. IEEE Trans. Softw. Eng. 25(4), 557–572 (1999)
13. Fourth International i* Workshop (iStar 2010). CEUR Workshop Proceedings (2010), `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-586/`
14. Persson, A., Strirna, J.: An Exploratory Study into the Influence of Business Goals on the Practical Use of Enterprise Modeling Methods and Tools. In: Proc. 13th Int. Conf. on Information Sys. Development (ISD 2004), pp. 275–288. Springer, Heidelberg (2002)
15. Elahi, G., Yu, E., Annosi, M.C.: Modeling Knowledge Transfer in a Software Maintenance Organization – An Experience Report and Critical Analysis. In: Stirna, J., Persson, A. (eds.) PoEM 2008, LNBIP, vol. 15, pp. 15–29 (2008)
16. Cysneiros, L.M.: Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements. In: Proc. Workshop em Engenharia de Requisitos (WER 2007), pp. 107–115 (2007)
17. Strohmaier, M., Horkoff, J., Yu, E., Aranda, J., Easterbrook, S.: Can Patterns improve i* Modeling? Two Exploratory Studies. In: Paech, B., Rolland, C. (eds.) REFSQ 2008. LNCS, vol. 5025, pp. 153–167. Springer, Heidelberg (2008)
18. Rolland, C., Stirna, J., Prekas, N., Loucopoulos, P., Persson, A., Grosz, G.: Evaluating a Pattern Approach as an Aid for the Development of Organisational Knowledge: An Empirical Study. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 176–191. Springer, Heidelberg (2000)

Adapting UML Activity Diagrams for Mobile Work Process Modelling: Experimental Comparison of Two Notation Alternatives

Sundar Gopalakrishnan, John Krogstie, and Guttorm Sindre

Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)
Sem Sælands Vei 7-9, 7491 Trondheim, Norway
sundar@idi.ntnu.no, krogstie@idi.ntnu.no,
guttorm.sindre@idi.ntnu.no

Abstract. Even if geographical aspects such as location is included in several enterprise architecture frameworks [15], enterprise modelling notations seldom capture the "where" aspect, such as the location for performing some activity in a business process. However, for mobile information systems it is often relevant to model where something is supposed to take place. In a previous paper, we suggested some alternatives for small modifications to UML Activity Diagrams to address this, but then only comparing the alternatives analytically. In this paper, we report on a controlled experiment comparing the two most promising notations from the previous paper, one adding location to the activity diagrams by annotations, another indicating location by colour. The experiment investigated both the participants' opinions about the notations and their performance on some tasks requiring understanding of the models. For opinion there was no significant difference, but for task performance there was a significant difference in favour of the notation using colour.

Keywords: Requirements specifications, mobile information systems, model-based development, UML activity diagram, enterprise modeling.

1 Introduction

Mainstream process notations used in IS modelling tend to ignore the "where" aspect. For instance, BPMN [1] and UML activity diagrams [2] capture what (objects), how (sequence and parallelism of activities and decisions), who (swimlanes), when (time triggers and time events), and to a very limited extent why (e.g., how a decomposed activity diagram satisfies a higher level activity) - for the latter some extensions with process goals have also been suggested [3] - but not the location of the activities performed. With a view on traditional information systems, where work is performed by people sitting in their offices using desktop computers, the neglect of physical location is understandable - it is much more important whether a task is performed by the purchasing or salary department than whether the worker is sitting in office 221 or 325. Hence the usage of swimlanes to denote organizational placement rather than geographical placement is easily justified.

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 145–161, 2010.

For mobile information systems, however, the location and context of activities performed may be of major importance [4]. Whether a certain task should be performed in the office before going to a client, in the car while driving, after arriving to the client - or possibly any of these places, up to the personal preference of the worker - could have a large impact both on quality, efficiency, worker satisfaction and customer satisfaction, and would therefore be an important process design decision. In turn, this decision would also have a lot of impact on what applications would have to be developed to support the work process, and what requirements that these applications would have to satisfy. For instance, if the task were to be performed while driving a car, this would imply quite different usability requirements than what a desktop application is normally faced with.

It is therefore interesting to look at the possibility of adapting mainstream process notations such as UML activity diagrams to also be able to capture the location of activities. A minor adaptation of a mainstream notation like UML seems more tempting than inventing an entirely new notation because industry is more likely to pick up a notation they are already largely familiar with. Relating to UML, there is also a long tradition in providing modelling profiles of this sort, having small extensions to the core notation. In previous work [5] we presented a number of notation ideas for including location in UML diagrams and compared them analytically using some examples. Some of the notations turned out to be clearly inferior, either having lacking expressive power or economy or becoming messy due to a high number of crossing lines. Two of the proposed notations came out as more promising than the others, one adding location/context by means of annotations (i.e. UML notes) and the other using colour. The rest of the paper is structured as follows: Section 2 presents the result from our prior work and some related work in this area, section 3 discusses the research method, and section 4 presents the experiment results. Section 5 provides analysis and discussion, and section 6 concludes the paper. More insignificant results of this experiment are documented in appendix.

2 Prior and Related Work

As already stated in the introduction, two of the proposed notations came out as more promising than the others.

The two alternative notations are shown in Fig 1, both capturing part of the work process within a home care unit, offering practical help and home nursing care to its clients. In the ‘Mobile Care’-project, it is planned to better support the mobile aspects of the home care service by providing the employees continuous access to the central health information system (software used in PDA to log/receive info) and other relevant systems from wherever they are using a combined PC/PDA-solution. This is related to the ‘Wireless Trondheim’-project [6], which is currently managing and extending a mobile broadband (WLAN) infrastructure for Trondheim. The shift leader distributes patient visits on available personnel in the morning meeting, each homecare assistant then decides on the sequence of visits to be made while still in the office. Then while driving to the patient's home, the assistant prepares for the visit by obtaining some information about the patient (typically through an audio interface, to

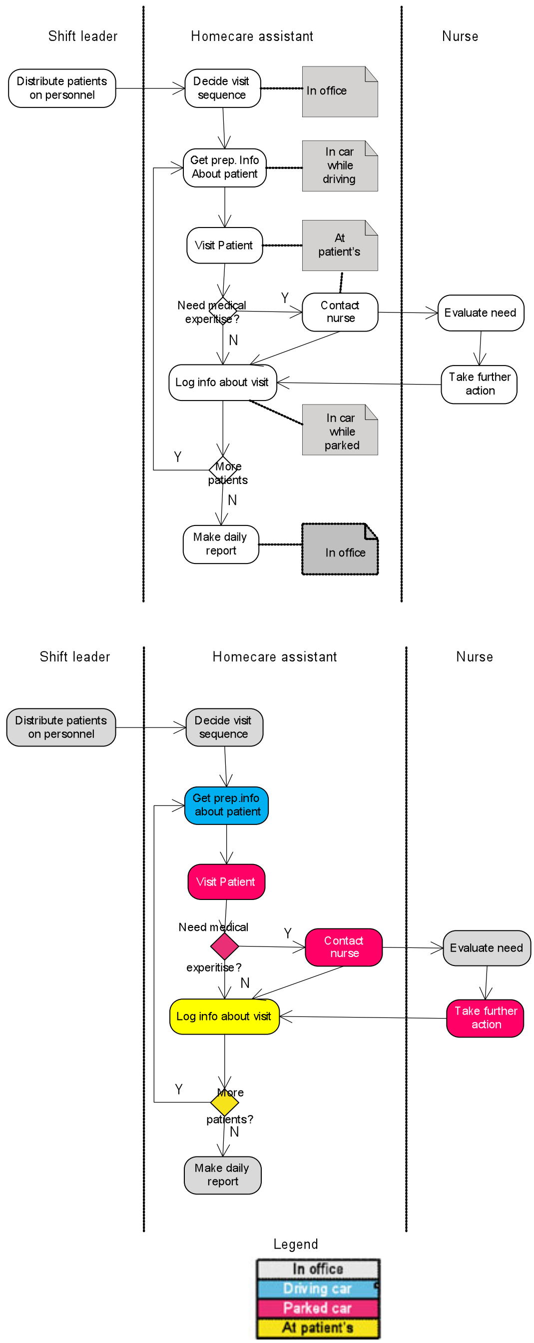

Fig. 1. The two alternative notations – Home Care Case

be less disruptive for concentrating on the driving). Normally, the patient only needs help with day-to-day activities (e.g., shopping, cleaning, taking the right amount of medication), but in case there are some health complications that the assistant cannot handle, a nurse is contacted. Using a system called Gerica accessible by her PDA, the home care assistant can log information about patients on the go. If the health care assistant needs further medical expertise he/she can request help from the nurse at hospital through logging in information via Gerica. The nurses at the hospital get the request and provide further info/advice to be followed by the healthcare assistant (HCA). Finally the HCA finishes her job by reporting at the office. Locations (e.g., office, car) and context (e.g., parked, driving) are indicated by notes, which are already available in the UML standard [7].

The first notation in the Fig. 1 use notes (in grey) to indicate where each activity is taking place, e.g., in the office, in the car while driving or while parked, or in the patient's home. The second notation in Fig.1 instead uses colour, where office activities have grey notes, activities in the car while driving have blue nodes, while parked red, and in the patient's home yellow. It is easy to observe some pros and cons for each notation. The annotation alternative is a much lesser deviation from standard UML than the colour alternative. The annotations could also have the advantage that the location/context is always quickly found just beside the activity node, while the colour notation requires the user either to remember or consult the colour legend to understand the diagram details. On the other hand, the annotation alternative causes the diagram to have more nodes and links, thus becoming more complex. Another advantage of the colour notation, which is not directly visible in figure 1, is if one wants to compare several design alternatives putting them side by side (e.g., one where getting info about the patient is done while driving, another where it is done while parked or in the patient's house). Having two such alternatives side by side with the colour notation, it would be immediately spottable where the difference between the two process designs are (because two nodes would be differently coloured while the others are identical), while it would not be so obvious with the annotations, requiring detailed inspection of the text inside the notes. Parts of the outcome of the analytical evaluation in [5] are summarized in Table 1 and 2 below where the alternative notations are compared against traditional UML activity diagrams (but here only including the two above mentioned notations, not some others that were less favourably evaluated). All in all, the analytical evaluation might indicate some advantage for the colour notation. However, as the discussion above reveals this advantage is not entirely clear, and it is impossible to know whether the list of evaluation criteria was complete, and whether all criteria should have equal weight or maybe some are more important than others in determining the practical usefulness of the notations. Hence, such analytical evaluations must be supplemented by empirical evaluation, which is the purpose of the research presented in this paper. A controlled experiment where participants use both notations for the same tasks could be able to reveal whether any notation has advantages when it comes to supporting understanding and problem solving related to the modelled case. Hence such an experiment was performed, and the purpose of the current paper is to report on that experiment. For such experimental study on comparing information modelling methods [19], it is important to show that the two notation alternatives and the cases are informationally equivalent. In other words, we do not want one language to be more comprehensive than the other. Since both notations are extensions of the same modelling language, adding the possibility to add location information in two different ways, we can argue for such equivalence.

Table 1. Evaluation of proposed notations with simple and large models

Notation	Min. deviation from standard		Expressiveness		Intuitive / Easy to read		ModelComplexity	
	Simple	Large	Simple	Large	Simple	Large	Simple	Large
Trad. UML	+ +	+ +	- -	- -	+ +	+ +	+ +	+ +
Annotated	+	+	+	+	-	- -	+	- -
Colours	-	-	+ +	+ +	+	+	+ +	+ +

Table 2. Evaluation of proposed notations with SEQUAL framework

Notation	SEQUAL Framework [8] on Language Quality		
	Organizational Appropriateness	Domain Appropriateness	Comprehensibility Appropriateness
Traditional UML Act. Diag.	+	+	++
Annotated	+	+	++
Colours	++	++	++

Walderhaug et al. used UML notations extensively in the MPOWER project [4] with homecare services and conclude that UML profiles [10] can be used as a mechanism for tool chains based on OMG's Model Driven Architecture (MDA) and UML standards [2] [7]. Work on mobile ontologies by Veijalainen [9] supports the idea of the 'where' aspect as essential in mobile processes, but excludes the 'what' aspect. Larsson [10] proposes the three building blocks for knowing the processes list How, What and Why, adds Who for use oriented design approach but omitted the 'Where' concept. Whereas the use of colour is not so common in conceptual modelling, in other 'modelling' areas such as cartography colour is widely used to improve comprehension. Related to comprehension is the work by Bertin [11] on visual variables, where colour is one important differentiator. Moody [12], in his proposal for a 'physics' of notation based his work on among other things the work of Bertin.

3 Research Method

Our goal in this paper is to compare the two notations presented above to find out which one gives the best results for the user, in terms of understanding process diagrams or using such diagrams in problem solving activities. Of course, there are a number of ways to make such comparisons, ranging from analytical comparisons, through controlled experiments on small-scale tasks, to usage in large-scale industrial projects. Since our previous work made an analytical comparison, a natural next step is to go empirical, i.e. either experiment or industrial case study. Moreover, since the goal here is to compare two notations rather than investigate the merits of one notation, it makes most sense to go for a controlled experiment. With industrial case studies, it would be impossible to do exactly the same project twice; hence any comparison would easily be blurred by a number of confounding factors. With the decision to go for a controlled experiment, the next question is to decide the details about the experimental treatment and the tasks that the participants will perform. The normal "treatment" [13] when experimenting about new diagram notations is to give the participants a quick tutorial about the notation they are going to use, and then supply them with a case and some tasks to solve. For generalizability of the results, a wide variety of tasks might be a good idea, but on the other hand controlled experiments normally have to be constrained to a quite limited time frame - it is hard to recruit enough subjects if it lasts longer than a couple of hours. Hence, we chose to go for two experimental tasks: (i) testing the participants' understanding of the case, measured by their score in answering a number of true/false questions, and (ii) the participants' problem solving abilities, measured by their ability to detect errors in a diagram - this investigated by providing a textual case description declared to be correct and they a diagram assumed to capture this, but deliberately seeded with some errors. In addition we elicited the participant's opinion about the notation through a post-task questionnaire. We thus had three main variables to measure about each notation in the experiment:

Understanding: the participant's fraction of correct answers to the true/false questions.
Error_detection: the participant's fraction of correctly detected errors to the total number of errors in the deficient diagram.
Average_opinion: the participant's average score on 14 questionnaire items about the notation.

Another question is whether to do a within-subjects (i.e., all subjects get both treatments) or between-subjects (half the subjects get one treatment, the other half the other treatment) experimental design. According to [13] a within-subjects design is advantageous if it is possible with the given treatments, since it doubles the sample size and also controls better for selection bias. On the other hand, this design has some added challenges with learning effect, meaning that two different cases must be used and participants divided in four groups in a so-called Latin Squares design, as will be explained below.

Given the abovementioned variables, and since the colour notation came out slightly better than the annotated notation in the analytical comparison in [5], we have the following key hypotheses for our experiment:

H1: the Understanding scores for the colour notation will be better than those for the annotated notation
H2: the Error_detection scores for the colour notation will be better than those for the annotated notation
H3: the Average_opinion scores for the colour notation will be better than those for the annotated notation

Corresponding null hypotheses could also be formulated, but are not presented here for space concerns. Also, there could be more detailed hypotheses relating to different question groups investigated in the post-task survey, i.e., 5 questions related to Perceived Ease of Use, 5 to Perceived Usefulness, and 4 to Intention to Use. These hypotheses would be similar to H3, i.e., assuming that the colour notation would score better than the annotated notation. Again, these more detailed hypotheses are not shown to save space and found to be less significant.

46 students were recruited from a second year computer science class to take part in the experiment. With the Latin Squares design, these were divided into 4 groups according to which annotation to try first, and on which case the annotation was used, creating 4 different permutations, e.g., annotated first with the home care case, then colour with another case about traffic control, annotated traffic, then colour with home care, colour with home care then annotated traffic, or colour with traffic then annotated home care.

1. The 46 participants were randomly distributed to the four experiment groups in the Latin Squares design. The questionnaire prepared contained four parts, 1. Pre-experiment questionnaire 2. Questionnaire on CN or AN with case 1 and post experiment evaluation. 3. Questionnaire on AN or CN with case 2 and post-experiment evaluation. 4. Identifying mistakes in CN and AN with brief case explanation. The participants performed the following activities during the experiment: Answering a pre-experiment questionnaire: The purpose of the pre-experiment questionnaire was simply to investigate the participants' prior knowledge of related topics like UML, process modelling, etc., which can be used to control for any accidental group selection bias in spite of random selection (e.g., one group accidentally containing people with much more relevant experience). This is much less important for a Latin Squares design than for a between-subjects design, but since it only takes a couple of minutes for the participants to answer a few questions about their prior experience with relevant modelling techniques, it still felt worthwhile to do. Questions investigated previous knowledge on modelling, UML, activity diagrams, specifications, IT work experience and knowledge about the domains the cases were taken from (home care and traffic control), in total 8 questions that were to be answered within 5 minutes.
2. Reading a tutorial about the first diagram notation (annotated or colour), using a flight check-in case as an example case description followed by corresponding annotated or colour notation diagram were presented as a tutorial part in the experiment.
3. Being presented with experimental textual case description (home care or traffic control), together with a diagram (annotated or colour), and at the end of this, participants must answer 12 true/false questions related to that particular case.
4. Answering a post-task questionnaire about the notation just used, containing 14 questions investigating Perceived Ease of Use (PEOU), Perceived Usefulness (PU), and Intention to Use (ITU) as inspired by the TAM model [14].
5. Repeating steps 2-4 using the other notation on a different case. Totally 54 minutes were allotted to complete steps 2-5 and return the booklet.

6. A separate booklet with textual description and notations on both cases deliberately seeded with some errors was distributed again and now the task was to find all the errors in the diagram (i.e., discrepancies between the diagram and the case description) i.e., all the students had questionnaire on both notations and both cases in steps 2-5 and also in 6. Of course, the seeded errors (5 errors per notation) were the same both for the annotated and coloured variants of the diagrams. The allotted time to complete these tasks was 10 minutes. The experiment performed in Latin squares design and the cases distribution is as shown in table 3.

Table 3. Latin square experiment questionnaire distribution to student groups

Group Id	**(Understanding+ TAM factor) Questionnaire on**	**Error Identification Questionnaire on**
Group A	Annotated Home Care + Colour Traffic Control	Annotated traffic Control + Colour Home Care
Group B	Colour Traffic Control + Annotated Home Care	Colour Home Care + Annotated traffic Control
Group C	Annotated traffic Control + Colour Home Care	Annotated Home Care + Colour Traffic Control
Group D	Colour Home Care + Annotated traffic Control	Colour Traffic Control + Annotated Home Care

4 Experiment Results

The results for the performance of the participants on the tasks of understanding (answering 12 true/false questions) and problem solving (detecting errors in diagrams relative to a natural language case description) are summarized in table 4. As can be seen, both variables turned out with a significant advantage in favour of the coloured notation, with effect sizes of 0.51 for understanding and 0.45 for error detection, which are both small to moderate effects according to [16]. For the error detection task, there were however 3 students who performed extremely poorly compared to the others, one found zero errors with both notations, one found zero errors with colours and only 1 with annotations, and another found 1 with annotations and 2 with colours. Assuming that these participants might not have done their best with the task, they might be treated as outliers and excluded from the analysis, but the result (N=43) still gives a significant advantage in favour of colours, now with an effect size of 0,77 which is a moderate effect.

The results for the participants opinions about the two notations, as indicated by their answers to the TAM-inspired post-task questionnaire, is shown in Table 5. As indicated there was a slight advantage for the coloured notation for PEOU and ITU, but not for PU, and no differences were significant, neither for the overall average nor for the single TAM factors.

Table 4. Comparison of performances with the two notations

Compared variable (N=46)	**Coloured diagram**		**Annotated diagram**		**Difference**	**Effect Size**	**Sign.? Y/N (p-value)**
	Mean	**SD**	**Mean**	**SD**			
Understanding	0.960	0.057	0.926	0.078	0.0344	0.51	Y (0.01)
Error detection	4.50	1.34	3.93	1.17	0.57	0.45	Y (0.001)
Errors (w/o outliers, N=43)	4.77	0.53	4.16	1.04	0.60	0.77	Y (0.001)

Table 5. Comparison of TAM factors with the two notations

Compared variable (N=46)	**Coloured diagram**		**Annotated diagram**		**Difference**	**Significant? Y/N (p-value)**
	Mean	**SD**	**Mean**	**SD**		
PEOU	3.97	0.63	3.86	0.51	0.11	N (0.2)
PU	3.95	0.79	3.96	0.67	0.01	N (0.12)
ITU	3.49	0.89	3.26	0.83	0.23	N (0.27)
Average_ opinion	3.83	0.67	3.73	0.54	0.1	N (0.16)

All in all, then, we have the following situation for our hypotheses:

- H1 was confirmed; there was indeed a significant advantage for the coloured notation when it came to the measured understanding in terms of the scores for the true/false questions.
- H2 was also confirmed; there was a significant advantage for the coloured notation when it came to measured problem solving capability in terms of the number of identified errors in the diagrams.
- H3 must be rejected, as no significant advantage could be found for any notation when it came to responses to the post-task questionnaire investigating the participants' opinions about the notations and these data are documented in appendix.

Normally, one might think that the modelling technique that gives the best performance would also get the best evaluation from the participants when asking about their opinion. However, the situation that performance advantages has not necessarily yielded a similar advantage in opinion has also been observed in other experiments, for instance in [17] where one technique had a significant performance advantage, yet no such advantage was found for opinion, indeed there was no notable correlation between the participants' performance with a technique and opinion about that technique.

5 Threats to Validity

Wohlin [18] suggests four relevant categories for discussing threats to validity in experiments: conclusion validity, construct validity, internal validity and external validity. *Conclusion validity* concerns the relationship between the treatment given and the outcome in measured variables. One important question is whether the sample size is big enough to justify the conclusions drawn, which can be investigated by means of the calculated effect size (ES). We accepted two hypotheses, one about better understanding with the coloured notation (ES=0,51) and one about better error detection (ES=0,45, or with outliers removed, ES=0,77). Denoting the Type I error probability by α (accepting a relationship which really is not there) and the Type II error probability by β (overlooking a relationship that really was there), the following holds:

$$N = \frac{4(u_{\alpha/2} + u_{\beta})^2}{ES^2} .$$

If we use $\alpha = 0.05$ (our threshold for accepting a relationship as significant) and $\beta = 0.20$, we get $N = 32/(ES)^2$ [16] as a required sample size. This means that we should have had a sample size of 123 for understanding and 158 for error detection (or 54 for the ES achieved when outliers were removed). Our sample size was only 46 (or 43 with outliers removed). The fact that our sample size is smaller than the required ones means that our results have to be interpreted with caution, although the difference clearly came out as significant.

Construct validity is concerned with the inference from the measures made in the experiment to the theoretical constructs we were trying to observe (understanding, problem solving effectiveness). Of course, there are other ways to explore understanding than true/false questions after looking at a case description and diagram, and other ways to explore effectiveness than asking participants to identify errors. But at least, identification of errors is an important task in system development (for instance in connection with reviews / QA), and answering questions is at least one relevant way of testing understanding. Given the limited type (only true/false) and nature of the questions, it must still be admitted that they will not measure every aspect of understanding, and that more experiments with a wider range of experimental tasks would be necessary to draw more certain conclusions.

Internal validity means that the observed outcomes were due to the treatment, not to other factors. Our Latin-Squares experimental design was used to eliminate selection bias, and to control for any learning effects or effects of which case was used with which technique. In addition we performed a pre-experiment questionnaire to test whether other factors such as previous relevant experience could explain the differences between the various groups, but not finding any such effects.

External validity is concerned with the question of whether it is possible to generalize from the experimental setting to other situations, most importantly to industrial systems development. The use of students instead of practitioners is a notable threat. However, this threat is reduced by the fact that we are only trying to compare two notations in relative terms, not evaluate their merits in more absolute terms. Moreover, these adapted notations would be new also to practitioners, thus reducing the advantage they might otherwise have had over students (e.g., if the practitioners were familiar with the notations from before and had used them a lot at work).

6 Conclusion

For the development of mobile and multi-channel information systems it might be important to have process notations able to capture the "where" aspect, i.e., the location or context of the various activities. For instance it might have a substantial impact on usability requirements if an activity is supposed to be performed in office with a desktop PC or using some mobile device while driving a car. Hence, we have suggested some alternative adaptations of the UML Activity Diagram notation to enable the capturing of location and context. In this paper, we have reported on a controlled experiment comparing two such adapted notations, one using annotation and the other using colour to capture the location/context of activities. In a previous analytical evaluation, the colour alternative came out as slightly better, hence we hypothesized an advantage for the colour notation for all the variables we measured, i.e., that the colour notation would prove better both for understanding, error detection and participant opinion about the notations. Statistical analyses confirmed an advantage for understanding and error detection, but not for the opinion.

The effect sizes, however, were not large enough to support conclusion validity. A natural first step for further work would therefore be to perform the same or similar experiments with more subjects. It might also be interesting to modify the experimental tasks. The case descriptions and corresponding activity diagrams were fairly small and simple, and more complex cases would make future experiments more representative of realistic industrial enterprise modelling task. More complex tasks could also bring the advantage of larger effect sizes - *if* indeed one notation has an advantage over the other. In particular, the understanding task with true and false questions seems to have been fairly easy, the average score being above 90% with both notations, which means that any difference will necessarily be small.

Other venues for future work could be to include yet other notations in similar experiments. While colour came out as the best in this experiment, it does have problems w.r.t. colour-blind users. Hence, it could be interesting to see if a notation using pattern fills (also mentioned as a possible visual variable by Bertin) in the activity nodes could do just as well as colour.

Finally, of course, experiments should be supplemented with larger case studies in enterprise modelling / systems analysis projects, preferably in industry. This would provide a better indication whether any advantages observed in a limited experimental setting also hold for real world usage.

References

[1] Unified Modelling Language, `http://www.uml.org` (accessed 4.6.2010)
[2] Business Process Modelling Notation, `http://www.bpmn.org/` (accessed 4.6.2010)

[3] Korherr, B., List, B.: Extending the UML 2 Activity Diagram with Business Process Goals and Performance Measures and the Mapping to BPEL. In: Roddick, J., Benjamins, V.R., Si-said Cherfi, S., Chiang, R., Claramunt, C., Elmasri, R.A., Grandi, F., Han, H., Hepp, M., Lytras, M.D., Mišić, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (eds.) ER Workshops 2006. LNCS, vol. 4231, pp. 7–18. Springer, Heidelberg (2006)

[4] Walderhaug, S., Stav, E., Marius Mikalsen, M.: Experiences from Model-Driven Development of Homecare Services: UML Profiles and Domain Models. In: Chaudron, M.R.V. (ed.) Models in Software Engineering. LNCS, vol. 5421, pp. 199–212. Springer, Heidelberg (2009)

[5] Gopalakrishnan, S., Sindre, G.: Alternative Process Notations for Mobile Information Systems. In: Proc. I-ESA 2010, Coventry, UK, Springer, Heidelberg (2010)

[6] Andresen, S., Krogstie, J., Jelle, T.: Lab and Research Activities in Wireless Trondheim. In: Proceedings of IEEE International Symposium on Wireless Communication Systems, pp. 385–389. IEEE Computer Society, Los Alamitos (2007)

[7] Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language: User Guide. Addison-Wesley, Reading (1999)

[8] Lillehagen, F., Krogstie, J.: Active Knowledge Modelling of Enterprises. Springer, Heidelberg (2008)

[9] Veijalainen, J.: Developing Mobile Ontologies; who, why, where, and how? In: International Conference on Mobile Data Management, Manheim, Germany, pp. 398–401. IEEE, Los Alamitos (2007)

[10] Larsson, A. V.: Designing for use in a future Context – Five Case Studies in Retrospect, PhD thesis No: 1034, Institute of Tech., Linkoping Univ., Sweden (2003)

[11] Bertin, J.: Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press (1983)

[12] Moody, D.L.: The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transanctions on Software Eng. 35(6), 776–779 (2009)

[13] Field, A., Hole, G.: How to Design and Report Experiments. Sage Publications, London (2003)

[14] Davis, F.D.: Perceived usefulness, perceived ease of use and user acceptance of information technology. MIS Quarterly 13, 319–340 (1989)

[15] Zachman, J.A.: A framework for information systems architecture. IBM Systems Journal 26(3), 276–291 (1987)

[16] Hopkins, W.G.: A New View of Statistics. University of Queensland, Australia, Brisbane (2001)

[17] Opdahl, A.L., Sindre, G.: Experimental comparison of attack trees and misuse cases for security threat identification. Information and Software Technology 51(5), 916–932 (2009)

[18] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic, Norwell (2000)

[19] Siau, K.: Informational and Computational Equivalence in Comparing Information Modeling Methods. Journal of Database Management 15(1), 73–86 (2004)

Appendix

In this section, some of the additional results, questionnaire information and detailed description of the other case (police traffic control) of this student experiment are documented. This below section will give the reader a bird's eye view on how the experiment has been performed too.

A1 Police Traffic Control Case

The UML diagram notations presented in this study is based on a simple task in the complete police traffic control case. In this study we used alternate notation – Annotated Notation (AN) in view with mobile information systems. Before getting into the case, it may be useful to know the background information about the case, how the traffic police utilising mobile devices for their work. The leader of the police allots control locations for each police assistant personnel for controlling the vehicles through those locations. Controlling includes following up things such as driver license, speeding, drunken driver...etc. Police Assistant Personnel (PAP) receives info

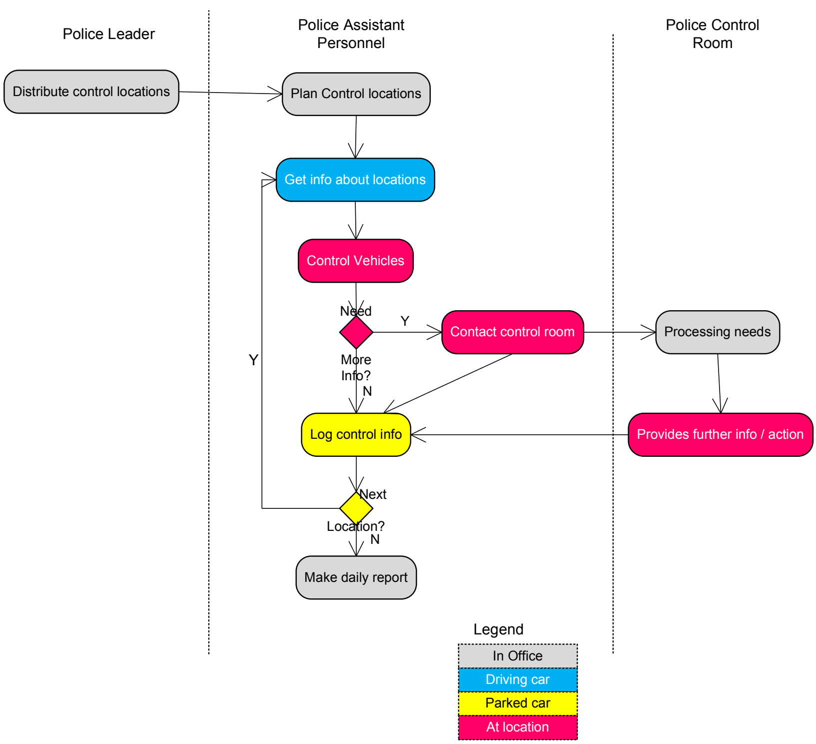

Fig. 2. Police traffic control case- Color Notation (CN)

about control locations at the office from leader. PAP plans and gets info about the control locations while driving the car. After reaching the control location, PAP controlls the drivers and vehicles. If he decides he need more info/personnel he contacts through mobile/radio/hand held devices...etc the Police control room. The control room provides necessary info/further actions to PAP at control locations. PAP completes the scheduled task for scheduled hour at particular locations and logs all the info from a parked car. He repeats the task until PAP finishes all control locations for the day. After completion, PAP returns to the office and make a daily report. Fig. 2 is based on Color Notation (CN) and Fig. 3 Annotated Notation (AN) of UML activity diagram reflecting the police traffic control case.

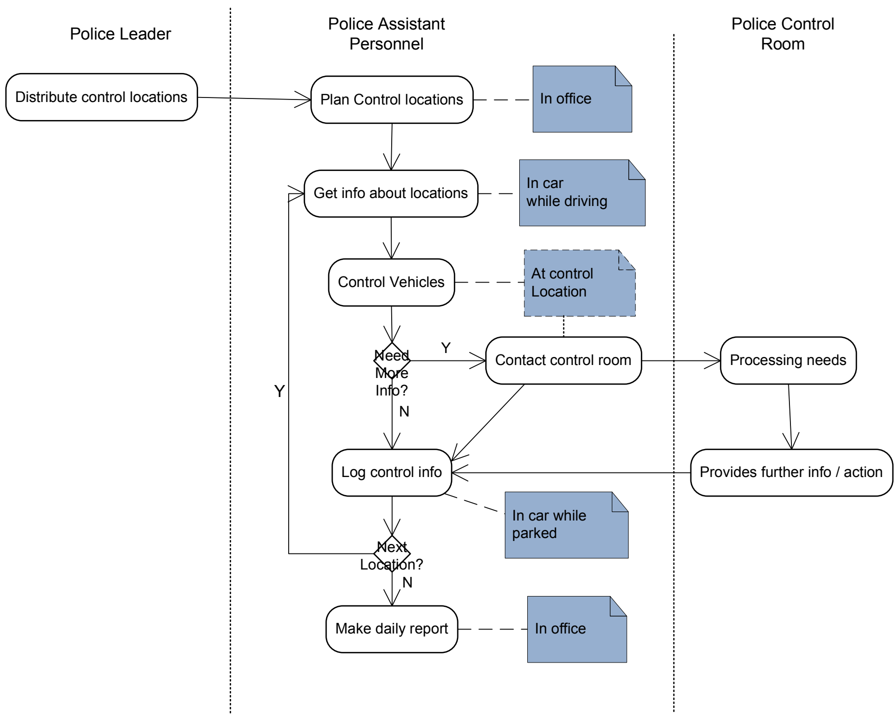

Fig. 3. Police traffic control case –Annotated Notation (AN)

A2 Questions about Home Care Case with Annotated Notation

(Please write down whether statements are True/False)

1. Home Care Assistant (HCA) receives the patient list to be visited at the office.
2. HCA has been given the patient list along with visit sequence.
3. HCA contacts the nurse while driving the car usually.

4. If HCA needs medical experts suggestion HCA returns to office to meet Leader.
5. All the patients visit in the list visits HCA at the hospital.
6. HCA collects all preparatory info about patients at the office itself.
7. Patients visited in sequence as per patient's desire.
8. HCA visits all the patients in the list at the patients place.
9. HCA gets the nurse/experts suggestion while driving car.
10. Nurse comes down to patient's location and provide further info if it is needed by HCA.
11. HCA returns to office after visiting one patient in the list to log info about visit.
12. HCA starts and completes his/her duty at office.

A3 Questions about Police Traffic Control Case with Color Notation

(Please write down whether statements are True/False)

1. Police Leader (PL) visits all control locations along with Police Assistant Personnel (PAP).
2. PAP starts his duty from office everyday.
3. Always PAP gets the control locations distributed by police Leader (PL).
4. After controlling vehicles at each location PAP return to office to log info.
5. If PAP requires more info during controls from control location, he/she contacts police control room.
6. PAP usually prepares daily report at control locations.
7. PAP logs info after each control location visit in parked car.
8. Daily report was made each day by PAP at office.
9. PAP must contact control room before each control location visit.
10. Police Leader prepares daily report for PAP at office.
11. PAP contact Police Leader if PAP needs more info from control locations.
12. PAP returns to office only after completing all scheduled control locations visit and makes daily report at office.

A4 Pre-experiment Analysis

Based on the results of questionnaires A2 and A3, it is found that colour specification marginally influence on understanding the context of the cases. The below Fig. 4 shows reflects the same.

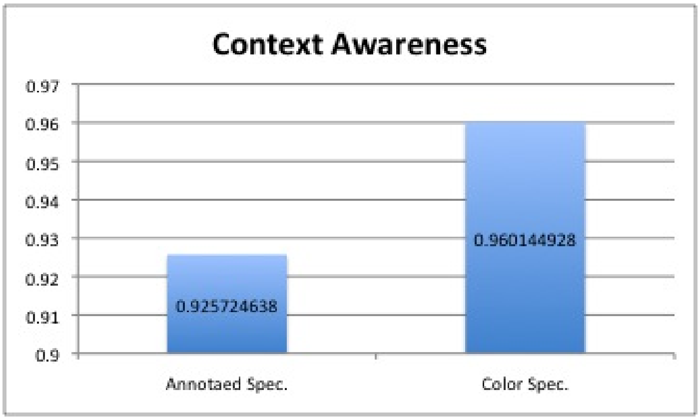

Fig. 4. Understanding (Context Awareness) of case definition through Notations

A5 Post Experiment Questionnaire and Results

The TAM model was used with the three factors Perceived ease of use (PEOU), Perceived usefulness (PU) and Intention to use (IU). As a part of this evaluation fourteen questions put for the students to answer based on their experience on using these two notations. On evaluating these experiences it is found that the colour notations outperformed insignificantly the annotated notation in all three PEOU, PU and IU aspects. Post-experiment questionnaire with 14 questions with results are Fig. 5 and in table 6.

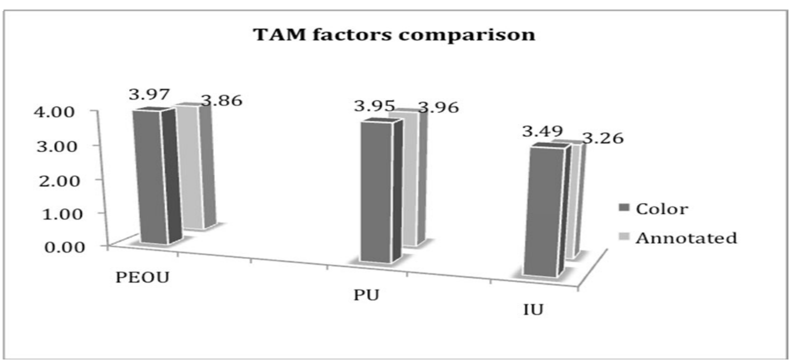

Fig. 5. Post-experiment TAM Questions analysis

Table 6. Result with TAM questionnaire on both notations

Question id.	TAM Questionnaire	Related TAM factor	Annotated Notation (max.avg. value :5)	Colour Notation (max.avg. value :5)
1	Notation gave me a better understanding of the activity where it is performed	PEOU	4.435	4.652
2	It would be easy to get used to the Notation in a project.	PU	4.217	4.239
3	I found this Notation is very easy to master.	PEOU	4.000	4.087
4	If I need to identify where the activity process done in a future project, I would use this Notation	IU	3.130	3.435
5	I would have found where the activities has been performed by using common sense	PEOU	3.0434	3.174
6	I found not hard to use and recognize this notation	PEOU	3.978	4.109
7	I was not often confused about how to apply this Notation to the problem (activity diagrams)	PEOU	3.848	3.848
8	This Notation made the activity diagrams more systematic.	PU	3.848	3.978
9	If I am employed in a company in future discusses what technique to introduce for where the activities performed argue and someone suggest this Notation, I would support that	IU	3.348	3.696
10	I can read this Notation quickly, understand the notation diagram	PU	3.957	3.870
11	This Notation will be easy to remember	PU	3.891	3.891
12	This Notation made me more productive in finding where the activities been performed.	PU	3.891	3.761
13	I will try this notation if I been assigned in my future work involving mobile process	IU	3.239	3.478
14	If I am working as a freelance consultant for a customer who needs help finding where the activities(mobile processes) performed to his project, I would use AN in discussions with that customer.	IU	3.326	3.348

A Repository Architecture for Business Process Characterizing Models

Shang Gao and John Krogstie

Department of Computer and Information Science (IDI),
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
shanggao@idi.ntnu.no, krogstie@idi.ntnu.no

Abstract. In this paper, the Business Process Characterizing Model (BPCM) repository architecture is presented. The repository architecture follows a three-layer model composed of a presentation layer, a repository management layer and a storage layer. The objective of the architecture proposed here is to organize BPCM models in a manner that enable their reutilization, and some guidelines on how to use the BPCM repository are provided. Business users are expected to be able to benefit from the advanced functionalities provided by the BPCM repository. Furthermore, based on the evaluation of the BPCM repository by comparing it with some other process repositories, we find that the BPCM repository can offer most features other process repositories are able to provide with the exception of control flow.

Keywords: Business Process Characterizing Model, the BPCM Repository, XML.

1 Introduction

Today, since organizations are required to react to ever changing market needs in a timely agile manner to stay competitive, an increased flexibility in the business processes is highly desired. Business Process Management (BPM) is the approach to manage the execution of IT-supported business processes from a business stakeholder's point of view rather than from a technical perspective [23].

Information systems development often starts with the development of process models. Model developers usually create process models in a graphical process modeling tool. Subsequently, the process models can be implemented by developers. The process models are supposed to reflect business stakeholders' views on the business domain. However, the proposed processes often deviate from business stakeholders' expectations on the desired processes. One of the main reasons is that some concerns from business stakeholders' views could not be appropriately placed in the process models. And changes which might occur to some tasks in the process models are also not easy to track as the model is transformed among different levels in business systems development according to the needs from various users. In addition, some industry projects and case studies [13], reveal that graphical process

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 162–176, 2010.

models are not a good starting point for identifying stakeholder requirements. Many business stakeholders want to start a project with the development of a business oriented model addressing essential business aspects (i.e., what are the essential requirements of the project, what is offered by whom to whom, etc), rather than look at a relative complicated business process model showing how things are executed operationally.

It is believed that modeling the processes of a business domain is a time-consuming and challenging task. Therefore, it is not surprising that some additional models and frameworks are proposed to capture processes at an abstract level. Also additional frameworks and notations are needed to deal with ad-hoc changes and bridge the gap between business stakeholders and model developers. We have proposed a Business Process Characterizing Modeling language (BPCM) [10] to create a link between business needs and IT implementations. In another paper [27], the authors addressed the problem in the following way: given a high-level process model and a set of previously modeled artifacts in a repository, all semantically annotated, a composition approach should come up with the required set of artifacts from the repository, orchestrating the artifacts in a way that reflects the business process model's structure and business semantics. Compared to some other recent work addressing similar issues, the notable difference in our BPCM work is that we are creating a BPCM language to help business stakeholders create a high level representation of process knowledge in a tabular format in order to share knowledge with other business stakeholders and facilitate the design work of model developers.

During the course of business process support system development, many models in different modeling languages are defined and created. For example, model developers use process models to model business processes. Business stakeholders use some basic modeling tools to express their expectations of the process support system. It is common to have a repository to store all relevant models.

In this paper, we will present a BPCM repository for storing and querying BPCM models. We will also describe the functions that the BPCM repository is able to provide. Thus, the main contributions of this work are the proposed BPCM repository architecture and an evaluation of the BPCM repository by comparing it with other existing process repositories.

An organization can use the BPCM repository for advanced search functionalities. The collection of BPCM models to be analyzed and reused can be retrieved from the BPCM repository. Business users are supposed to profit from the advanced functionalities provided by the repository. For instance, they can search for relevant BPCM models from specific business domains in terms of keywords. Alternatively, they can also perform their searches by using the name of the processes in which they are interested.

The remainder of this paper is organized as follows. Section 2 states the motivation and some related concepts of this research work. In Section 3, we briefly describe the BPCM modeling language, and the BPCM meta-model. Section 4 presents the BPCM repository architecture. Some guidelines on how to use the BPCM repository are provided in Section 5. Section 6 compares the BPCM repository with other process repositories. In Section 7, the contributions of this research are summarized. Finally, Section 8 concludes this work and points out some further research directions.

2 Motivation and Related Concepts

A BPM lifecycle often contains several stages (e.g. Planning, Design, Integration, etc). It is essential to create a fundamental process model in the early phase of a BPM lifecycle, which is a key to the success of business process management of an organization. However, business process modeling is a complicated process. It is time-consuming and error-prone. Therefore, it is important to provide a common repository for business stakeholders to find existing business processes and simplify their effort in the process of business process modeling. We believe that creating a BPCM repository can help them achieve this goal.

Building a good process model is not trivial. It requires a comprehensive understanding of the business domains of the organization, while keeping the business goals, and best practices from other organizations in the same business domains in mind. The constructs of our BPCM model can cover the basic requirements of constructing a business process. Therefore, building a BPCM repository can help business users reuse existing BPCM models for various purposes.

According to Bernstein and Dayal [4], a repository is “a shared database of information about engineered artifacts produced or used by an enterprise”. It should provide a common database system for model creation, model modification, model retrieval, model version management. Examples of such artifacts include documents, models, information systems, etc. In our case, the used artifacts in the repository includes: BPCM models on a presentation level, and the BPCM meta-model on a data level. A meta-model is the data that describes meta-data. The concept of meta-model has been successfully used in data interoperability. In [29], the authors proposed the use of meta-models as an advanced architecture for information mediation, where XML is a main driving force because of its enablement of interoperability. This meta-model based mediation improves the flexibility of choosing common data schema and query language and provides a high level of interoperability. Furthermore, the functionalities for a general repository, as summarized in [4], inspired us to develop repositories that are specific for storing and managing BPCM models.

In general, storing models in a common repository has several benefits. Firstly, it is easy for other users to reuse the relevant models in the future. Second, the models in the repository are subject to common control services. Different access rights can be given to different groups of users by setting up access controls of the repository. Third, it provides a platform for various users to share information so that they can benefit from each other.

Concerning the BPCM model, one of the main benefits of charactering business processes in general is the enablement of searching business oriented process information, such as finding processes in a similar domain or at the same or adjacent step in the SCOR-model (see next section), which can for example later be used for constructing better business process models.

With regard to BPCM models, we use a systematic way to describe BPCM models in order to reuse them. A document file which is specified in XML is defined to describe the characteristics of business processes by following BPCM modeling approach. In an effort to make readers’ have an impression on the data level of a BPCM model, a screenshot of a part of a sample BPCM Model in XML is shown in Fig. 1. XML is a marking meta-language that allows the structured representation of several types of information. This information can be easily stored and processed on the web. We have seen more and more successful repositories enabled by XML. In our case, XML, which is considered a solution for interoperability, also provides an opportunity to transmit data between BPCM models and other process models. Since most process models can be stored in XML format. For example, BPMN [28] models can be stored in XPDL [8], EPC [2] models can be stored in EPML [20], BPEL [14] can be stored in the BPEL XML format. As a result, this will ease the interaction between BPCM models and other process models. In addition, the repository can also store information about the revision history of various models to better manage them.

```
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://www.owl-ontologies.com/Ontology1269468601.owl">
<owl:Class rdf:ID="Soft_Goal">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Goal"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Soft Goal</rdfs:label>
 </owl:Class>
 <owl:Class rdf:ID="Hard_Goal">
```

Fig. 1. A part of a sample BPCM Model in XML

3 Business Process Characterizing Model (BPCM)

In the course of business process support systems development, model developers often focus on operational and procedural aspects of business process systems, while various business stakeholders are more likely to express different concerns with regard to process models in terms of business oriented concepts. The business process characterizing model (BPCM) is a model used to represent a high level knowledge of business processes.

In an effort to facilitate readers' understanding of the BPCM modeling, we summarize the general definition of the elements for business process characterizing modeling (see Table 1) and present the BPCM metamodel (see Fig. 2) in this section. Since some elements in the BPCM model refer to other ontologies or concepts as presented in the Table 1, a brief description of those concepts and the motivation to incorporte those concepts is provided as follows.

Concerning the element context, there is no universal or absolute definition for context. [6] describes context as "typically the location, identity, and state of people, groups and computational and physical objects". Context is the reification of the environment, that is, whatever provides a surrounding in which the system operates. People can base their own perceptions or understanding to define context in different ways. In order to better design business process support systems, it is crucial to understand the working contxet and collect and deliver contextual information in a better way. By including a context element in a BPCM model, the correspondent business process support system can be made to serve people better in both mobile and more stationary computing settings.

Table 1. Definition of the elements in BPCM

Element	General Definition
Process	The business process people want to characterize. This element can be related to a common business process ontology such as SCOR [5].
Resource	This element is inspired by the resource concept in the REA framework [19]. This element can clearly address what are consumed and what are gained in a business process.
Actor	This element describes the people and organizations with different roles involve in a business process. This element can illustrate who are important to which business process.
Context	It includes contextual characteristics in terms of devices, software on the devices and networks providing connections between the devices and others.
Business Domain	This element classifies the related business domain(s). We link this element to the North American Industry Classification System (NAICS).
Goal	This element can address what goals need to be fulfilled in the business process. The goals may be related to operational goals and strategical goals. Operational goals are related to hard-goals, usually covering functional requirements; while strategic goals are related to soft-goals, which set the basis for non-functional requirements.
Process Type	According to REA [19], REA does not model only exchanges but also conversions. Exchange and conversion can be seen as two typical process types.
Version	This element is designed for keeping track revisions of BPCM models.

Recently, W3C has released a draft version of the Delivery Context Ontology (DCO). This ontology constructed in OWL provides a model of characteristics of the environment in which a device interacts with the web or other services. In this research work, we incorporate some key entities of DCO into the context element of BPCM. Some other research work has also started addressing the relationship between context and system development at the requirement level. For instance, [3] investigates the relation between context and requirements at the beginning of goal oriented analysis, and [21] extends the application of the problem frames approach with context monitoring and switching problems.

REA [19] [12] was originally conceived as a framework for accounting information systems, but it has subsequently broadened its scope and developed into an enterprise domain ontology [15] and e-commerce framework [1]. The core concepts in the REA ontology are *resource*, *event* and *agent*. The intuition behind this approach is that every business transaction can be described as an event where two agents exchange resources. In order to acquire a resource from other agents, an agent has to give up some of its own resource. The duality of resource transfer is essential in commerce. It never happens that one agent simply gives away a resource to another without expecting another resource back as compensation. Basically, there are two types of events: *exchange* and *conversion* [15, 19]. An exchange occurs when an agent receives economic resources from another agent and gives resource back to that agent. A conversion occurs when an agent consumes resources to produce other resources. Annotating process with process type enable users to identify the attribute of business processes around the resource lifecycle. As illustrated in [9], the element resource of BPCM is important in identifying relevant tasks or activities for the construction of process models. For each resource in a BPCM, it should include a message flow which links two associated tasks in a BPMN process model, whereby the source of the message flow connected to the dependee's task and the destination of the message flow connected to the depender's task.

Last but not least, the North American Industry Classification System (NAICS) is a standard for the collection, tabulation, presentation, and analysis of statistical data describing the U.S. economy. NAICS is based on a production-oriented concept, meaning that it groups establishments into industries according to similarity in the processes used to produce goods or services. Each business process is labeled with a business domain. This is of help for model users to search or retrieve business processes within specific business domain.

The BPCM meta-model [11] is described in Fig. 2, which presents the major modeling concepts and the relationships between them using a UML class diagram.

The central class of the meta-model is *process*. One process may consists of zero or more sub process(es). All other main classess can be associated with the central class process. Around the central class, business stakeholders could express different concerns, like required actors, required resources, contextual information, and so on. We use UML generalizations in case of elements extensions (e.g. context, process types). Then, process model developers can take these concerns into consideration when designing process models accordingly.

The delivery context ontology mainly focuses on the following three entities: a). environment including information about the location and network, b). software describes whether the delivery context supports certain APIs, document formats, operating systems, etc, c). Hardware provides information about various hardware capabilities including display, input, memory, etc. For the class *context*, we do not attempt to cover all entities above. We focus on two major aspects: location and network information. For example, a user using its device is connected to an information system through a wireless network or wired network. Mobile workers need to work in various locations (e.g. at home, in the office, on the way). Mobility has become an important trend for various activities in both working and non-working settings. Several taxonomies of remote mobility are discussed in the literature. In [16], the authors distinguish between travelling, visiting and wandering. Travelling is movement between different locations in a vehicle. Visiting is a prolonged period spent in one location before moving back to the original location or on to another one. Wandering is moving about — usually on foot — in the local area.

Business process scenario itself is a complex, dynamic network that involves many business stakeholders. Integrating SCOR with the BPCM meta-model for the business process development process will facilitate knowledge sharing and communication among the various parties involved in the processes. As illustrated in Fig. 2, we have incorporated process best practices and process levels from SCOR into the BPCM meta-model.

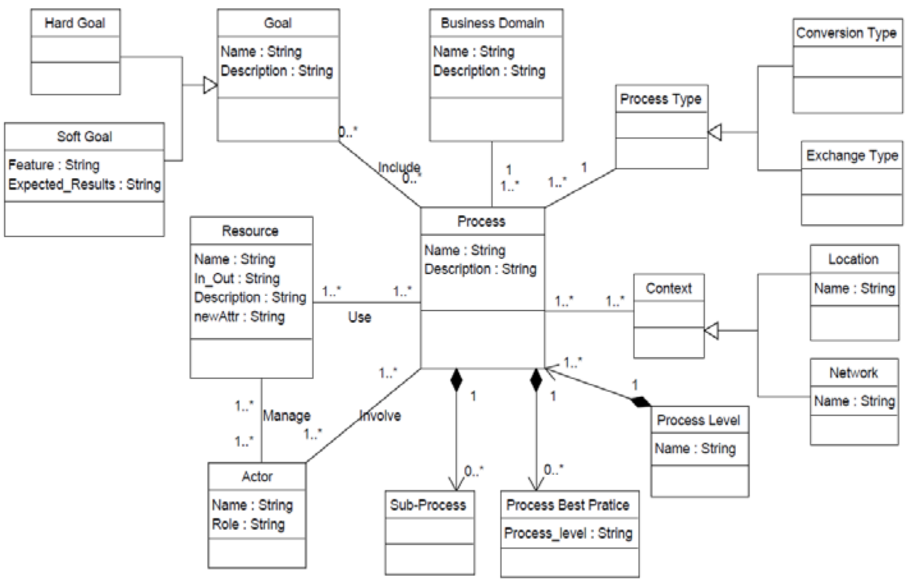

Fig. 2. BPCM Meta Model

4 The BPCM Repository Architecture

This section presents the architecture for the BPCM repository. The architecture, as shown in Fig. 3, follows a three-layer model composed of a presentation layer, a repository management layer and a storage layer. Moreover, the BPCM repository architecture can link to some external systems.

The objective of the architecture proposed here is to organize BPCM models in a good manner to enable their reutilizations. Because of the reusing and sharing facilities of the repository, it is possible for various business users to use the existing knowledge of different business processes, which have already been built by other relevant organizations.

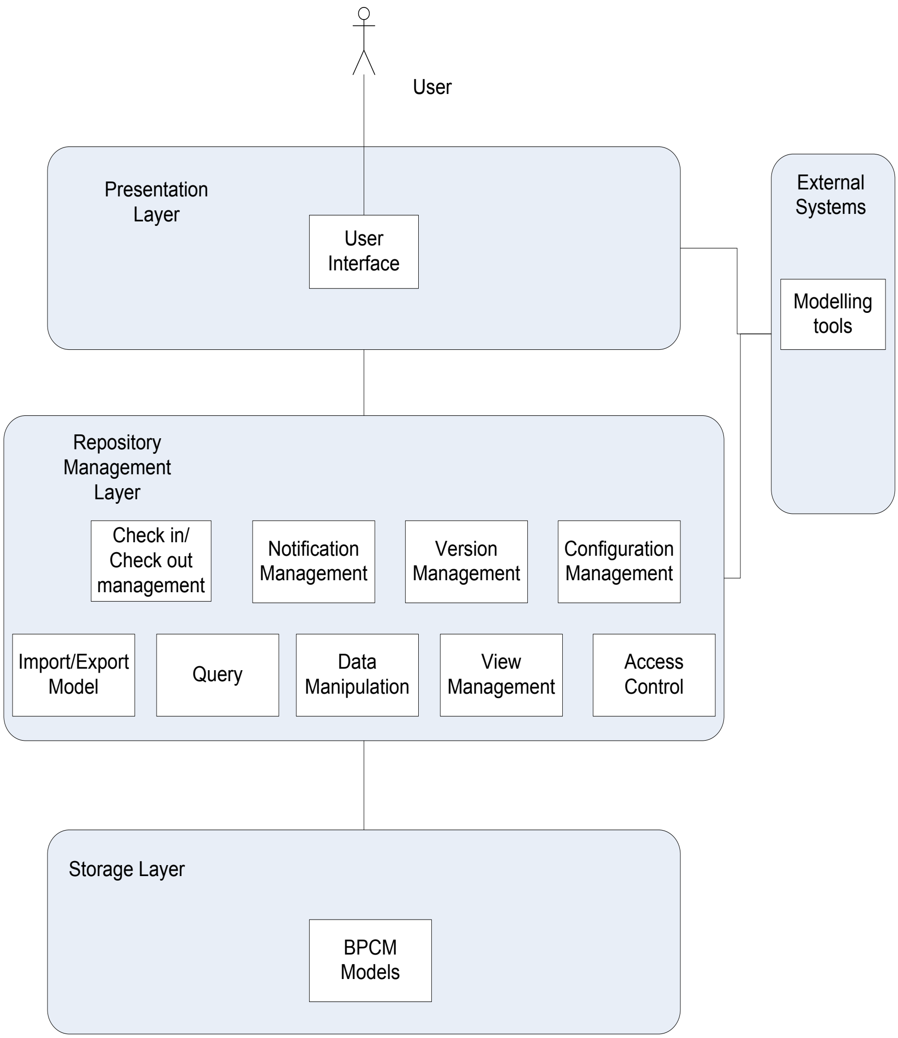

Fig. 3. The BPCM Repository Architecture

The presentation layer provides a graphic user interface for users to interact with the BPCM repository. As a result, the users can easily interact with the various functions provided by the repository. This layer can also allow users to browse and navigate the repository easily, give queries to search for interesting data or BPCM models, and visualize the results in some given formats according to their preferences. For example, given some key words such as a specific business domain, all the related BPCM models in this domain, already existed in the repository, will appear on the user interface. Then, users can review the retrieved BPCM models for various usages.

The repository management layer provides the general amenities of a repository, such as access control, notification management, querying, view management, version management, and check-in/check-out management, etc. The access control amenity can ensure the right people have the right access to the right objects and data in the repository. Notification management enables notifications to be generated in case any change occurs to any object in the repository. Querying functionality enables users to search for the desired information in the repository. Version management allows users to keep track of changes occurred to the specific object in the repository. It makes it possible to maintain multiple BPCM models of the same business process. The view management provides a possibility for users to have multiple views on a BPCM model. Check in/Check out allows users to check in data/objects to the repository or check out data/objects from the repository. For example, a user can obtain a copy of the BPCM model for a specific business process analysis and design from the BPCM repository and store it at a local directory (Check out). A user can also view an extracted BPCM model and make some modifications to the model. Once all the changes are made to the model, the user can export it back to the repository (Check in). It is possible that more than one user are working on the same BPCM model and may be modifying the model at the same time. Thus, it is necessary to offer a solution to deal with it. One possible solution is to simply lock the current model, so that other users cannot make changes on the current version of the model until it is checked back into the repository by releasing the lock.

The storage layer stores the BPCM models in terms of XML formats. This layer aims to provide faster querying services and better navigation services of the BPCM repository. The data structure of the BPCM models in this layer is described in the BPCM meta-model (see Fig. 2). Thanks to the power of meta-model, all the BPCM models are transferable between a tabular format and XML in this layer. Users can add new BPCM models to the BPCM repository. Then, the BPCM models will be stored in the storage layer. From that time onwards, these BPCM models can be accessed by all registered users of the BPCM repository. This layer is mainly invoked by the repository management layer.

In addition, the presentation layer and repository management layer can be made to also interact with external systems (e.g. process modeling tool). For example, technical users can open a process modeling tool (e.g., BPMN modeling tools) from the graphical user interface at the presentation layer of the BPCM repository to model relevant business processes based on the approved BPCM models by business stakeholders. And external systems can invoke the data and objects in the BPCM repository by interacting with the repository management layer.

5 Guidelines to Use the BPCM Repository

In this section, we provide some guidelines for business users to use the BPCM repository. Prior to providing the business requirements to model developers to develop a business process support system, the BPCM repository should be queried by business stakeholders to ensure that the useful best practices and relevant information from the existing BPCM models, which are appropriate to model a process support system from business perspective, has been taken into consideration.

5.1 Identify the Appropriate Candidate BPCM Models

The identification phase aims at identifying potential reusable BPCM models. Some relevant BPCM models are supposed to be extracted from the BPCM repository based on a given search keyword. To limit the retrieved models to a specific business domain, one or more keywords can be used to search for the most relevant BPCM models. This would reduce analysis complexity and time effort of this identification process. Then, the retrieved BPCM models can be assessed according to business stakeholders' needs to select the best BPCM model to be used.

5.2 Redesign the BPCM Model/ Create New BPCM Model

The result of the query against the BPCM repository may be a full match, a partial match, or no match. A full match indicates that the extracted BPCM model is able to provide all business oriented information the business stakeholder expects. In this case, the BPCM model can be entirely reused as a sketch model for business process support system development. But in most cases, the extracted BPCM model may need to be redesigned to better fulfill the needs of business stakeholders or achieve business goals. For example, users may add or delete some attributes to some elements of the BPCM model according to their requirements on the business process. When some changes are made to BPCM models, it is better to annotate the BPCM models with additional information in terms of the version element of the BPCM modeling language. This will help users explore the revision history information on a specific BPCM model more efficiently and effectively for later use. In case no relevant BPCM models could be found from the BPCM repository, the graphical user interface at the presentation layer will allow users to model a completely new BPCM model from sketch, so that other business users may benefit from it in the future.

6 Comparing BPCM Repository with Other Existing Repositories

In this section, some of existing process repositories are presented firstly. Then, we evaluate the similarities and differences between the BPCM repository and the other process repositories.

6.1 Existing Process Repositories

MIT process handbook

The MIT process handbook [18] is a knowledge base of process description. This handbook focuses on organizing knowledge about processes rather than providing detailed process models. The knowledge about the process is provided in a text-based manner. The goal of the MIT process handbook is to develop rich online libraries for sharing and managing many kinds of knowledge about business. A process may be a generalization/specialization of another process, and a process may have other processes as parts or use other processes. Processes available in the repository are grouped into ten root categories: procurement, supply chain management, marketing, sales, information systems, human resources, strategic planning, finance/accounting, manufacturing/ logistics, and engineering.

RepoX

Repox [24] is an XML based process model repository, that has been developed for the METEOR workflow system. It maps XML documents to a relational-object database and also provides extraction/retrieval, version control, check in/check out, and querying functions. It stores the control flow aspect of process models along with the data that is used in the processes. And it is mainly designed as the repository for a workflow engine. In addition, it supports for adaptive workflows, which may need workflow definition information from the repository at runtime, in a dynamically changing environment.

The BPMN repository

The BPMN repository architecture [25] is a conceptual architecture for cross-enterprise processes planning, implementation, and controlling. Core component of the architecture is a distributed repository managing all required data and information, which especially obtains a process-oriented view on collaboration networks. Cross organizational processes need close coordination among networking partners. This is achieved through the integration of business process models. The architecture prescribes that processes and related information should be stored in XML formats.

The Semantic business process repository

The Semantic Business Process Repository (SBPR) [17] is an ontology-based repository for storing business process models. It is used for storage and management of business process modeling artifacts. This repository is open for change (e.g. updating, deletion) by potential users. It requires that the repository is configured with a process ontology. The ontologies must be specified in WSML. It supports querying, versioning and check in /check out.

In SBPR, business process models are enriched by annotating business process artifacts with entities from pre-defined ontologies such as organizational ontologies, Semantic Web Service ontology, business functions ontology, resource ontology, and domain ontologies. It stores instances of process models which are based on ontologies.

SCOR

Supply-Chain Operations Reference Model [5] can be seen as a repository that stores business process related to supply chain management. SCOR is a process reference model designed for effective communications among supply chain partners. SCOR integrates concepts of business processes, benchmarking, and best practices into a cross-functional framework. From a business process reengineering perspective, the SCOR model builds a hierarchy of supply chain processes, which can be divided into three levels of details [5]: process type, process category, and process element. SCOR classifies the operations of supply chain as Plan, Source, Make, Deliver and Return.

BPEL Repository

The BPEL Repository [26] is an Eclipse plug-in originally built for BPEL business processes and other related XML data. This repository uses the XML format as its external format and stores BPEL models as objects in an EMF repository.

The repository can easily be extended with additional XML schemas because of its flexible architecture. Users can query the XML files as EMF objects using an object-oriented query language, namely the Object Constraint Language (OCL) that is part of the UML specification.

6.2 Comparison with Other Process Repositories

In order to compare the BPCM repository with other repositories that were presented in the last section, a number of criteria are identified. The criteria were selected by surveying the literature on requirements for process repositories [7, 22]. The resulting criteria are presented in the column one in the Table 2.

Table 2. Repositories Comparison

	BPCM Repository	MIT Process Handbook	RepoX	BPMN Repository	SBPR	SCOR	BPEL Repository
Control Flow	Not Support	Not Support	Support	Support	Support	Support	Support
Retrieval	Search	Search	Xquery	Not Available	WSML	Search	OCL
Model Storage	XML	Textual	XML	XML	XML	Textual	XML
Integration with other Tools	Yes	No	No	Yes	Yes	No	Yes
Relation with Model Building	Provide Process related Info	Provide Process related Info	Reuse/ Adapt	Reuse/ Adapt	Reuse/ Adapt	Provide Process related Info	Reuse/ Adapt
Goal Inclusion	Yes	No	No	No	No	Partly	No
Domain Specific	No	No	No	No	No	Yes	No
Version Management	Yes	No	Yes	No	Yes	No	No

The BPCM repository and the MIT process handbook do not consider storage of control-flow aspect of process models. More specifically, the BPCM repository focuses on storing tabular descriptions of major characteristic of business processes in XML files, whereas the MIT process handbook aims at storing textual descriptions of the processes and the activities that occur within those processes.

Generally, the repositories can be used in two different ways. In the first category, the extracted models and data can be used as knowledge based foundation to guide developers to model process models. The MIT process handbook, SCOR, and the BPCM repository fall into this category. The second way focuses on reuse of the existing process models/workflows in the repository to develop customized process models. This category includes BPMN repository, BPEL repository and RepoX.

Most repositories use XML based formats for data interchange. The two exceptions are the MIT process handbook, and SCOR. The MIT process handbook is available only through a web-based user interface, through which users interact using natural languages. SCOR is also a web-based textual description resource. Most repositories are able to provide search or query functions in terms of different query languages or search methods.

The BPCM repository, the MIT process handbook, and SCOR are mainly intended to store process related information rather than implementation details of business process models.

All these repositories are mainly meant to store process models and process related information. Most of the repositories in the current formats do not support the integration with external tools. Therefore, we believe that it is beneficial to integrate those repositories with some external modeling tools. That is why we would like to connect the BPCM repository with other external tools.

Concerning the goal inclusion, this is another aspect our BPCM repository distinguishes from the other repositories. Although SCOR includes process metrics, this is only one type of relevant goals to be represented. Goal inclusion means that business goals are given with the process. The BPCM modeling approach allows business users to depict the business goals a process aims to achieve, which in turn the BPCM repository can benefit from.

Only one of the repositories is limited in terms of business scope. SCOR can only store process related information about supply chain management.

The BPCM repository, RepoX, and SBPR can provide facilities for version management.

7 Contributions

Since business process modeling is growing in importance and popularity in most organizations nowadays, business process knowledge storing, sharing and reusing in terms of process repositories are becoming more and more important. In this study, we strive to achieve a shared database of process knowledge through the BPCM repository. The main contribution of this research is the three-layer BPCM repository architecture, including a presentation layer, a repository management layer and a storage layer. Another contribution of this research is the evaluation on the similarities and differences between the BPCM repository and other existing process repositories. This understanding is important to the success of process repositories endeavors and efforts.

Another possible implication that can be drawn from this research is that the BPCM models in the BPCM repository could be used to help design and deploy process models that better match an organization's goals from a business point of view. Since the data for BPCM models could be represented in XML, BPCM models are able to provide a higher level of interoperability with other process models. We hope the insights from this research can draw practitioners' attention to create a better process repository by including the essential characteristics of business processes to facilitate communications between business stakeholders and model developers.

8 Conclusion and Future Work

In this paper, the BPCM repository architecture is presented. It follows a three-layer model composed of a presentation layer, a repository management layer and a storage layer. The BPCM repository architecture is the foundation to provide functions, such as access, import/export, and so on, to manage all major characteristic of business processes of enterprises in terms of BPCM models in the form of XML documents. Since the BPCM meta-model is of help to transfer BPCM models from a tabular format to a XML file, this repository can also be regarded as an XML repository.

Based on the evaluation of the BPCM repository by comparing with some other process repositories, we find that the BPCM repository can offer most features that other process repositories are able to provide with the exception of control flow. This is partially attributed to the fact that BPCM models, which mainly focus on providing a simple view of a business process in terms of characteristics, differ from business process models, which are more about how a business scenario is implemented in terms of executable business process models. We believe this kind of modeling approach is what most business stakeholders expect.

However, it must be admitted that the evaluation of the BPCM repository is currently quite limited since we have not tested it in some case studies. Future research will implement the functions of the BPCM repository and test the usability of the BPCM repository in some business cases. More specifically, we are currently evaluating the usability of the BPCM modeling language in a real conference arangement case. Following an explanation of the BPCM modeling language, how to use the language, and the tabular template to represent a process, participants are asked to characterize the business processes in connection to a secientfic conference series.

References

1. UN/CEFACT Modeling Methodology (UMM) User Guide (2007)
2. Aalst, W.v.d.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41(10), 639–650 (1999)
3. Ali, R., Dalpiaz, F., Giorgini, P. (eds.): A Goal Modeling Framework for Self-contextualizable Software, LNBIP, vol. 29, pp. 326–338. Springer, Heidelberg (2009)
4. Bernstein, P.A., Dayal, U.: An Overview of Repository Technology. In: Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., Santiago de Chile (1994)
5. Council, S.-c. SCOR Model 8.0 Quick Reference Guide (2006)
6. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16(2), 97–166 (2001)
7. Feldmann, R.L. and Nick, M.: Guidelines for Evaluation and Improvement of Reuse and Experience Repository Systems through Measurement Programs. In: The Third European Software Measurement Conference (2000)
8. Fischer, L.: Workflow handbook 2005. Workflow Management Coalition, WfMC (2005)

9. Gao, S., Krogstie, J.: A Combined Framework for Development of Business Process Support Systems. In: Persson, A., Stirna, J. (eds.) The Practice of Enterprise Modeling, vol. 39, pp. 115–129. Springer, Heidelberg (2009)
10. Gao, S., Krogstie, J.: Facilitating Business Process Development via a Process Characterizing Model. In: International Symposium on Knowledge Acquisition and Modeling 2008. IEEE CS Los Alamitos (2008)
11. Gao, S., Krogstie, J.: A Meta-model for a Language for Business Process Characterizing Modelling. In: Popplewell, K., Harding, J., Poler, R., et al. (eds.) Enterprise Interoperability IV, pp. 69–79. Springer, London (2010)
12. Geerts, G.L., McCarthy, W.E.: An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. IEEE Intelligent Systems 14(4), 89–94 (1999)
13. Gordijn, J., Akkermans, H., Vliet, H.V.: Value Based Requirements Creation for Electronic Commerce Applications. In: HICSS 2000. IEEE CS, Los Alamitos (2000)
14. Havey, M.: Essential Business Process Modeling. O'Reilly Media, CA (2005)
15. Hruby, P.: Model-Driven Design Using Business Patterns. Springer, New York (2006)
16. Kristoffersen, S., Ljungberg, F.: Mobility: From stationary to mobile work. In: Braa, K., Sørensen, C., Dahlbom, B. (eds.) Planet Internet (2000)
17. Ma, Z., Wetzstein, B., Anicic, D., et al.: Semantic Business Process Repository. In: The Workshop on Semantic Business Process and Product Lifecycle Managemen 2007. CEUR-WS (2007)
18. Malone, T.W., Crowston, K., Herman, G.A.: Organizing Business Knowledge: The MIT Process Handbook. MIT Press, Cambridge (2003)
19. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment 57, 554–578 (1982)
20. Mendling, J., Nüttgens, M.: EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). Information Systems and E-Business Management 4(3), 245–263 (2006)
21. Salifu, M., Yu, Y., Nuseibeh, B.: Specifying Monitoring and Switching Problems in Context. In: RE 2007. IEEE CS Press, Los Alamitos (2007)
22. Shahzad, K., Andersson, B., Bergholtz, M., et al. (eds.): Elicitation of Requirements for a Business Process Model Repository, pp. 44–55 (2009)
23. Smith, H., Fingar, P.: Business Process Management: The Third Wave. Meghan-Kiffer Press (2003)
24. Song, M., Miller, J., Arpinar, I.: RepoX: An XML Repository for Workflow Design and Specifications. Univeristy of Georgia, USA (2001)
25. Theling, T., Zwicker, J., Loos, P., et al. (eds.): An Architecture for Collaborative Scenarios Applying a Common BPMN-Repository, pp. 169–180 (2005)
26. Vanhatalo, J., Koehler, J., Leymann, F.: Repository for Business Processes and Arbitrary Associated Metadata. In: Demo section at BPM 2006 (2006)
27. Weber, I., Markovic, I., Drumm, C.: A Conceptual Framework for Composition in Business Process Management, pp. 54-66 (2007)
28. White, S.A.: Introduction to BPMN (2005)
29. Zhao, L., Siau, K.: Information Mediation Using Metamodels: An Approach Using XML and Common Warehouse Metamodel. Journal of Database Management 18(3), 69–82 (2007)

A Rule-Based Approach for the Recognition of Similarities and Differences in the Integration of Structural Karlstad Enterprise Modeling Schemata

Peter Bellström

Department of Information Systems, Karlstad University, 651 88 Karlstad, Sweden
Peter.Bellstrom@kau.se

Abstract. In this paper, we address the recognition of similarities and differences in schema integration while applying the notation in the Karlstad Enterprise Modeling approach. In doing so, we describe and present a set of "if -then" rules. In these rules, we make use of both concept name comparison and comparison of concept neighborhoods (their surroundings). Following the classification of schema matching approaches given by [26], the rules are classified as a composite schema-based matching approach. The rules should first of all be viewed as a step towards a semi-automatic method for the recognition of similarities and differences in the integration of structural Karlstad Enterprise Modeling schemata. By applying the proposed rules, several problems, such as homonyms and synonyms, might be recognized that otherwise could pass unnoticed.

Keywords: Schema Integration, Recognition of Similarities and Differences, Karlstad Enterprise Modeling Approach.

1 Introduction

In the information systems development process, we often deal with requirements that are gathered from various sources. These requirements are often represented in the form of structural and behavioral schemata. The structural part represents both what should be stored in the database and what data the information system needs for processing the functionality it should provide. On the other hand, behavioral schemata represent what functionality the future information system should provide. In this paper, we delimit the discussion regarding the recognition of similarities and differences to structural schemata, also known as static schemata. In doing so, we focus on schema integration in conceptual database design. One of the most quoted definitions of schema integration is given by [1] where the authors define schema integration as "the activity of integrating the schemas of existing or proposed databases into a global, unified schema." (p. 323). Schema integration means that the source schemata are prepared, compared, modified and finally merged into one global schema representing parts of, or a whole, database on a conceptual level. In schema integration, it is important to maintain the vocabulary used in the source schemata otherwise semantic loss might occur [5]. Semantic loss is a problem that not only

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 177–189, 2010.

causes interpretation problems but also causes problems related to integration transparency.

In this paper, we present and describe an approach for the recognition of similarities and differences in schema integration. The approach is comprised of a set of rules starting with concept name comparison followed by concept neighborhood comparison. Following the classification of schema matching approaches given by [26], the rules are classified as a composite schema-based matching approach. The rules should first of all be viewed as a step towards a semi-automatic method for the recognition of similarities and differences in the integration of structural Karlstad Enterprise Modeling schemata. By applying the proposed rules, several problems, such as homonyms and synonyms, might be recognized that otherwise could pass unnoticed.

The paper is structured as follows: in section two, we address some examples of related work and distinguish it from our own rule approach. In section three, we present the Karlstad Enterprise Modeling approach with focus on the structural aspects. In section four, we present the main contributions given in this paper, namely the rules for the recognition of similarities and differences in the integration of structural Enterprise Modeling schemata. Finally, the paper closes with a summary together with conclusions and some proposals for future research.

2 Related Work

One way to classify earlier work in the domain of schema integration is to divide it into three parts [4] as follows:

- Manual approaches to schema integration
- Formal approaches to schema integration
- Semi-automatic approaches to schema integration

Manual means that all tasks, such as recognizing similarities and differences, are done manually. Formal means that some type of formal language is used and semi-automatic means that at least one computer application is used. In this section, we focus on the semi-automatic approaches, since the work presented in this paper should be viewed as a first step towards a semi-automatic method for the recognition of similarities and differences in the integration of structural Karlstad Enterprise Modeling schemata.

In [26] the authors present an overview of approaches to automatic schema matching. The authors distinguish between schema-based and instance-based matching, where schema-based matching can be performed for both the concept and its neighborhood. Further, schema based matching might be linguistic or constraint-based. The work presented in this paper is classified as a composite schema-based approach since first we compare concept names followed by a comparison of concept neighborhoods.

In [21] the authors present a method for structural conflict resolution while applying the Entity-Relationship modeling language [11]. Even though the title of the paper focuses on conflict resolution – resolving similarities and differences between two source schemata – the authors still address recognition. In doing so, the authors adopt an assertion-based method. The authors state that a "declarative statement asserting that a modeling construct […] in one schema is somehow related to a modeling construct in another schema is called an inter-schema correspondence assertion or integration assertion for short." (p. 231). Since the authors use the Entity-Relationship modeling language in their method, they focus on a level closer to implementation than the work presented in this paper.

In [28] the authors present a method for solving structural conflicts while applying a modified version of the Entity-Relationship modeling language called Entity-Relationship for complex objects or ERC+ for short. In their approach, the authors point out that conflict resolution is performed without modification of the source schemata. The work presented in [28] focuses on ERC+, a modified version of the Entity-Relationship modeling language, and therefore also focuses on a level closer to implementation than the work presented in this paper.

In [30] the authors present an overview of their approach on how to integrate schemata on the pre-design level while applying the Klagenfurt Conceptual Pre-design Model. A Klagenfurt Conceptual Pre-design Model schema should be viewed as an intermediate schema placed between natural language and more complex conceptual schemata designed with a more traditional modeling language, such as the Unified Modeling Language [24]. The work presented in [30] is in line with the work presented in this paper, meaning integration is done on a high level of abstraction and early in the process of information systems development.

In [19] the authors address schematic discrepancy in the integration of Entity-Relationship schemata. According to the authors, schematic discrepancy is a problem that arises if "the same information is modeled as data in one database, but metadata in another." (p. 245). Schematic discrepancy is also a problem that might occur while applying the Karlstad Enterprise Modeling approach, since in our approach we have a symbol for instance-of (see Fig. 1). In other words, by applying the Karlstad Enterprise Modeling approach, it is possible to show that a concept in one schema is an instance-of a concept in another schema. Applying the instance-of dependency in conceptual database design might indicate several problems in schema integration, such as homonyms, that otherwise could pass unnoticed [3]. In [19] the authors adopt the Entity-Relationship modeling language, which again indicates that the authors are focusing on a level closer to implementation than the work presented in this paper.

Finally, it should be noted that in previous research, the Entity-Relationship modeling language, or some extension of it, has dominated schema integration research since the late 1980´s [27], which is also reflected in this section.

3 The Karlstad Enterprise Modeling Approach

The Karlstad Enterprise Modeling (EM) approach refers to a modeling approach developed to address the *pragmatic*, the *semantic* and the *syntactic* aspects of an information system. The pragmatic aspects of EM are represented by a set of pragmatic dependencies (connections/links) used to illustrate goals, problems, and opportunities together with positive and negative influences [17].

The semantic aspects of EM are represented as both a set of static dependencies (see Fig. 1) and a set of dynamic dependencies. Furthermore, the dynamic dependencies are divided into state and communication dependencies. State dependencies are used to illustrate state changes together with conditions (both pre- and post) for actions. Communication dependencies are used to illustrate relations between actors, their actions and communication flows [16]. By applying the semantic aspects of EM, it is also possible to illustrate and represent system analysis patterns such as sequence, synchronization, iteration, selection and search [18] and to integrate static and dynamic aspects into one schema.

The syntactic aspects of EM are represented by a set of syntactic elements representing databases, database management systems, software components and human components. Syntactic aspects are also viewed as CASE-tool dependent and therefore the elements that are needed can be added on demand, meaning the list is not exhaustive [16].

One way to describe EM is as a generalization and extension of system analysis and design [16]. In [29] the author instead focuses on business processes and describes Enterprise Modeling as an approach that deals with modeling and integration of business processes.

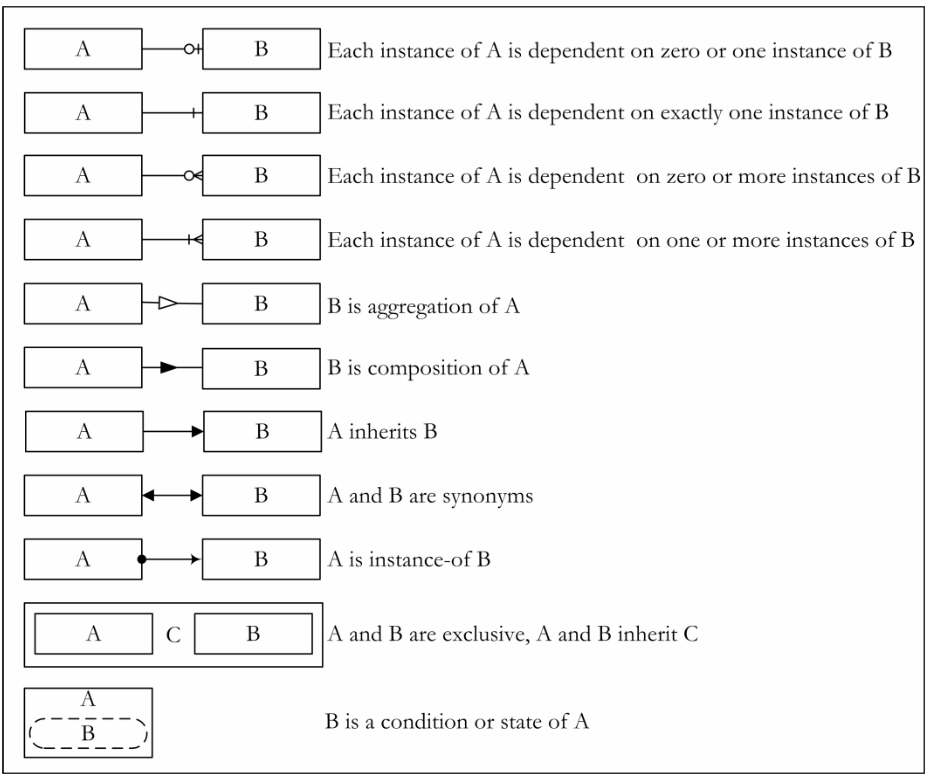

Fig. 1. Representation of static dependencies, structural aspects, in EM (adapted and modified from [16])

The structural aspects of EM (Fig. 1) show that boxes represent concepts and lines between concepts represent dependencies (connections/links). The structural dependencies illustrate what type of relation the concepts have to other concepts. For instance, one concept might be a specialization of another concept and one concept might be an instance-of another concept. It is important to point out that the only primitive, amongst the structural primitives, that is given a name is a concept. This also influences how similarities and differences between two EM schemata might be recognized. Compare this with the Entity-Relationship modeling language which names and distinct between entities, relationships and attributes.

Finally, while applying EM, we do not distinguish between classes and attributes but instead put focus on concepts and dependencies between them. This means that in our approach, we focus on content (what) and not on how the database is going to be implemented. This also means that our approach focuses on issues on a higher level of abstraction compared with traditional modeling languages such as the Entity-Relationship modeling language [11] and the Unified Modeling Language [24].

4 Recognition of Similarities and Differences in Schema Integration

Several approaches and methods for the integration of structural schemata have been proposed during the years. In [1] the authors conclude that an integration method is comprised of, or at least a mixture of, the following four phases: *pre-integration*, *comparison of the schemata*, *conforming the schemata*, and *merging and restructuring the schemata*.

In pre-integration, each single source schema is translated into a canonical modeling language, checked for conflicts and inconsistencies (e.g. intra-schema similarities and differences) and the integration strategy is selected [27]. In comparison of the schemata the schemata are compared aiming to recognize name conflicts, structural conflicts and inter-schema properties [20] (e.g. inter-schema similarities and differences). In conforming the schemata, the similarities and differences recognized in the former phase are resolved. In the fourth and last phase, merging and restructuring the schemata, the schemata are first merged and later on also restructured.

Integration of structural schemata is important since there are often several users, or user groups, producing several schemata representing the same or different parts of a database. Several overlaps often exist between the produced schemata and therefore we need to integrate them into one schema. The method of first designing and later integrating conceptual schemata has been pointed out as important by many researchers. For instance [25] points out that local conceptual schemata preserve and highlight differences in how different users view their organization while a global conceptual schema may instead mask these. In [23] the authors mention that local conceptual schemata may not only prevent premature design decisions but also ensure that all local conceptual schemata are taken into account. Schema integration has also been mentioned as an effective – perhaps the most effective – technique for developing [14][15] and managing [22] large database schemata. Finally, as argued in [12] using a semantic model during conceptual design also simplifies schema integration.

In this paper we put focus on the second phase in the integration method, *comparison of the schemata*, meaning we address how to recognize similarities and differences in the integration of structural Karlstad Enterprise Modeling schemata.

Comparison of the schemata has been described as not only important [27] but also as the most difficult phase in schema integration [13][21] and therefore needs more attention.

Our approach starts with *comparison of concept names* used in two structural schemata. The number of compared schemata is two since binary integration [1][2] is always assumed. If comparison of schema names yields either 'equivalent' or 'similar' the method continues with *comparison of concept neighbors*. This means comparing the concepts and dependencies directly connected to the compared concepts where a name match has been recognized [3][8]. The result of the proposed method might yield that either the compared concepts are '*equivalent*', '*similar*' or *'different'*. However, this distinction is seldom easy to distinguish and therefore the following "if then" rules are proposed for equivalent concept names (E1-E7) and similar concept names (S1-S4) (see also [6]).

Before describing the rules, it should be noted that it is possible to complement the presented rule-based approach with not only a name comparison based on ontology but also a name comparison based on taxonomy [10].

It is also possible to complement the neighborhood comparison with several influence factors such as *polysemy count*, *concept valency* and *domain weight*. These influence factors could be used to relieve and optimize an implementation of our approach, meaning the influence factors could be used to decide if neighborhood comparison is even necessary.

A polysemy count gives the number of meanings a word has in a specific language, concept valency gives the number of parameters a word needs for getting its full meaning and a domain weight can be manually given to concepts by a domain expert.

However, it should be noted that whenever ambiguity exists it is recommended to continue with a neighborhood comparison [10].

4.1 Rules for Equivalent Concept Names

In rules **E1-E7,** name comparison always yields ***equivalent***, meaning the compared concept names (labels) are the same. The rules for equivalent concept names are:

Rule E1: ***If*** comparison of concept names yields *equivalent*, and comparison of concept neighborhoods yields *equivalent* ***then*** *equivalent* concepts might be recognized.

Rule E2: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *different* ***then*** *homonyms* might be recognized between the two concept names in the source schemata. Homonyms occur if one name is used for several concepts with different meanings. For instance using concept name *Name* in the context of *Customer* (schema one) and in the context of *Product* (schema two). Fig. 2 illustrates the recognition of homonyms by applying Rule E2. In Fig. 2, comparison of concept names yields equivalent for the concept named *Name*. However, comparison of concept neighborhoods yields no match, which could indicate homonyms between the two source schemata.

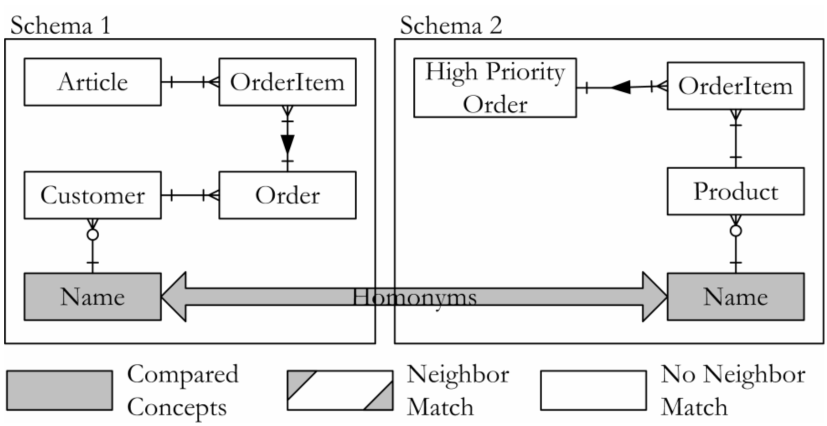

Fig. 2. Recognition of homonyms based on Rule E2

Rule E3: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *similar*, meaning one concept in each source schemata is named different, ***then*** *synonyms* might be recognized between the two concept names. Synonyms occur if several names are used for one concept with the same meaning. For instance using concept name *Article* in schema one and *Product* in schema two. Fig. 3 illustrates recognition of synonyms by applying Rule E3. In Fig. 3, comparison of concept names yields equivalent for the concept named *OrderItem*. However, comparison of concept neighborhoods yields similar meaning; one concept in each schema is named differently, which could indicate synonyms between the two source schemata.

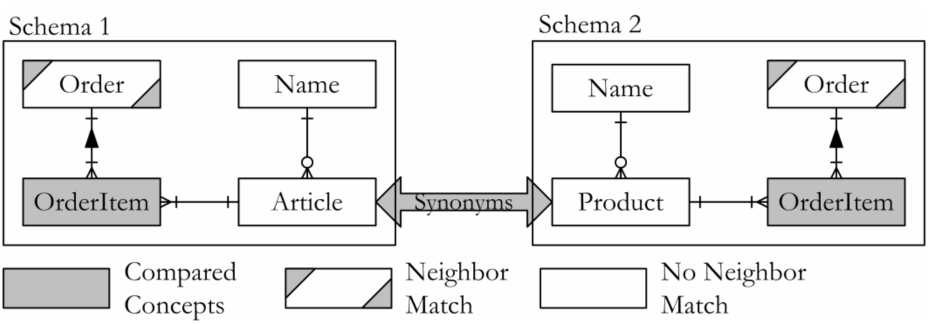

Fig. 3. Recognition of synonyms based on Rule E3

Rule E4: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *similar*, meaning one concept name in one of the source schema is a composite of a concept name in the other source schema with a following addition, and cardinality indicates "one-to-one", ***then*** an *association dependency* between the two concept names might be recognized. An association between two concepts means that they have a trivial dependency (link/connection) to each other and the cardinality is specified. For instance, using *Customer* in schema one and

Customer Number in schema two. Fig. 4 illustrates recognition of an association dependency between *Customer* in Schema 1 and *Customer Number* in Schema 2 by applying Rule E4. In Fig. 4, comparison of concept names yields equivalent for the concept named *Name*. However, comparison of concept neighborhood yields similar meaning; *Customer Number* is a composite of *Customer* in schema 1 with the following addition *Number* and cardinality indicates uniqueness ("one-to-one").

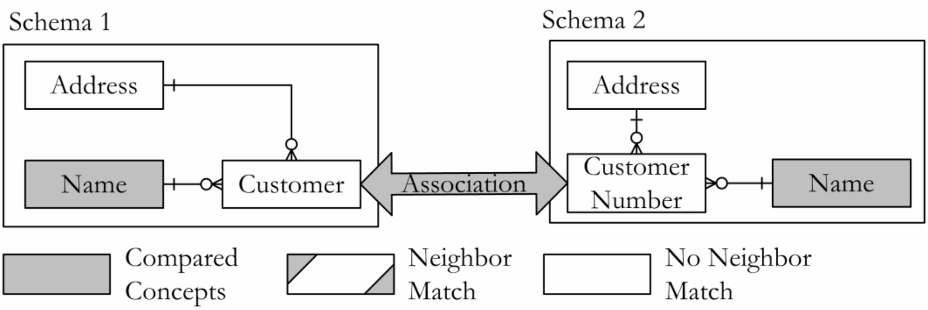

Fig. 4. Recognition of association based on Rule E4

Rule E5: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *similar*, meaning one concept name in one of the source schemata is a composite of a concept name in the other source schema, with a prior addition, ***then*** a *hypernym-hyponym dependency* might be recognized between the two concept names. A hypernym-hyponym dependency occurs if one concept is recognized as a specialization of another concept, which at the same time is recognized as a generalization of the other concept. A hypernym-hyponym dependency is also known as inheritance (is-a) and as generalization-specialization (see Fig.1). For instance, using *High Priority Order* (hyponym) in schema one and *Order* (hypernym) in schema two. In Fig. 5, comparison of concept names yields equivalent for the concept named *OrderItem*. However, comparison of concept neighborhood yields similar neighbor match, which could indicate a hypernym-hyponym dependency between the two source schemata.

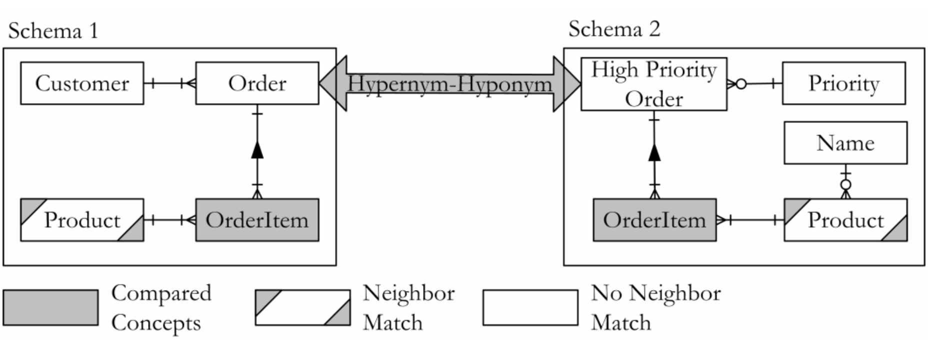

Fig. 5. Recognition of a hypernym-hyponym dependency based on Rule E5

Rule E6: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *similar*, meaning one concept name in one of the source schema is a composite of a concept name in the other source schema with a following addition and cardinality indicates uniqueness with "many", ***then*** a *holonym-meronym dependency* (composition) between the two concept names might be recognized. A holonym-meronym dependency occurs if once concept is recognized as a part of another concept, which at the same time is recognized as composed (see Rule E6) or aggregated (see Rule E7) of the other concept(s). The holonym-meronym dependency is also known as aggregation and composition (see Fig. 1). For instance, using *Order* in schema one and *OrderItem* in schema two. In Fig. 6, comparison of concept names yields equivalent for the concept named *OrderItem*. However, comparison of concept neighborhood yields similar neighbor match, which could indicate a holonym-meronym dependency between the two source schemata.

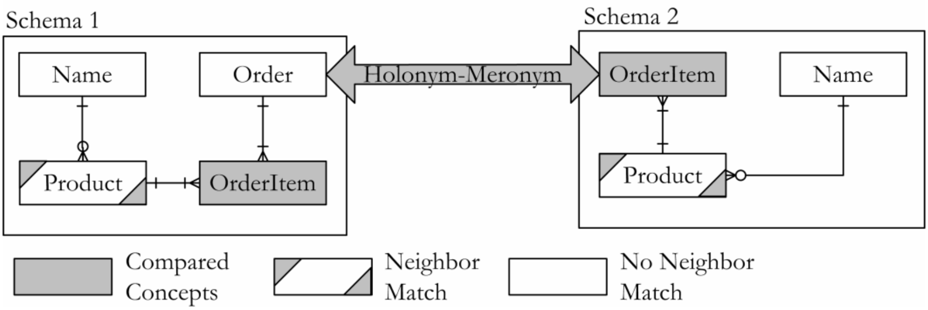

Fig. 6. Recognition of a holonym-meronym dependency based on Rule E6

Rule E7: ***If*** comparison of concept names yields *equivalent* and comparison of concept neighborhoods yields *similar*, meaning one concept name in one of the source schema is a composite of a concept name in the other source schema with a following addition and the cardinality between the two concepts indicates "many" without uniqueness, ***then*** a *holonym-meronym dependency* (aggregation) between the two concept names might be recognized. For instance using *Order* in schema one and *OrderLine* in schema two.

A few issues need to be addressed in the presented rules. Depending on how the dependencies are used in the source schemata, rules four, five and six could indicate either sharper or looser dependencies. This means that rule E4 could indicate an association or a composition (part-of) dependency and rule E5 and E6 could indicate an association dependency. It should also be noted that in schema integration synonyms and homonyms are often called name conflicts [1] or semantic conflicts [27] and hypernym-hyponym and holonym-meronym dependencies as inter-schema properties [20]. An inter-schema property occurs if two source schemata have certain constraints in common [20] such as a hypernym-hyponym dependency and a holonym-meronym dependency.

4.2 Rules for Similar Concept Names

In rules **S1-S4**, name comparison always yields ***similar***, meaning the compared concept names (labels) are not exactly the same but very similar. Rules S1-S4 can be viewed as a refinement and extension of the two rules expressed and proposed in [10] stated as:

- **Rule 1:** If concept A is atomic and concept AB is a composite consisting of A followed by another word B then the concept AB 'belongs-to' concept A. *Example: "Customer" and "Customer Name" → "Customer Name" 'belongs-to' "Customer".*
- **Rule 2:** If concept A is atomic and concept BA is a composite consisting of a word B followed by word A then the concept BA 'is-a' concept A. *Example: "Order" and "High Priority Order" → "High Priority Order" 'is-a' "Order"*. (p. 112)

However, the rules expressed and proposed in [10] are proposed for any modeling language, meaning the rules are modeling language independent, while the rules described in this paper are adapted for the Karlstad Enterprise Modeling approach. The rules for similar concept names are:

Rule S1: ***If*** comparison of concept names yields *similar*, meaning one concept name and one concept name with a following addition to the first one, and comparison of concept neighborhoods yields *similar or equivalent* with a "one-to-one" in cardinality ***then*** an *association dependency* between the two concept names might be recognized. For instance using *Customer* in schema one and *Customer Id* in schema two.

Rule S2: ***If*** comparison of concept names yields *similar*, meaning one concept name and one concept name with a following addition to the first one, and comparison of concept neighborhoods yields *similar or equivalent* with uniqueness and "many" in cardinality ***then*** a *holonym-meronym dependency* (composition) between the two concept names might be recognized. For instance using *Order* in schema one and *OrderItem* in schema two.

Rule S3: ***If*** comparison of concept names yields *similar*, meaning one concept name and one concept name with a following addition to the first one, and comparison of concept neighborhoods yields *similar or equivalent* and the cardinality indicates "many" without uniqueness ***then*** a *holonym-meronym dependency* (aggregation) between the two concept names might be recognized. For instance using *Order* in schema one and *OrderLine* in schema two.

Rule S4: ***If*** comparison of concept names yields *similar*, meaning one concept name and one concept name with a prior addition to the first one, and comparison of concept neighborhoods yields *similar or equivalent*, ***then*** a *hypernym-hyponym dependency* might be recognized between the two concept names. For instance using *High Priority Order* (hyponym) in schema one and *Order* (hypernym) in schema two.

Occasionally it could be useful to continue with comparison of concept neighborhoods even if the comparison of concept names yields different (see also [10]). In doing so, synonyms that otherwise could pass unnoticed might be recognized (see Fig. 7). The result of a comparison of concept neighborhoods, with the prior result of different from comparison of concept names, could result in three cases. If different, then the concept names are different or if similar or equivalent then concept names could be synonyms. In line with the issues addressed in connection with rules E4-E6, rules S1-S3 could indicate either sharper or looser dependencies. In Fig. 7, two concept names yield equivalent (see *Name* and *OrderItem*). Based on these two matches, a neighborhood comparison might be conducted indicating that *Article* in Schema one and *Product* in Schema two are synonyms.

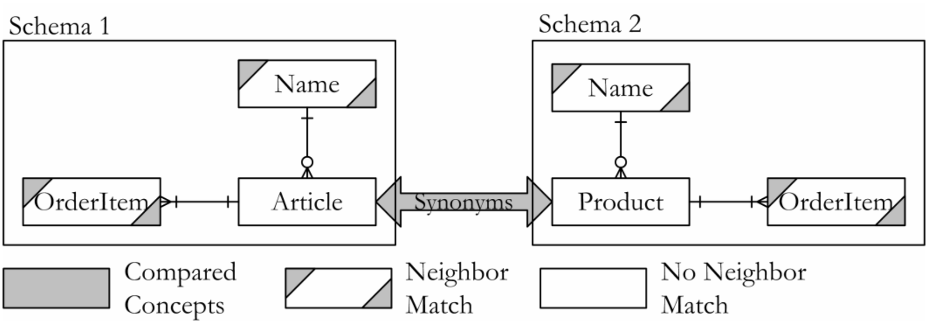

Fig. 7. Recognition of synonyms based on neighborhood comparison

5 Summary and Conclusions

In this paper we have described and illustrated a rule-based approach for the recognition of similarities and differences in the integration of structural Karlstad Enterprise modeling schemata. Stating the chosen modeling language and approach is very important since the chosen modeling language often influences not only how similarities and differences between two source schemata are recognized but also how similarities and differences between two source schemata are resolved in the integration process.

In the Karlstad Enterprise Modeling approach, structural and behavioral aspects are closely connected, they might even be integrated into one schema, and similar problems, such as homonyms and synonyms, might occur while integrating each type of schema (structural and behavioral). Therefore the work presented in this paper is also partially applicable to the integration of behavioral schemata. Integration of behavioral schemata is a topic that will be addressed in depth in future research. However, before integrating the behavioral schemata it is preferable to integrate the structural schemata. The motivation for this is that then several problems, such as homonyms and synonyms, will already have been resolved [7]. Additionally, the integrated structural schemata can be used as an information resource while integrating the behavioral schemata [9].

The presented and illustrated rule-based approach should first of all be viewed as a step towards a semi-automatic method for the recognition of similarities and differences in the integration of structural Karlstad Enterprise Modeling schemata. As shown in section four, by applying the proposed rules several problems, such as homonyms, synonyms, associations, hypernyms-hyponyms and holonyms-meronyms, might be recognized that otherwise could pass unnoticed.

Future research will focus on developing a semi-automatic method for the integration of structural Karlstad Enterprise modeling schemata. In future research we will also address methods and approaches, both manual and semi-automatic, for the integration of behavioral Karlstad Enterprise modeling schemata.

References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18(4), 323–364 (1986)
2. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design An Entity-Relatinship Approach. The Benjamin/Cummings Publishing Company, Redwood City (1992)
3. Bellström, P.: Using Enterprise Modeling for Identification and Resolution of Homonym Conflicts in View Integration. In: Vasilecas, O., Caplinskas, A., Wojtkowski, W., Wojtkowski, W.G., Zupančič, J., Wrycza, S. (eds.) Information Systems Development Advances in Theory, Practice, and Education, pp. 265–276. Springer, Heidelberg (2005)
4. Bellström, P.: View Integration in Conceptual Database Design Problems, Approaches and Solutions. Licentiate Thesis. Karlstad University Studies 2006:5 (2006)
5. Bellström, P.: On the Problem of Semantic Loss in View Integration. In: Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (eds.) Information Systems Development Challenges in Practice, Theory, and Education, pp. 963–974. Springer Science, Heidelberg (2009)
6. Bellström, P.: Schema Integration – How to Integrate Static and Dynamic Database Schemata. Dissertation. Karlstad University Studies 2010:13 (2010)
7. Bellström, P., Jakobsson, L.: Towards a Generic and Integrated Enterprise Modeling Approach to Designing Databases and Software Components. In: Nilsson, A.G., Gustas, R., Wojtkowski, W., Wojtkowski, S.W., Wrycza, S., Zupancic, J. (eds.) Advances in Information Systems Development: Bridging the Gap between Academia and Industry, pp. 635–646. Springer Science, Heidelberg (2006)
8. Bellström, P., Vöhringer, J., Salbrechter, A.: Recognition and Resolution of Linguistic Conflicts: The Core to a Successful View and Schema Integration. In: Magyar, G., Knapp, G., Wojtkowski, W., Wojtkowski, W.G., Zupančič, J. (eds.) Advances in Information Systems Development: New Methods and Practice for the Networked Society, pp. 77–87. Springer Science, Heidelberg (2007)
9. Bellström, P., Vöhringer, J., Kop, C.: Guidelines for Modeling Language Independent Integration of Dynamic Schemata. In: Pahl, C. (ed.) Proceedings of the IASTED International Conference on Software Engineering, pp. 112–117. ACTA Press (2008)
10. Bellström, P., Vöhringer, J.: Towards the Automation of Modeling Language Independent Schema Integration. In: Kusiak, A., Lee, S.-G. (eds.) Proceedings International Conference on Information, Process, and Knowledge Management, pp. 110–115. IEEE Computer Society, Los Alamitos (2009)
11. Chen, P.: The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
12. Ekenberg, L., Johannesson, P.: Conflictfreeness as a Basis for Schema Integration. In: Bhalla, S. (ed.) CISMOD 1995. LNCS, vol. 1006, pp. 1–13. Springer, Heidelberg (1995)
13. Ekenberg, L., Johannesson, P.: A Formal Basis for Dynamic Schema Integration. In: Thalheim, B. (ed.) Conceptual Modeling, ER 1996, pp. 211–226. Springer, Heidelberg (1996)

14. Frank, H., Eder, J.: Integration of Behavioral Models. In: Proceedings of the ER 1997 Workshop on Behavioral Models and Design Transformations: Issues and Opportunities in Conceptual Modeling [Electronic] (1997), http://osm7.cs.byu.edu/ER97/workshop4/fe.html (04-08-2010)
15. Frank, H., Eder, J.: Integration of Statecharts. In: Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems, pp. 364–372. IEEE Computer Society, Los Alamitos (1998)
16. Gustas, R.: Gustiené. P.: Towards the Enterprise Engineering Approach for Information System Modelling Across Organisational and Technical Boundaries. In: Camp, O., Filipe, J., Hammoudi, S. & Piattini, M. (eds.) Enterprise Information Systems V. pp. 204–215. Kluwer (2004)
17. Gustas, R., Gustiené, P.: Pragmatic-Driven Approach for Service-Oriented Analysis and Design. In: Johannesson, P., Söderström, E. (eds.) Information Systems Engineering: From Data Analysis to Process Networks, pp. 97–128. IGI Global (2008)
18. Gustas, R., Gustiené, P.: Service-Oriented Foundation and Analysis Patterns for Conceptual Modelling of Information Systems. In: Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (eds.) Information Systems Development Challenges in Practice, Theory, and Education, pp. 249–265. Springer Science, Heidelberg (2009)
19. He, Q., Ling, T.W.: Resolving Schematic Discrepancy in the Integration of Entity-Relationship Schemas. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.W. (eds.) ER 2004. LNCS, vol. 3288, pp. 245–258. Springer, Heidelberg (2004)
20. Johannesson, P.: Schema Integration, Schema Translation, and Interoperability in Federated Information Systems. Dissertation. Stockholm: Department of Computer & Systems Sciences, Stockholm University (1993)
21. Lee, M.L., Ling, T.W.: A Methodology for Structural Conflict Resolution in the Integration of Entity-Relationship Schemas. Knowledge and Information System 5(2), 225–247 (2003)
22. Mannino, M.V.: Database Design, Application Development, & Administration. McGraw-Hill/Irwin, Irwin (2007)
23. Nuseibeh, B., Easterbrook, S., Russo, A.: Making Inconsistency Respectable in Software Development. The Journal of Systems and Software 58, 171–180 (2001)
24. Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure. [Electronics] (2009), http://www.omg.org/spec/UML/2.2/Superstructure/PDF/ (25-01-2010)
25. Parsons, J.: Effects of Local Versus Global Schema Diagrams on Verification and Communication in Conceptual Data Modeling. Journal of Management Information Systems 19(3), 155–183 (2003)
26. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10, 334–350 (2001)
27. Song, W.: Schema Integration – Principles, Methods, and Applications. Dissertation. Stockholm: Department of Computer and Systems Sciences, Stockholm University (1995)
28. Spaccapietra, S., Parent, C.: View Integration: a Step Forward in Solving Structural Conflicts. IEEE Transactions on Knowledge and Data Engineering 6(2), 258–274 (1994)
29. Vernadat, F.B.: Enterprise Modeling and Integration: principles and applications. Chapman & Hall, Boca Raton (1996)
30. Vöhringer, J., Mayr, H.C.: Integration of Schemas on the Pre-Design Level Using the KCPM-Approach. In: Nilsson, A.G., Gustas, R., Wojtkowski, W., Wojtkowski, W.G., Wrycza, S., Zupančič, J. (eds.) Advances in Information Systems Development: Bridging the Gap between Academia and Industry, pp. 623–634. Springer Science, Heidelberg (2006)

Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games

S.J.B.A. (Stijn) Hoppenbrouwers and I. (Ilona) Wilmont

Radboud University Nijmegen
Institute for Computing and Information Sciences
Heijendaalseweg 135
6525 AJ Nijmegen, The Netherlands
stijnh@cs.ru.nl, ilona@cs.ru.nl

Abstract. This paper reports on a next step in a line of research taking the perspective that modelling as an activity is a 'constrained conversation'. We focus on concrete *communication situations* in context of (enterprise) modelling sessions, with special attention to the involvement of 'novice modellers'. We present some theoretical notions that are helpful in understanding why modelling performed by/with novice modellers can usually be best broken down in sub-tasks, and how such decomposed tasks can be analysed and structured to match the limited skills of (novice) modellers. The generic aspects presented are then linked to generic types of *questions and answers* that are both drivers and constraints for the 'dialogue games' played in conversations-for-modelling. We also present and illustrate an instrument for analysis, the 'Focused Conceptualisation' (FoCon), which can help identify, evaluate and create dialogue games for model-oriented communication situations; we discuss three working examples of the use of FoCons.

Keywords: Conceptualisation, enterprise modelling, dialogue games, collaborative modelling.

1 Introduction

The context of this paper is the study of what goes on, essentially, when people engage in enterprise modelling. Through such insight, we hope to eventually improve the practice in applied situations. We present a next step in our inquiries of modelling as a specialized class of conversation [1]. Earlier work in this line has identified goals of modelling [2], and explored the basic conversational view on modelling and its elementary units of analysis [3]. We now focus on concrete *communication situations*.

Modelling is increasingly becoming a required activity in many situations [4,5]. Often, participatory principles in system design require that modelling efforts, or at least some phases in them, become more accessible to participation of what we might call "novice modellers": people who are typically untrained in modelling, and who do

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 190–204, 2010.

not view modelling as a professional skill or responsibility relevant to them personally. In practice, novice modellers are typically non-technical stakeholders.

We observe, as before [5], that current tools for modelling are mostly expert-oriented editors, providing hardly any active support for the challenging, creative process of conceiving and formulating a good abstract model. Support of a ‘way of working’ may not be needed or even wanted by expert modellers (who often find a prescribed modelling procedure an impeding factor rather than a help [6], and just want an aid for representing the model they conceive) but in cases where novice modellers are involved, methods tailored to support a stepwise way of working that novices can understand and deal with can be of considerable value. While we embrace the use of formal modelling languages, we emphasise that the cognitive burden their direct, unguided use places on the novice modeller is often too high [7]. Hence we study ways of making formal conceptualisation easier and more accessible without abandoning formality as such.

The main question underlying this paper is: ‘how can we understand and frame the questioning-and-answering that constitutes conversations for modelling, and as an extension of this, how can we usefully analyse concrete model-oriented communication situations, in particular in view of their accessibility to novice modellers’.

We will first provide some theoretical background and relevant concepts. Next we will discuss stepwise, focused questioning and answering within modelling processes, and how this corresponds to the existing notion of *dialogue game*. We will then discuss in detail a novel ‘operational method analysis’ concept, the Focused Conceptualisation, and illustrate it. Finally, we will conclude and suggest directions for further work.

2 Theoretical Background: Modelling As a Conversation

With ‘modelling’ we here mean *the purposeful creation of structured and coherent texts or graphical artefacts and subject to strong conceptual (and other) constraints*. Such constraints may for example pertain to syntax (often with some mathematical semantics underlying it), the domain targeted, aspects focused on, and the level of agreement to be reached among co-modellers [2]. We take a goal-driven perspective on modelling: every model serves or works towards at least one clear, utilitarian purpose [4]. Examples of such models include all regular types of modelling as encountered in enterprise engineering, information systems engineering, and software engineering (most prominently, requirements engineering). Similar practices can also be found in fields like knowledge engineering, system dynamics and operations research.

Regular, language-oriented textbook methodology does not offer much help concerning detailed, situation-specific ways of working. This has led us to look for specialized ways of researching *operational modelling*, i.e. the detailed (inter)actions performed by people when they model, and how this relates to resulting models, modelling languages, procedures, and tools [3]. We have found it very useful to view modelling methods and procedures as *interactive systems*. The activity of modelling is strongly language-oriented and produces explicit conceptual constructs (the model);

in particular if conducted collaboratively, it also involves communication *about* the model. It is quite appropriate in this vein to view the model as a *text* (in the theoretical sense [8], consisting of interrelated *propositions* [1]. These propositions can be represented by graphical or textual means; often, a combination thereof. This implies that modelling processes are *conversations* leading to *models*. The three basic concepts in our view on modelling processes are shown in Fig. 1.

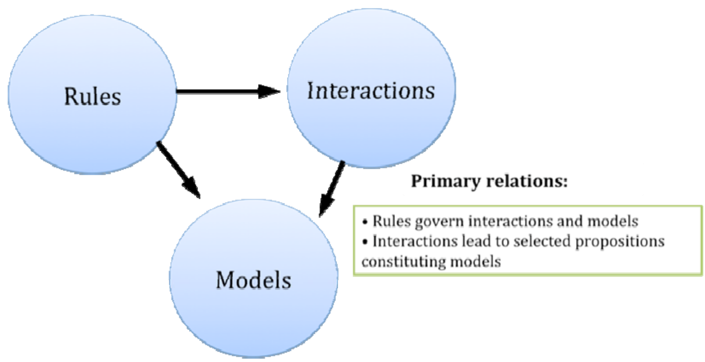

Fig. 1. The RIM framework

The Rules essentially serve to constrain the conversations-for-modelling. We distinguish between 3 types of rules: goal rules, interaction rules and procedural rules. *Goal rules* define the goals set for the conversation-for-modelling: what sort of model it is to render. The tighter the goal rules, the more constrained the product of the conversation. The most important goals-for-modelling within goal rules are *content goals, syntax goals, validation goals* and *argumentation goals* [2]. *Interaction rules* constrain interaction itself. They may simply be the implicit set of conventions for acceptable and anticipated conversation patterns [9], but they may also be more explicit and specialized. Finally, *procedural rules* determine the order in which certain activities within the process are to be carried out ('workflow'). Such activities are typically related to goals and sub-goals of the modelling effort at large.

We can also define a fairly standard set of interaction types [3]: *propositions, questions, agreements, disagreements, arguments, clarifications, acceptations, rejections*. These are in line with categories commonly used in fields like argumentation theory and conversation analysis. We have by now analysed a number of conversations-for-modelling using the RIM framework, refining and validating it to a reasonable degree.

Two of the RIM framework's three basic concepts, *rules* and *interactions*, reflect the basic elements of what in Argumentation Theory [10] is referred to as Dialogue Games. [11] reports on an implemented educational interactive system called InterLoc, designed for creating and playing collaborative versions of dialogue games (fig. 2). Another implementation illustrative of our approach is CMapTools [12], a software package supporting the creation of Concepts Maps, which are explicitly defined and implemented as graphical representations of sets of textual propositions [13], thereby aptly mirroring the third RIM concept: 'model'. Combining InterLoc and CMapTools, InterLoc's structured dialogue games can indeed render the propositions that can then be visualised and refined by means of Concept Maps. Both applications are rather basic and do not support advanced meta-models (ways of modelling) or interfaces (editors) of the kind needed for adequate support of specialized modelling efforts. Combined, they do, however, convincingly illustrate the basic workings of the RIM framework and show that it is at the least a viable and realistic view on the process of modelling.

Game Master
(1) This is the problem variable: The number of regular first year students signed up for the Information Science (IS) curriculum in august of a particular academic year.
Game Master
(1) This is the problem variable: Let's call the variable "Intake"
Game Master
Directive of the Game Master: Please list 5-10 ideas of factors you think may influence or be influenced by the problem variable.
Game Master
(1) Please propose and IDEA and if possible a VARIABLE, [player]: Player 1?
Player 1
(1) I propose the following IDEA: Requirements for high school profile
Player 2
I have a question about this proposition What do you mean?
Player 1
I would like to clarify this: I mean the skill required by high school students. i.e. if they have math in their profile
Game Master
Directive of the Game Master: could you name a VARIABLE for this?
Player 1
I have a question: I'm not sure how to quantify this. I simply mean that there could be requirements to enter the study and high school students either comply to them or not
Player 2
I have a remark: you have to be able to count it. So why not: "number of people that have a fitting profile"?
Player 1
(1) I propose this VARIABLE for the idea: number of high school graduates with a fitting profile
Player 2
I accept the proposition

Fig. 2. An example fragment of a logged InterLoc conversation (part of step 1 in the GMB dialogue game: see section 4.2)

We are fully aware that constrained conversations-for-modelling as illustrated in fig. 2 may serve well as a theoretical and methodological basis for analysing and reasoning about modelling processes, but that as an operational interface for interactive systems supporting modelling, they leave much to be desired. We therefore emphatically distinguish between the *deep interface* and the *surface interface* of such systems.

The deep interface closely mirrors the RIM framework and distinguishes, enables and (possibly) logs interactions one-by-one, as the moves in a dialogue game, thus creating and logging an incremental set of deltas on the model that is created. A surface interface (*not* illustrated in fig. 2) should support more efficient and user-friendly execution of deep surface interactions, including graphical editing, textual editing, form filling, and so on.

Still, every interaction at surface level can in principle be reanalysed as a set of related interactions at deep level. Note that this implies that any existing setup for the support of a modelling effort (even very low-tech setups) can be analysed as a dialogue game under the RIM framework, at the level of the deep interface. This provides a generic analytical instrument across all sorts of operational modelling sessions.

3 Focus Questions and Abstract Conceptualisation

In this section we discuss an issue key to understanding the sort of abstract conceptualisation that is modelling: what essential (types of) questions are asked in view of a modelling effort, and how do these relate to constrained abstract conceptualisation as performed using certain 'ways of modelling' (modelling languages)?

3.1 Abstract Conceptualisation: Three Activity Types Distinguished

Modelling is largely about abstraction (or rather, abstract conceptualisation). As explained in [14]:

> "There are many ways to define abstraction, depending on which perspective is taken. In fields such as philosophy, mathematics and logic abstraction is characterised as information neglect: "eliminating specificity by ignoring certain features" [15]. […] The highly dynamic and interactive nature of computer science is fundamentally different and therefore requires a different interpretation: information hiding. A key concept in information hiding is the deliberate omission of irrelevant information so that the focus is only on the relevant aspects of conceptualisation. […] Human beings engage in abstraction all the time, often without noticing it. Abstraction always requires a form of representation or description, making it inextricably bound to language."

We observe that in the conceptualisation of abstract propositions there can typically be a distinction between three different flavours of activity:

- *generation* of proposals (by 'thinking them up' or alternatively by importing them from existing sources)
- *classification* of the proposals (e.g. according to topic or aspect, or according to word forms or meaning categories: 'nouns', 'verbs'; 'causes', 'activities', 'objects'; and so on)
- *selection* of certain categorised concepts based on criteria matching the conceptualisation goals set (i.e. the modelling goals).

Thus, a generative aspect (idea formulation) is combined with a constraint aspect (checking of conformance to pragmatic, syntactic and semantic restrictions of various sorts). It is quite possible that in operational conceptualisation, the three activities are in fact collapsed into a single visible activity, i.e. in such cases no distinct intermediate products are created except perhaps in the mind of the participant. However, in many other cases (in particular those involving novice modellers) one or more of the activities are performed in separation. This seems to be a general property of explicit conceptualisation, not one unique for specification and modelling [16].

3.2 Two Basic Foci in Conceptualisation

Zooming in on the types of focus used in conceptualisation, we observe that in modelling practice, two main foci can be distinguished: focus on information that is deemed relevant for the purpose of the model (let us call this *pragmatic focus*) and focus on concepts that fit some prescribed semantic and/or syntactic frame of description: a modelling language (this we call the *semantic-syntactic focus*). Both pragmatic focus and semantic-syntactic focus can play a role in the generation, classification and selection of propositions.

Pragmatic focus concerns the *purpose for which the domain description is created*. What is the description to be used for? Explaining to someone how to do something? Where to find something? Why things in the domain are how they are? What problem to solve? What to build? What is and is not allowed? Such questions bear great resemblance to what [13, p19] call “focus questions”. They are also (more distantly) related to aspects of text functionality [17].

People intuitively know how to formulate a description for specific pragmatic purposes, and they are normally quite capable of fine-tuning their description for the goal it is meant to serve. Explorative research [14] suggests that they will intuitively prefer the use of certain semantic/syntactic conceptual categories (e.g. activities, objects, is-a relations, causes, etc.) as they aim for a certain type of description. However, this indeed happens very intuitively, and within the boundaries and means of normal natural language. Hence, applying a pragmatic focus is to a considerable degree accessible to novice modellers: it is largely a 'natural' capacity in people, connected to pragmatic language skills (using language in specific contexts, for specific situational goals).

The second type of focus (semantic-syntactic) is not so 'natural', and is associated with the use of restricted artificial languages with some special purpose syntax. It concerns form and concept classification rather than ‘content’: actively selecting certain types of conceptual categories, instantiating them, and combining them into structures with strong limitations. Such constrained syntax is typically introduced for very good reasons to support a particular, limited goal in language expression (for example, describing process flow in terms of Petri Nets, or entities and relationships in terms of first order predicate logic). They do, therefore, provide a kind of focus that is imminently useful in particular modelling contexts, and that is in line with some pragmatic focus –indeed it very much supports the 'intention of the description' as seen from the engineering side. The problem is that to use a constrained syntax in such a way demands special skills to a much higher degree than does a pragmatic focus [7].

3.3 Discussion: Pragmatic and Semantic-Syntactic Focus in Practice

Typically, pragmatic focus should be leading in the choice of modelling language (right language for the job); we observe, on the side, that in practice this is not always the case, and that indeed this can seriously hamper realization of proper pragmatic focus and delivery on pragmatic goals.

In ‘informal modelling’ the pragmatic focus (what the model is for) need not necessarily be augmented by a constrained modelling language: such use of a formal language is typically due to additional requirements on the language of model formulation, usually of a technical or mathematical nature. In other words, pragmatic focus can be pretty well achieved if descriptions are phrased in informal diagrams or natural language, but additional goals-for-modelling may still demand a strong semantic-syntactic focus. Note that it is also possible to adhere to the semantic-syntactic focus but still model irrelevantly because of a flawed pragmatic focus.

Concerning the semantic-syntactic focus, it is a well known phenomenon in modelling that a specific aspect of a domain may well be modelled in various ways (for example, imperatively or declaratively: as a flow or as rules) but that if the mindset of the modeller is not clear on one specific paradigm/meta-model from the beginning, an entangled mess of concepts may be the result. In other words, meta-models do not mix well in conceptualisation (in particular if they are based on differing modelling paradigms). So from a semantic-syntactic focus point of view it is in principle a bad idea to start off with a 'general (unfocused) exploration of what the domain looks like' if a constrained syntax is what is aimed for, while pragmatic focus may well call for such a generic overview. For semantic-syntactic focus, the sharper a focus question is put from the very start, the better it is: it focuses the mind of the domain expert and in her conceptualisation makes her weed out concepts and propositions irrelevant to the goal of the model that is being constructed. If the semantic-syntactic focus is insufficiently clear, the description will become less sharp, leading to the introduction in early versions of the conceptualization of concepts and propositions that may be valid in the sense that they reflect a real and even significant aspect of the domain, but that are not *relevant* to the specific modelling task at hand.

Disentangling a mixed conceptualization is a job that is not for the faint-hearted; it is usually best to go back to start and reconceptualise the model. While this may not always be avoidable, it is time consuming and frustrating for all involved. Starting off instead with a clear semantic-syntactic goal/focus in mind (in this case, a clear meta-model) therefore helps greatly if clean and efficient conceptualization is to take place. Unfortunately, this means that 'generic and free descriptions' of domains, that are intuitively useful as a starting point for novices and that for pragmatic reasons may well be desirable, can in fact very much hinder the semantic-syntactic focus in conceptualization down the road. This is a tricky point of which we should be well aware.

So for a good model to serve some clear pragmatic goal (and assuming a constrained meta-model is needed for analytical or engineering purposes), *both* the pragmatic focus and the semantic-syntactic focus are required. Expert modellers are usually capable of combining pragmatic and semantic-syntactic focus in one creative act of modelling (discarding the workings of their inner thoughts). Not so for novices.

All too often models are created only explicitly employing a semantic-syntactic focus, leading to syntactically correct models in some modelling language which may nevertheless be quite *useless*. We observe that in many modelling efforts (especially in educational and training contexts), the situational purpose of a model is hardly made explicit and modelling is indeed reduced to a syntactic exercise. We thus point out the importance of clear awareness of the pragmatic focus *in addition to* the semantic-syntactic focus for any modelling effort that is meant to be 'useful'. If syntactic exercises are performed separately (and indeed this seems alright *as a limited exercise*), we should also train/guide modellers in developing a proper *pragmatic focus*: train them to produce relevant descriptions (perhaps in natural language at first) that serve specific purposes.

To conclude: in view of breaking down modelling efforts into manageable steps (for novices but also for experts), it seems potentially helpful to separate steps involving pragmatically focused description from steps involving semantic-syntactic focus, and have the former precede the latter. Also, it can be helpful to break down conceptualisation into three further steps: generation, classification, and selection of propositions. In other words, a maximal breakdown combining pragmatic versus semantic-syntactic focus combined with the generation – classification – selection distinction leads to a 2x3 matrix of possible main foci.

The separation in skills that may or may not be required for the application of these possible foci in conceptualisation creates opportunities for nuanced analysis and design of 'mini dialogue games' built around assignments with clear and sufficiently limited focus questions, fit for participants with particular (possibly limited) competencies.

4 Focused Conceptualisations

Based on the theoretical findings and assumptions sketched in the previous sections (that in turn are based on the literature, practical experience, and focused explorative research), we now zoom in on a detailed aspect of the formulation and execution of conversations-for-modelling (viewed as dialogue games): concrete 'mini-games' as parts of larger modelling efforts, as discussed in section 1.

As mentioned, expert modellers typically want cut to the chase and create a conceptual model straight away (though perhaps taking sub-steps 'inside their heads'), but all novice-oriented approaches to modelling we are aware of break up the modelling process in smaller, more workable ('playable') chunks. Novices indeed seem better capable of dealing with small, focused conceptualisation assignments than with immediate and holistic 'diagram drawing', and this is commonly reflected in many of the more developed elicitation and modelling practices.

In our analysis of 'mini dialogue games', that we dubbed *Focused Conceptualisations* or *FoCons*, we are particularly interested in *the questions asked in them*, and *the restrictions on answers to be given in them.* Indeed, the FoCon concept is designed to support thinking about questions and answers. This basic principle should never be forgotten when dealing with focused conceptualization of any kind. Note, however, that questioning and answering are primarily theoretical devices here (deep interface) that may or may not be explicitly present in the surface interface.

In close relation with the identification of questions and answers (explicitly stated or not), we can then for specific situations consider *who is or is not capable of understanding particular questions and provide acceptable (sufficient quality) answers to them*. This is a second analytical use of FoCons: providing more detailed insight in what is asked of modellers in actual situations, and whether this is realistic in view of their actual competencies and expertise [18].

Essentially, a Focused Conceptualisation is a *description of a communication situation* in which one or more participants engage in a focused conversation in order to arrive at a specific conversational goal, typically some sort of highly structured abstract description, specification, or (partial) model.

FoCons can in principle be used both descriptively and prescriptively; they can be used in the analysis of observed model-oriented conversation, but also in method design, *framing* such conversations in order to guide them (the frame being a dialogue game). Once a FoCon (or set of FoCons) is sufficiently clear, it can be used as a basis for designing one or more deep level dialogue games, which in turn may serve as a basis for a Surface Interface design. In this paper, however, the main emphasis lies on the FoCon as an analytical instrument. We will report on FoCon-based dialogue games elsewhere.

4.1 The Content of FoCons

A modelling effort can in principle be covered by a single (rather large) FoCon[1], but as should be clear by now we are mostly interested in the use of FoCons as a means to decompose a modelling effort in phases/steps/activities with a more constrained focus than the modelling effort as a whole. Previous research suggests that even unguided, real life modelling exercises nature often consist of such focused sub-activities [3]; often this even holds expert modellers.

In describing a FoCon, we distinguish between the following main aspects of a communication situation (Table 1):

Table 1. Information categories that make up a FoCon description

Short description	**Clarification**
What may or must ‘go in’	Types of information; sources thereof
What should ‘come out’	Pragmatic as well as semantic/syntactic constraints on textual and other results (including social results), preferably phrased as ‘modelling (sub)goals’ (see section 2)
The type(s) of abstraction activity involved	Generation, classification or selection (separate, or integrated in one step)
The specific focus questions asked (literally so)	This concerns both the pragmatic and the semantic-syntactic focus)
The (types of) participants involved and their relevant competencies and expertise	Possibly, flaws therein; possibly, other relevant information on the participants
Instructions given and/or procedures, conventions and guidelines	Explicit and also implicit if relevant; adhered to by the participants (‘rules of the game’)
Further situational aspects or constraints	E.g. media involved, resources required, organisational issues, social issues, political issues, or whatever is deemed relevant.

4.2 Examples of the Use of FoCons in Case Analysis

We will illustrate the use of FoCons as an analytical device at the hand of the three examples.

[1] Note that though strictly speaking a FoCon is a *description* of a communication situation, we allow ourselves to refer to both specific communication situations and situation *types* as “FoCons”.

The first example concerns a relatively unguided modelling assignment in which three participants (students) were asked to create an as-is process model of a well-documented domain. The modelling language was not dictated, but the students chose to use UML activity diagrams. As discussed in [3], the students' modelling behaviour clearly showed division of their 18 minute session in 3 phases: i. choosing the modelling language and sub-division of work, ii. exploring and deciding which actors play a role in the first partial process model, and iii. modelling the sub-process. The second of these phases is a good example of an (improvised) FoCon. The participants first needed to establish which of a fair number of actors (roles) in the domain were relevant for a particular sub-process (swim lane diagram) they set out to create. Thus they focused on a sub-set of questions of those underlying the whole effort assigned to them:

- They focused on the selection of relevant *actors* only
- They restricted themselves to only one sub-process (out of five given in the domain description)

At the hand of the FoCon analytical categories listed earlier in this section (Table 1.), we will consider this communication situation in more detail (note that this is a marked extension of the initial analysis reported in [3]).

- *What may or must 'go in'*: The participants were provided with an existing list of roles, hence they merely had to select relevant roles. The problem lay in understanding the roles and how they related to activities in the process. Hence, important input was also the list of activities as understood to be part of the sub-process under consideration (with the domain description as a source). It served to create (implicitly!) a 'table' of relevant activities and actors involved. Ordering of activities was not relevant within this particular FoCon. The textual source that was provided proved to be insufficient: after a failed attempt to solve the puzzle based on documentation alone, additional information was obtained by actively involving the person (researcher) who drew up the textual documentation.
- *What should 'come out'*: For this FoCon, the participants really only wanted to identify the actors (in order to set a 'swim lane' for each of them). The activity list was a by-product that would nevertheless be useful later on. The pragmatic focus was clear: "describe who is involved in the sub process as described in the case documentation". The matching semantic-syntactic focus was also clear: just an unordered, agreed list of actors (a basic concept in UML activity diagrams).
- The *type(s) of abstraction activity (generation, classification, selection).* Generation and classification of the main concepts involved, *actors*, was not necessary since they were given, leaving selection. However, in order to do this the activities involved needed to become clear as well as, more importantly, the activities which specific actors were involved with. Obtaining this information did involve some active content generation in the end: they asked an expert. Classification never was a real issue, so selection was what this FoCon was mostly about: setting up a clear-cut way of reasoning about determining the actors involved, and gathering the facts needed to carry through such reasoning.

- The *specific focus questions* asked. Though the focus questions were not all asked explicitly, they could be reconstructed as follows:
    - MAIN Which actors are involved in the sub-process concerned?
        - SUB1 Which activities are part of the sub-process?
        - SUB2 Which activities involve shortlisted actors?
        - SUB3 Which actors are involved in the activities that are part of the sub-process?
- The (types of) *participants involved*. The participants did not have enough information (until they asked an external source), but they did have sufficient basic skill to execute the FoCon. Importantly, they also had the skill to conceive and enter the FoCon as a sub-step in solving their problem: after all, the FoCon as a sub-game in the modelling effort was not dictated, but initiated by the modellers themselves as a strategic step of their own devising.
- *Instructions, procedures, conventions and guidelines*. Being of their own making, the FoCon largely concerns "rules *in* the game" [2,3]: goal rules set by the players themselves (chosen strategy) as a step in solve a larger problem. Conventions were mostly those of general constructive conversation.
- *Further situational aspects or constraints*. The FoCon would probably have been much more effectively executed if its questions and its simple underlying reasoning had been explicit from the start.

We trust the example above sufficiently illustrates the sort of information contained in a FoCon as an instrument of analysis. Also, we hope the reader will see that it immediately sparks ideas about possible use in method design. In the other two examples below, we will not repeat the point-by-point representation followed above for reasons of space.

The second example concerns analysis of an existing and well developed procedure for Group Model Building (System Dynamics) in context of reshaping it as a guided dialogue game.

Group Model Building (GMB) is a method for creating causal models in the context of 'problem structuring', based on input given by an (often heterogeneous) group of people. GMB is typically used as an instrument in performing interventions in (unwanted) situations: a method in management science. Main concepts used in constructing the most basic diagrams in GMB ('Causal Loop Diagrams') are the *problem variable* (some quantifiable indication of a problem, like "teenage pregnancies (too high)" or "number of cars produced (too low)"; *causes* (changes of which influence the problem variable, like "money spent on education" or "number of people employed"); *consequences* (effects of changes in the problem variable, like "rising unemployment" or "dropping customer satisfaction"). Crucial are also *causal* or *feedback loops*, circularly linking some causes and consequences, that capture the dynamics of the system (problem situation) described. GMB has been used for well over a decade, and has a well developed practice [19].

We analysed the existing 'script' (as GMB practitioners call it [20]) used by facilitators of the group model building process. The script contained explicit 'steps', but these did not fully match the five FoCons identified by us (more steps were distinguished in the script, based on instructional structuring rather than systematic process analysis). Though the script was quite workable as an informal guideline, it had been somewhat unsystematically drawn up. Also, it was incomplete with respect to explicit statements concerning specific questions asked and decision criteria used (i.e. the rules of the game). Our initial analysis of the script was used as a basis for extensive questioning (two two-hour interviews) of a GMB expert. It soon became clear that this person could answer all our questions, i.e. indeed the script was lacking detailed and well-structured information but the GMB expert did have detailed knowledge of how to do the job. The FoCon structure and Dialogue Game metaphor (which was first explained to the expert) proved highly valuable as a scaffold for the interviews, which provided not only us, but also the GMB expert with a very satisfactory increase in insight in the structure and rules of the GMB operational procedure.

After the in-depth analysis, the five FoCons now defined were successfully used as a basis for the design of a playable dialog game implemented in InterLoc (see fig. 2). We plan to report on this game elsewhere. We briefly list the GMB basic FoCons below:

1. Create individual lists of ‘ideas’ and ‘variables’ with respect to factors influencing and/or being influenced by the (stated) ‘problem variable’
2. Create a list of ‘ideas’ and ‘variables’ understood by and agreed on by the group (‘group list’)
3. Decide (and discuss) which of the variables listed is a ‘cause’ variable
4. Decide (and discuss) which of the variables listed is a ‘consequence’ variable
5. Check for any causal loops found whether they are understood by all participants, and deemed relevant (loops are identified through a deterministic ‘game procedure’ (like running the bank in Monopoly), so they only need to be pointed out, understood, and proclaimed relevant)

FoCons 1. and 2. emphasise focused generation of ‘ideas’ and distillation thereof in ‘variables’. In fact, one could say 1. and 2. both consist of two sub-FoCons, the first (‘idea’ generation) enforcing only a pragmatic focus, the second (‘variable’ generation) enforcing a more constrained pragmatic focus as well as a semantic-syntactic focus. This latter sub-step implies a mild form of classification and selection, and thus combines generation-classification-selection in one step. FoCons 3. and 4. are aimed not at generation but strictly at classification and selection: of ‘causes’ and ‘consequences’, respectively. The pragmatic and semantic-syntactic foci neatly fit together here. FoCon 5. does not require conceptualisation as such (except by the facilitator or ‘game master’), but only an understandability check and validation of content derived from other findings.

The step-by-step succession of focus questions thus creates a series of tasks which most educated novice modellers find quite feasible, as has been proven in many GMB application cases (GMB typically involves only novices except the facilitator).

This is only a basic example of a GMB dialogue game. We intend to later expand our analysis into the more technical, formally more challenging aspects of GMB and system dynamics modelling, which have so fare remained outside our scope.

The third example concerns analysis and critical discussion of a number of problematic model-related communication situations that occurred in the operational process of a large Knowledge Based Systems company; only one situation will be presented here.

The situation in question concerned elicitation of some legal decision criteria (declarative in nature) that had to be obtained, clarified, and reworked into the description of a stepwise decision process. Time pressure was great, and the one expert involved was not available for a personal interview. Instead, some telephone conversations had to do the job. Not surprisingly, this did not turn out too well.

We employed the FoCon template and Dialogue Game metaphor in systematically discussing the situation with the analyst who performed the sessions with the domain expert. This approach first served to create a new awareness with the analyst that such situations could be discussed as isolated 'communication games' and become subject to systematic evaluation and analysis. We quickly found that this particular FoCon suffered from a number of flaws, some of them obvious, some less so:

- The domain expert was not 100% knowledgeable about the legal details, even though he was an authority in the legal process
- The existing legal decision criteria were not phrased in a way directly fit to be transformed into 'decision steps'. This concerns a clash of the pragmatic focus of the existing documentation with that of the pragmatic and semantic-syntactic focus of the FoCon, making necessary the active generation of conceptual content.
- The domain expert was not able to rephrase the declarative decision criteria as a decision procedure (i.e. generate the required content), thus leaving this to the analyst, who had insufficient legal knowledge. Thus, the focus questions could not be answered without the analyst 'making up a decision process'.
- The medium used for this FoCon (telephone) was seriously sub-optimal for the task at hand
- Time pressure led to a cut-off of the elicitation process before validation of the results was complete (validation goal not achieved)

The example neatly demonstrates one particular use of FoCons: to determine whether a 'player' is fit for playing the game in the first place. Whether the flawed FoCon is to be blamed on the player (the 'novice modeller' domain expert) or the game (the questions asked, set by the analyst) is not our concern here, but in the example the deployment of the FoCon as an instrument enabled efficient, to-the-point and instructive analysis of the communication situation and provided valuable lessons learned as well as useful guidelines for the staging of similar situations in the future.

5 Conclusions and Future Work

In section 1., our main questions for this paper were put as follows: "how can we understand and frame the questioning-and-answering that constitutes conversations for modelling, and as an extension of this, how can we usefully analyse concrete model-oriented communication situations, in particular in view of their accessibility to novice modellers". While certainly much more could be said in answer to these questions, we have done the following:

We recapitulated previous work in which the activity of modelling is viewed and analysed as a *constrained conversation*, driven and constrained by *rules* (also defining *goals*), consisting of *interactions* and rendering sets of propositions constituting *models*. This makes it possible to view, and also shape, conversations-for-modelling as *dialogue games*.

Abstract conceptualisation being a core activity in modelling, we distinguished three generic types of conceptualisation task: *generation*, *classification*, and *selection*. While such logically divided goals may be catered for in a single task, they can also be separated, making the task easier (and thereby more accessible for novice modellers, at the least).

Noting that *focus questions* play a central role in guiding a dialogue game for modelling, we distinguished two essential and complementary types of focus: *pragmatic* and *semantic/syntactic*. Again, though both foci can be combined in one task, it is possible to separate tasks and take on once focus at the time, or to otherwise work with the separate foci to decompose the main task and create easier sub-tasks.

Finally, we introduced the analytical instrument of the Focused Conceptualisation or FoCon, a template that helps identify and analyse *communication situations* along the lines of the framework sketched in the other sections. A specific use of FoCons is to help establish whether certain players are up to the task set for them in terms of their conceptualisation skills and expertise. We illustrated the FoCon concept at the hand of three rather different cases in which FoCons were used.

Indeed we believe this paper represents a step forward in understanding at a micro level ('operational' level) how modelling as an activity works, and how we may deal with making it more accessible and structured. We have identified and demonstrated a number of concepts that are helpful in thinking about generic (types of) questions asked and answers given in modelling sessions, and how a decomposed conversation-for-modelling can be analysed and, possibly, constructed. We already experienced that the foci and other concepts introduced help us substantially in understanding and describing specific modelling situations, either observed ones or future ones.

Future research will focus on refining and expanding the framework, and applying it to more complex and diverse types of modelling. We will increasingly aim beyond analysis: for the design, implementation and testing of dialogue games, hoping to tackle increasingly challenging (and more formal) modelling tasks and making them accessible to a novice audience. However, experts may also benefit from our findings; in general we hope to provide a sound set of principles and patterns for the situational design of operational modelling methods as interactive systems. As we learn more about 'deep interface' operations and requirements, we intend to gradually shift our focus to 'surface interface' issues.

References

1. Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.v.d.: Formal Modelling as a Grounded Conversation. In M. Goldkuhl, G Lind and S. Haraldson (eds.): Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP05), Kiruna, Sweden. Linkopings Universitet and Hogskolan I Boras, Linkoping, Sweden, EU, pp. 139-155 (2005)
2. Hoppenbrouwers, S.J.B.A., Weigand, H., Rouwette, E.A.J.A.: Setting Rules of Play for Collaborative Modelling. In: Kock, N., Rittgen, P. (eds.) International Journal of e-Collaboration (IJeC), vol. 5(4), pp. 7–52. IGI Publishing, USA (2009); Special Issue on Collaborative Business Information System Development

3. Ssebuggwawo, D., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Interactions, Goals and Rules in a Collaborative Modelling Session. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 54–68. Springer, Heidelberg (2009)
4. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., Roelefs, J.: Concepts and Strategies for Quality of Modeling. In: Halpin, T.A., Krogstie, J., Proper, H.A. (eds.) Innovations in Information Systems Modeling, ch. 9, IGI Publishing, Hershey (2008)
5. Hoppenbrouwers, S.J.B.A., van Bommel, P., Jarvinen, A.: Method Engineering as Game Design-An Emerging HCO Perspective on Methods and CASE Tools. In: Halpin, T., et al. (eds.) Workshop Proceedings of EMMSAD 2008: Exploring Modeling Methods for Systems Analysis and Design affiliated to CAiSE 2008, Montpellier, France, pp. 97–111 (2008)
6. Leahy, W.M.: Cognitive Load Theory and Instructional Design: An Outline of the Theory and Reflections on a Need for New Directions to Cater for Individual Differences and Motivation. In: Larson, J.E. (ed.) Educational Psychology: Cognition and Learning, Individual Differences and Motivation, ch. 8, Nova Science Publishers, New York (2009)
7. Bransford, J.D., Brown, A.L., Cocking, R.R. (eds.): How People Learn: Brain, Mind, Experience and School; expanded edn. National Academy Press, Washington (2000)
8. Taylor, J.R.: Rethinking the theory of organizational communication: how to read an organisation. Ablex Publishing, New Jersey (1993)
9. van Eemeren, F.H., Grootendorst, R.: A systematic theory of argumentation: The pragma-dialectical approach. Cambridge University Press, Cambridge (2004)
10. van Eemeren, F.H., Grootendorst, R., Snoeck Henkemans, F., Blair, J.A., Johnson, R.H., Krabbe, E.C.W., Plantin, C., Walton, D.N., Willard, C.A., Woods, J., Zarefsky, D.: Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments. Lawrence Erlbaum Associates, New Jersey (1996)
11. Ravenscroft, A., McAlister, S.: Designing interaction as a dialogue game: Linking social and conceptual dimensions of the learning process. In: Juwah, C. (ed.) Interactions in Online Education: implications for theory and practice, ch. 4, pp. 73–90. Routledge, New York (2006), `http://www.interloc.org/pubs/InterDG_AR&SRM.pdf`
12. Cañas, A.J., Novak, J.D., González, F.M. (eds.): Proceedings of the First International Conference on Concept Mapping, Concept Maps: Theory, Methodology, Technology, Pamplona, Spain (2004)
13. Novak, J.D., Cañas, A.J.: The Theory Underlying Concept Maps and How to Construct and Use Them. Technical Report, Florida Institute for Human and Machine Cognition, IHMC (2008)
14. Wilmont, I., Brinkkemper, S., van de Weerd, I., Hoppenbrouwers, S.J.B.A.: Exploring Intuitive Modelling Behaviour. In: EMMSAD 2010. LNBIP, vol. 50, pp. 301–313. Springer, Berlin (2010)
15. Colburn, T., Shute, G.: Abstraction in computer science. Minds and Machines 17(2), 169–184 (2007)
16. Kolfschoten, G.L.: Theoretical Foundations for Collaboration Engineering. PhD Thesis, Delft University of Technology (2007)
17. Lentz, L., Pander Maat, H.: Functional Analysis for Document Design. Technical Communication 51(3), 387–398 (2004)
18. Frederiks, P.J.M., van der Weide, T.P.: Information modeling: the process and the required competencies of its participants. Data and Knowledge Engineering 58(1), 4–20 (2006)
19. Rouwette, E.A.J.A., Vennix, J.A.M.: System dynamics and organizational interventions. Systems Research and Behavioral Science 23(4), 451–466 (2006)
20. Andersen, D.F., Richardson, G.P.: Scripts for Group Model Building. System Dynamics Review 13(2), 107–129 (1997)

Towards a Unified Business Strategy Language: A Meta-model of Strategy Maps

Constantinos Giannoulis[1], Michael Petit[2], and Jelena Zdravkovic[1]

[1] Department of Computer and Systems Sciences
Stockholm University
Forum 100, SE-164 40 Kista, Sweden
[2] PReCISE Research Center, Computer Science Department
University of Namur
Rue Grandgagnage 21, B-5000 Namur, Belgium
{constantinos,jelenaz}@dsv.su.se,
mpe@info.fundp.ac.be

Abstract. Alignment between business strategies and the resources engaged ensuring their realization, has been a continuous concern of enterprises of all kinds in last few decades. Commonly, enterprises fail to establish the traceability from business strategies towards operational tasks carried by employees. From the requirements engineering perspective this problem leads also to a misalignment between business and IT assets. In this study, we argue that for communicating high-level intentions and strategies down to the operational perspective, i.e. tasks and resources, the core necessity is to have a rich and well-defined language for modeling business strategies. Such a language could be further utilized for facilitating formalizations and a constructive analysis of high-level business aspects of enterprises, as well for comparing and unifying existing intentional modeling languages from the business and requirements engineering domains. As a reference proposal for formalizing business strategies, we consider the well-established *strategy maps* [1] from the Management Information Systems community which provide textual concepts of strategy-related notions establishing causal relationships between them. We have set an effort to formalize strategy maps in the form of a meta-model, usage scenarios and constraints, providing a systematic basis for obtaining a unified language/ontology for business strategy modeling.

Keywords: Business strategy, strategy maps, goal modeling, meta-model.

1 Introduction

New business environments foster enterprises to streamline their organizational structures, activities and resources in a way to align them with long-term business visions and strategies.

In this effort, the major concern is to enable communicating high level strategies from executives toward employees, and from product and service offers toward concrete tasks [2]. A proposal concerning notions pertinent to capturing business

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 205–216, 2010.

strategies and their realizations comes from the work on *strategy maps* by Kaplan and Norton [1]. Briefly, the maps represent strategic blueprints of enterprises observed from several perspectives, i.e. from business visions and product offers down to internal capabilities and assets.

In the “modeling” community, business intentions and following strategies are typically conceptualized using goal-based languages, such as in Goal-Oriented Requirements Engineering (GORE) [3] with i* [4], KAOS [3], etc. as well as others coming from the business domain, such as the Business Motivation Model, BMM [5].

The major question in the previously described context concerns the capabilities of goal-oriented languages to capture end express the perspectives, the notions and their relations as mandated by strategy maps. As formal and as precise they are, being generic makes questionable their representational capabilities for business strategies as opposed to dedicated schemes such as strategy maps which provide rich expressiveness.

We aim to analyze, compare and unify goal modeling languages from the requirements engineering domain, which support formality, preciseness and are generic enough to capture any goal related aspect, with those from the business domain, which are rich in expressiveness and focus on business strategies. The overall objective is to leverage from both domains and obtain a rich and expressive language capable of capturing intentions and strategies from visionary to operational enterprise perspectives supporting mappings to current schemes practiced.

In order to define this rich language, we will use a systematic approach in which existing languages for modeling intentions are analyzed and evaluated with respect to their suitability to represent aspects relevant to business strategy formulations. Our approach will be based on ideas developed in the Unified Enterprise Modeling Language (UEML) approach whose objective is to create a framework for interoperability of enterprise modeling languages and to define “a core language for enterprise and IS modeling” [6]. The UEML approach has analyzed a number of existing modeling languages by mapping their constructs onto a common and well defined ontological base. By doing so, the understanding of concepts of existing languages was improved and it was possible to progressively grow a larger ontology containing well connected concepts by extending the ontological base.

For our research effort we will apply a similar approach with the aim to define a core language for business strategy modeling, by analyzing one by one existing languages allowing the representation of business goals cited earlier (BMM, i*, KAOS, etc.). For our starting point in this direction, we have to choose an ontological base that will serve as the basis for analysis of each language. In our case, we have chosen strategy maps as a starting ontological base because it is a widely used method to represent business strategies. Gradually, during the coming steps of our work we will examine other approaches that represent business strategies. Along with the gradual evolution of our ontological base we will be incrementally building relevant mappings of our core language to existing goal-based languages to provide applicability for real practice.

The first step in this research, which forms the scope of this paper, requires the conceptualization of strategy maps, as well as a clarification of possible usage scenarios, to provide a well-formed and systematic basis for further analysis of goal-modeling languages.

The paper is structured as follows: Section 2 presents the related work on goal modeling languages, business strategy and alignment. Section 3 presents our contribution, starting from the strategy map template to its conceptualization to a meta-model. In Section 4, the conclusions and directions for further research are briefly discussed.

2 Related Work

2.1 Goal Modeling

Intentions and consecutive strategies are typically conceptualized using goal-based languages stemming from the requirements engineering domain as well as from the business domain. Languages in the former domain focus on formality and preciseness of systems to be developed, business goals, processes, etc. supporting organizational objectives, though significant variations exist. The latter focuses on expressiveness of business models for supporting business plans.

2.1.1 Requirements Engineering Domain

Within requirements engineering, the work of Kavakli and Loucopoulos [7] provide an evaluation-oriented overview of the most prominent GORE methods. Methods are grouped according to their role in requirements engineering along the four phases in the RE process: elicitation, negotiation, specification and validation. Additionally, the methods are distinguished in terms of formality as semi-formal and formal. Semi-formal methods are imprecise and relations between elements are loosely defined, while they allow stakeholders to establish a shared view of the situation and set an agreeable frame around further analysis to be performed (e.g. GSN [8], GQM [9]). Formal methods lack on goal elicitation, simplicity and flexibility, while they are unambiguous, they support consistency as well as precise representation of goals and have strong semantics (e.g. i*, KAOS). Regardless of grouping, neither based on role nor on level of formality, none of the evaluated methods addresses adequately the complexity of goal analysis. Within an enterprise context, goal analysis relies primarily on stakeholders' interests making their involvement in the modeling process essential. Additionally, a lack of the methods and techniques to identify potential stakeholders, as well as to facilitate their cooperation is also acknowledged.

A more recent evaluation within GORE by Amyot et al [10] describes goal models using the Goal-oriented Requirement Language (GRL), which is part of the User Requirements Notation (URN). URN is a new Recommendation of the International Telecommunications Union, provides the first standard goal-oriented language. GRL integrates concepts of i* and the NFR Framework [11] along with a scenario notation included constitutes GRL, which is scalable and consistent in terms of representation of different view for the same goal model, thus supporting the views of various stakeholders. Additionally, GRL, based on the i* language, can capture the high-level business goals and the non-functional requirements for a stakeholder as well as possible alternatives for the achievement, and therefore, allowing the evaluation of trade-off analysis between alternatives.

2.1.2 Business Domain

Goal-based languages from the business domain are focused on the expressiveness of business model and their relation to business plans.

The Business Motivation Model (BMM) is a proposed standard of the Object Management Group (OMG) - it is focused on business plans, including their motivating factors, elements and their interrelations. The main parts of the BMM are the Ends, the Means and the Influencers. The Ends capture enterprise goals and objectives; the Means captures enterprise strategies, tactics, rules and policies, which together represent what is an enterprise aiming at achieving. The Influencers capture the elements of business plans as well as assessments on their impact towards Means and Ends, which justifies the existence of each element of a business plan.

Another goal modeling approach is the formal framework of Popova and Sharpanskykh [12] which uses performance indicators to monitor the progress of an enterprise. Essential principle of this approach is the fact that "each organization exists for the achievement of one or more goals". Goals are clearly related to the performance indicators of an organization and a clear connection between them is established through goal structures. The approach allows goal classifications and different degrees of goal satisfaction but enforces a formal and unified representation of goal hierarchies, which makes them dependent to the modeler's perspective.

2.2 Business Strategy

Strategic planning is the process during which a strategy is defined by analyzing the current status of the enterprise and the competitive environment in which it resides. As pointed out in [13], citing Barney [14], three main types of approaches focusing on different aspects have been proposed. The "resource-based view" argues that the competitive advantage of a firm depends mainly on its distinctive unique capabilities, provided by its resources.

The "industrial organization perspective" (exemplified by the work of Porter) argues that the competitive advantage will result from a clear positioning of the firm with respect to its environment, described by the structure of the industry based e.g. on the five forces model of Porter. The positioning requires the choice of a strategic line among three possible ones (cost leadership, differentiation or focus) that will be implemented by carefully considering the required capabilities of internal processes of the value chain in terms of cost and quality.

The last perspective, the "Schumpeterian view of competition" centers on competitive innovation and rests on the idea that radical innovations can disrupt the industrial environment in which the firm operates, thereby giving opportunities to take an advantage over companies who's capability to innovate is lower.

A good strategic planning process commonly takes into account these three different but complementary views on the subject.

Strategy Maps and Balance Scorecards [15] have been proposed by Kaplan and Norton as a means to represent the strategy of a company in order to be able to communicate and to monitor the achievement of the strategic objectives. A strategy map serves as a mediator between the Mission, core Values, the Vision and the Strategy of an enterprise to the work performed. It is one of the rare frameworks providing means for visual representation of a strategy. Therefore, for all outlined features, we give it a particular importance in our work.

2.3 Alignment

Several research works have proposed combining several languages in order to address the issue of aligning strategy and value proposition with related operational processes.

Within [16], i* is used for the representation of strategic goals, the value proposition model is described by business schematics and Role-Activity-Diagrams (RAD) describe processes.

The INSTAL method proposes a map-type to represent strategic intents and complementary process formalisms for business processes, integrated in a single meta-model [17]. Similarly, [18] puts forward a map-driven process modeling approach based on intentions and strategies capable of abstracting organizational goals and their achievement from detailed business processes.

In [19] and [20], the combination of i* for strategic goals modeling and e3value for value proposition description is proposed.

In [21], the combination of several goals languages (i*, BMM, KAOS) and value modeling ontology (e3value, REA, etc.) is analyzed and detailed rules for combination are proposed.

The 3g framework for business-IT alignment proposed in [22] is based on multidisciplinary goal concepts and focuses on linking task goals to strategic business goals.

The study presented in [23] is close to ours using strategy maps and requirements engineering approaches for alignment, which are demonstrated through a mapping of constructs between i* and strategy maps.

3 Modeling and Utilizing Strategy Maps

The content of this section includes our effort on conceptualizing strategy maps into a well-defined meta-model. For the development of the meta-model we have conducted a comprehensive literature study on strategy maps [15] [24] [25], as well as on published outcomes of their implementation [1] [15] [24] [26] [27]. Given the limited formalisms of the strategy map proposal and the ambiguities found in numerous textual descriptions, we have considered and analyzed them all, to obtain a precise and complete formalization. The first part of this section gives an overview of strategy maps and an analysis of identified usage distinctions, while the second part presents the developed meta-model.

3.1 Strategy Maps (SM)

A strategy map is a general representation of the four organizational perspectives of the Balance Scorecard framework (BSC) [15]. According to [26], a BSC presents an organization’s business activities through a number of measures typically from four organizational perspectives: financial, customer, internal, learning and growth, and provides a language to communicate priorities within an enterprise. A scorecard is considered balanced (BSC) due to the four perspectives that provide complete coverage of business processes, the time aspect covered in a bottom up manner suggesting that what lies on the bottom is the outcome of planning at the top and has taken place in the past. Additionally a scorecard is also considered balanced because it covers both the internal as well as the external aspects of an enterprise. Finally, a BSC is structured based on cause-effect links/assumptions who's monitoring and assessment is essential for identifying interdependencies across an enterprise.

A strategy map is a comprehensive representation of strategy that captures linkages of the four BSC perspectives in a cause-effect manner and facilitates executives in communicating direction and priorities across the enterprise on all levels, thus creating enterprise alignment. Kaplan & Norton have proposed a template for strategy maps (figure below) representing how an organization can create value [24].

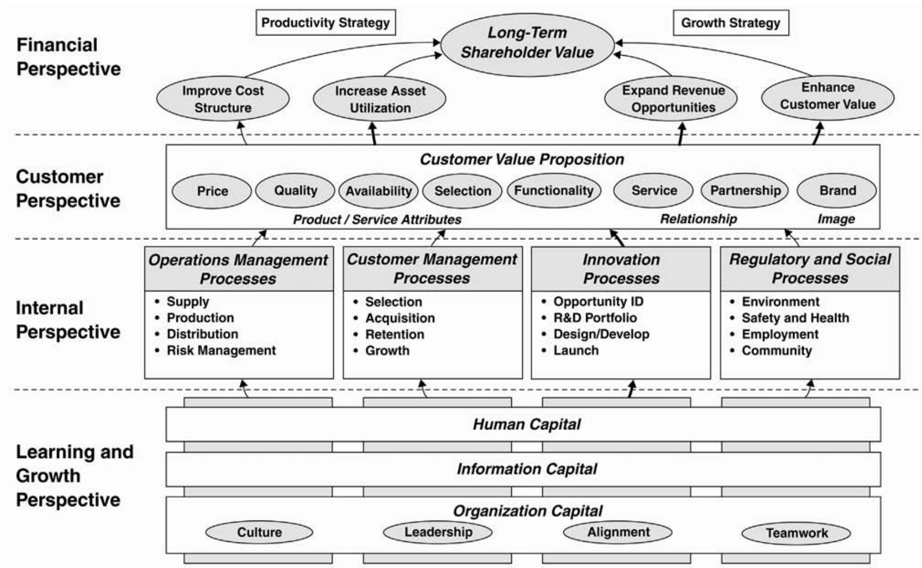

Fig. 1. The Strategy Map template [1] [15]

According to [24], a strategy map is based on five principles:

- Strategy balances long-term financial commitments aiming at profitable revenue growth and short-term financial commitments aiming at cost reductions and productivity improvements.
- Strategy is based on differentiated and clearly articulated customer value proposition.
- Value is created through focused, effective and aligned internal business processes grouped into four clusters: operations management, customer management, innovation and regulatory and social.
- Strategy consists of simultaneous, complementary themes highlighting the most critical processes supporting the customer value proposition.
- Strategic alignment determines the value and role of intangible assets: human, information, organization.

Strategy maps usage is typically aiming to establish a strategy to be achieved in the future; however, they may also be used for analyzing the status of a current business strategy striving for completeness. We classify the former as the *to-be* usage (i.e. perspective), while the latter as *as-is*.

The *to-be* usage refers to a map representing a strategy for the future of an organization. It could be a strategy map for a newly established organization or a refinement of an existing strategy map being the result of having incorporated the BSC framework with specific objectives, measures and targets and results from the prior period have shown changes that need to be made. The best way to build a strategy map is to follow a top down manner, as suggested by [25], starting from a mission statement and core values to develop a strategic vision, which should project the organization's overall goal. Consequently, according to the template, all four perspectives need to be explored from the financial to the customer, to the internal to the learning and growth. In this category of strategy maps completeness is taken for granted because for an existing strategy map, completeness is essential, as no incomplete strategy map may exist. For a newly established strategy map *completeness* must be achieved, because the BSC framework cannot be incorporated into an incomplete strategy map and consequently any relevant action plan, as well as the fact that an incomplete strategy map is not a considered as a strategy map.

The *as-is* usage refers to a map representing a current strategy of an organization. Such usage serves analysis purposes of current goals by migrating to strategy maps. An *as-is* map can be used to identify missing goals, processes and assets but may also be considered as an intermediate state for an existing organization that regards to adopt strategy maps for the first time. Such effort will require a few iterations taken in the same top-down manner described above. Such analysis purposes allow a strategy map to be in an incomplete state requiring further improvement, thus not being applicable to the BSC framework and any relevant action plan.

In the strategy map meta-model that follows, we have considered capturing both the *as-is* and the *to-be* usage.

3.2 The Strategy Map Meta-model

According to [28], also echoed by [29], a meta-model defines the conceptual elements of a language as well as their possible interrelations. Contextual conditions, or static semantics, are constraints which we have defined through inference from the strategy map template and implementations found in the studied literature. For each language construct, constraints define the interrelations allowed and restrictions imposed.

The strategy map meta-model is presented in the figure below. The strategy map template [1] [15] constitutes the basis for the presented interrelating concepts, while additional conceptual variations of strategy map applications are embodied through constraints.

We present classes and interrelations of the meta-model by referring to the strategy map template and the relevant constraints by referring to applications that introduce conceptual variations.

The meta-model consists of 6 classes: the Strategy-map class, the Group class, the Perspective class, the Theme class, the Goal class and the Causality-relation class. Additionally we have introduced cardinality constraints for the relations between classes as we could infer them from the strategy map template and strategy map implementations found in the studied literature.

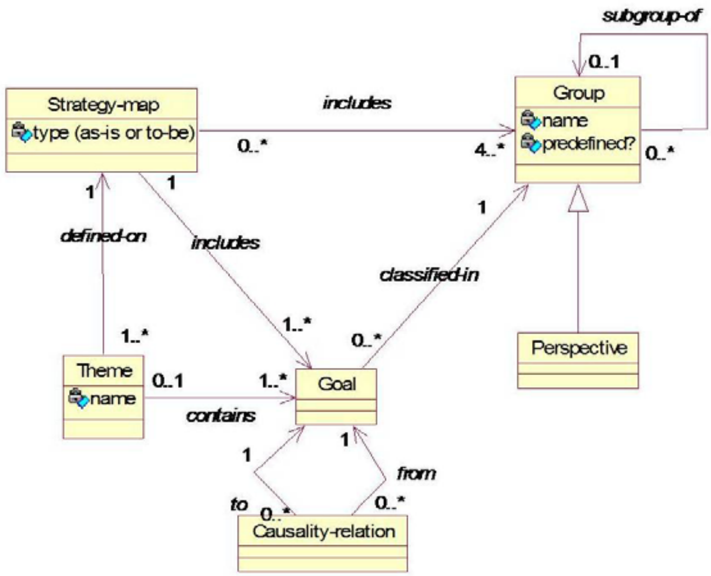

Fig. 2. The Strategy Map Meta-model

3.2.1 Classes

The Strategy-map class refers to the whole strategy map and holds a type attribute, which includes two possible values, either "as-is" or "to-be". "As-is" refers to a state where a strategy map is built to capture the current strategy of an enterprise for analysis purposes implying the possibility that the map is in an incomplete state requiring further improvement. "To-be" refers to a map representing a strategy for the future of the company. We here make the hypothesis that such a map must be complete and ready to incorporate the BSC framework with specific objectives, measures and targets. Our hypothesis is based on the fact that strategy maps are rooted on balanced scorecards (BSC), therefore, "balance" requires completeness. The distinction is made through the type attribute to distinguish between strategy maps built for analysis and strategy maps built for strategy.

The Goal class refers to goals set throughout the strategy map. Goals[1] are defined and grouped within the financial and customer perspectives considering both the strategy map template as well as other applications of strategy maps [26]. For the internal perspective, as well as the learning and growth perspective we consider that both processes and capital appear in a strategy map in the form of goals. A process is executed to satisfy a goal [18] which though the cause-effect relation within a strategy map supports the goals at the customer perspective. In the same token, goals are set for all groups of capital referring to particular assets (in this context we imply goals

[1] The terms goal and objective are interchangeably used even within the sources referring to the same concept, a desired future state. However, given the fact that for the BSC framework objectives are measurable goals based upon which targets are defined, we use only the term goal within strategy maps, thus, the Goal class. Objectives are used when the balanced scoreboard is applied to a strategy map.

on assets) aiming at desired competencies, capabilities needed to support, through the cause-effect relation within a strategy map, the internal processes.

The Causality-relation class refers to the cause-effect relation between goals within a strategy map and results into a tree structure of goals that when completed link goals on all perspectives.

The Group class refers to any grouping and any categorization of goals included in a strategy map (e.g. internal process clusters, capital, etc.) and holds two attributes, a name and a Boolean declaring whether the group is predefined (all groupings coming from the strategy map template are acknowledged as predefined). This class captures all possible groupings, the ones predefined in the strategy map template (e.g. internal processes, groups of capital, groups for the customer value proposition and groups within the financial perspective) while also allowing custom groupings to be introduced.

Sub-groups can be introduced into groups thereby making the nesting of groups inside other groups a tree. For example, "Operations Management Process" shown on the template is a sub-group of the "Internal Perspective" group.

The Perspective class refers to the highest level of grouping within a strategy map and is related to the Group class through generalization. Every strategy map includes the four predefined perspectives stated earlier.

The Theme class refers to the strategic theme(s) chosen within a strategy map. A strategic theme is a vertical slice within a strategy map that consists of a specific set of interrelated objectives. As discussed in [25] strategy is build top down, however, strategic themes are set by executives who identify the few critical processes (internal perspective) important for differentiating the customer value proposition (customer perspective) [24]. The vertical slice is then extended to all perspectives and their related goals are identified through the cause-effect relation.

3.2.2 Constraints

A number of constraints have been introduced to capture the variability of concepts found between the strategy map template and other strategy map applications. The following constraints apply to the meta-model:

- For the Strategy-map class:
    - Every strategy map includes the four predefined perspectives of the strategy map template.
    - If a strategy map is to-be, then every group that it includes and that itself has no subgroups included, must have at least one goal defined (it must be complete in the sense that goals must be defined in all categories).
- For the Group class:
    - If a group is a perspective, then it must not be a sub-group of another group. Perspectives constitute the highest level of grouping.
    - If a group is not a perspective, then it is included in another group. A group that is not a perspective belongs at least to a perspective, or to another group.
    - If a group is not predefined it must be included in at least one strategy map. New groups cannot be introduced unless they are used in a strategy map.

- For the Goal class:
  - Every goal included in a theme is also included in the strategy map for which the theme is defined.
- For the Causality-relation class:
  - A causality relation links two goals included in the same strategy map.

4 Conclusions and Future Work

In this study we have addressed the problem of aligning business strategies with operational aspects of enterprises. We have argued that for solving this problem the core need is to define a rich and well-defined language for modeling business strategies. Such a language can be further used for at least two purposes: (a) to serve as a reference for analyzing and comparing the capabilities of existing intentional modeling languages, both from the requirements engineering domain (i*, KAOS, GRL, etc.) and from the business domain (BMM, etc.), and (b) to facilitate transformations of high-level business notions related to stakeholder intensions and strategies to operational aspects, such as tasks and resources.

After a literature study, we have decided to consider the description of business strategies and the related notions proposed by Kaplan and Norton with strategy maps, as a relevant basis for ontology of business strategy. In this context, we have defined a well-structured and correct conceptualization (i.e. a meta-model) of strategy maps, and we have also elicited possible usages, which frame the scope of this study. We have outlined the meta-model by analyzing and interpreting numerous written sources on strategy maps and their utilizations. Our meta-model has included all major notions of strategy maps, such as the strategy perspectives, the containing groups of goals, as well as their causal relationships. The meta-model also supports the major utilization scenarios of strategy maps: (a) *as-is*, enabling modeling of current strategies for an enterprise aiming at analyzing them from a strategy map's requirements scope, and (b) modeling of *to-be* business strategies by following requirements for obtaining complete and correct strategy maps. Moreover, the meta-model generalizes some elements of the strategy map template such as the notion of group that covers both perspectives and lower level groups, while it introduces explicitly the notion of user defined perspectives and groups. Finally, it clearly separates classes and instances, where instances being contextually dependent, they are not explicitly shown but must be declared for an implementation.

The meta-model is aligned to the strategy map template as well as to the context dependent implementations found in the studied literature, inferring constraints from textually ambiguous descriptions. Therefore, our contribution is yet to be validated through case studies.

Future work concerns the extension of the meta-model to include the "realization" aspects of strategy maps using the BSC framework; consequently such a model will facilitate comprehensive evaluations of the well-known languages for intentional modeling from the requirements engineering and business domains such as i*, KAOS, BMM, etc. with respect to the concepts of business strategies as formalized in this paper on the basis of the strategy maps proposal. We also plan to design a formalization of strategy maps in an ontological form to provide a richer semantic basis when utilizing business strategies to be transformed to lower-level models, such as business process models. Our ontological base will also be gradually extended to facilitate more approaches for representing business strategies and consequently, the resulting model will also facilitate further comprehensive evaluations of goal-based languages.

References

1. Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston (2004)
2. Kaplan, R.S., Norton, D.P.: Mastering the Management System. J. Harvard business review 86, 63–77 (2008)
3. van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: 5th IEEE International Symposium on Requirements Engineering, pp. 249–262. IEEE Press, New York (2001)
4. Yu, E.: Towards Modeling and reasoning Support for Early-Phase Requirements Engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE 1997), pp. 226–235. IEEE Press, Washington (1997)
5. Business Rules Group (BRG): The Business Motivation Model. Group (2007)
6. Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevicius, R., Opdahl, A., Panetto, H., Verdecho, M.J.: The Unified Enterprise Modelling Language – Overview and further work. J. Computers in Industry 61 (2009)
7. Kavakli, E., Loucopoulos, P.: Goal Driven Requirements Engineering: Evaluation of Current Methods. In: 8th CAiSE/IFIP8.1 Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, EMMSAD (2003)
8. Wilson, S.P., Kelly, T.P., McMermid, J.A.: Safety Case Development: Current Practice, Future Prospects. In: 1st ENCRESS/5th CSR Workshop (1995)
9. Basili, V.R., Rombach, D.D.: The TAME Project: Towards Improvement-Oriented Software Environments. J. IEEE Transactions on Software Engineering 14, 758–772 (1988)
10. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating goal models within the goal-oriented requirement language. J. Intelligent Systems 25, 841–877 (2010)
11. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Dordrecht (2000)
12. Popova, V., Sharpanskykh, A.: Formal Modelling of Goals in Organizations. Technical report, VU University faculty of Science (2008)
13. Camponovo, G.: Conceptual Models for Designing Information Systems Supporting the Strategic Analysis of Technology Environments. PhD Thesis. Information Systems Institute, HEC School of Business, University of Lausanne (2006)
14. Barney, J.: Types of Competition and the Theory of Strategy: Toward an Integrative Framework. J. Academy of Management Review 11, 791–800 (1986)
15. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard: translating Strategy into Action. Harvard Business School Press, Boston (1996)
16. Bleistein, S.J., Cox, K., Verner, J.: Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. J. Systems and Software 79, 362–378 (2006)

17. Thevenet, L.H., Salinesi, C.: Aligning IS to organization's strategy: the INSTAL method. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 203–217. Springer, Heidelberg (2007)
18. Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., Rolland, C.: A strategy driven business process modelling approach. J. Business Process Management 11, 628–649 (2005)
19. Gordijn, J., Petit, M., Wieringa, R.: Understanding business strategies of networked value constellations using goal- and value modeling. In: 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 129–138 (2006)
20. van der Raadt, B., Gordijn, J., Yu, E.: Exploring web services ideas from a business value perspective. In: 13th IEEE International Conference on Requirements Engineering (RE 2005), pp. 53–62. IEEE CS, Los Alamitos (2005)
21. Edirisuriya, A.: Design Support for e-Commerce Information Systems using Goal, Business and Process Modelling. PhD Thesis. Department of Computer and Systems Sciences, Stockholm University (2009)
22. Singh, S.N., Woo, C.: Investigating business-IT alignment through multi-disciplinary goal concepts. J. Requirements Engineering 14, 177–207 (2009)
23. Babar, A., Zowghi, D., Chew, E.: Using Goals to Model Strategy Map for Business IT Alignment. In: 5^{th} International Workshop on Business/IT Alignment and Interoperability (BUSITAL 2010), pp. 16–30 (2010)
24. Kaplan, R.S., Norton, D.P.: The strategy map: guide to aligning intangible assets. J. Strategy & Leadership 32, 10–17 (2004)
25. Kaplan, R.S., Norton, D.P.: Having trouble with your strategy? Then map it. J. Harvard Business Review 78, 167–176 (2000)
26. Olve, N.G., Petri, C.J., Roy, J., Roy, S.: Making Scorecards Actionable: Balancing Strategy and Control. John Wiley & Sons Ltd., West Sussex (2003)
27. Cobbold, I., Lawrie, G.: The development of the Balanced Scorecard as a strategic management tool 2GC Conference Paper. In: PMA Conference, pp. 0-9 (2002)
28. Harel, D., Rumpe, B.: Meaningful Modeling: What' s the Semantics of "semantics" ? J. Computer 37, 64–72 (2004)
29. Lucena, M., Santos, E., Silva, C., Alencar, F., Silva, M.J., Castro, J.: Towards a unified metamodel for i*. In: 2nd International Conference on Research Challenges in Information Science (RCIS 2008), pp. 237–246 (2008)

Integration of Interactive, Behavioral and Structural Aspects of Conceptual Models

Remigijus Gustas

Department of Information Systems, Karlstad University, Sweden
Remigijus.Gustas@kau.se

Abstract. Conceptual modeling is an essential part of enterprise engineering activity. Unfortunately, enterprise modeling methods are projecting interactive, behavioral and structural dimensions of conceptualizations into totally different diagram types. If static and dynamic aspects are analyzed separately, they are more difficult to visualize and to understand for stakeholders. Moreover, in the traditional approaches, there is a paradigmatic mismatch among different enterprise architecture modeling dimensions. Analysis of interplay among interactions, state changes and object creation/termination effects is necessary for understanding integrity problems of conceptualizations. The goal of this paper is to present a modeling approach for semantic integration of static and dynamic views of conceptual models. The presented modeling method can be used for separation of crosscutting concerns of computation neutral specifications.

Keywords: Semantic integration of static and dynamic views, separation of crosscutting concerns, behavior, interaction, structural changes of objects.

1 Introduction

Unified Modeling Language (UML) (OMG, 2010) was developed with the ultimate goal for unifying the best features of the graphical modeling languages and creating a de facto industry standard for object-oriented software design. Recently, UML started to evolve into a language for business and enterprise modeling. However, the semantic integration principles of different diagram types are not completely clear in UML. One of the goals of this paper is to present a modeling approach for integration of static and dynamic aspects of conceptualizations.

Explicit modeling of interaction flows is crucial in system development. Understanding of interaction flow sequences among enterprise components is important for system architects to move smoothly from system analysis to design, without requirement to represent a complete solution. Interaction modeling (Gustas & Gustiene, 2009) helps to develop a coherent graphical representation of business process and business data. Interaction modeling in terms of data flows was the strength of structured analysis and design methods (Gane & Sarson, 1979), (Yourdon & Constantine, 1979). UML supports various types of associations between classes, actors and use cases, and between objects such as software or hardware components. However, sequence, activity or use case diagrams are not suitable for modeling explicit interaction flows between actors. If actor interactions cannot be explicitly captured, then they are

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 217–231, 2010.

hidden from business modeling experts. In this case, such important relations cannot be maintained by the conventional CASE tools.

Information system designers often focus on a use case (Jacobson & Ng, 2005) modeling. Use case is a unit of functionality that a system can provide for organizational or technical actors. It can be specified by conceptualizing interactions, activities, and state changes in various classes of objects. A coherent use case represents information system functionality that helps to achieve one goal. Unfortunately, use cases typically define implementation-specific services, which are placed inside technical system boundary. The alignment details of business processes with use cases are often not so easy to visualize and to comprehend.

One of the benefits of service orientation (Gustas & Gustiene, 2008) is to analyze business processes in terms of interaction flows. Declarative nature of service flows is very helpful in system analysis phase, because they have very little to do with dependencies between business activities. The particular strength of interaction dependencies is possibility to capture crosscutting concerns among organizational and technical components. Most of conceptual modeling approaches do not deal with the notion of service flow, which demonstrates value exchange among actors involved in business processes (Gordijn et al., 2000). Information system methodologies are quite weak in integrating data flows with related behavioral effects and representing consequences if commitments in delivering flows are broken. The treatment of such weaknesses would require modification of the UML foundation. Introducing fundamental changes in UML with the purpose of semantic integration of collection of models is a complex research activity. However, such attempts would allow using UML as an enterprise modeling language for developing computation neutral type of diagrams, which are more suitable for reasoning about enterprise architectures. It is recognized that UML support for such task is vague, because the semantic integration principles of different diagram types are lacking (Harel & Rumpe, 2004). In this paper, we present a modeling approach for semantic integration of interactive, structural and behavioral aspects of conceptual models.

2 Conceptual Modeling of Interactive Aspects

Many textbooks in the area of systems analysis and design recommend concentrating first on the structural aspects of domain modeling that are based on specification of classes, attributes and associations. The second step is typically analysis of behavioral aspects of domain, which are expressed in terms of events, state transitions and their related effects. Finally, it is recommended to start modeling of interactive aspects. Interactions are often considered as the third leg of the modeling tripod (Blaha & Rumbaugh, 2005). State modeling can be viewed as reductionist projection of behavior, because both states and interactions are necessary to describe behavior fully. Such way of modeling creates difficulties in the detection of discontinuities and breakdowns in business activities, because the knowledge on the choreography of business interactions is missing.

Many designers see use case diagram as an excellent starting point of system analysis. Parallel, sequential, branching or iterative use case execution can be described by using activity diagrams, but normally this type of specification is not associated with use case diagrams. Use case diagrams are typically not augmented with specification of state related behavior (Glinz, 2000). Many world-class modelers downplay use case diagrams in the early requirement engineering phases. Instead they focus on writing scenarios (Larman, 2009). Another problem is that use case diagram enforces early implementation-related decisions about a technical system boundary and its environment, which is defined in terms of organizational and technical actors. Note that any conceptual representation should follow the basic conceptualization principle (Griethuisen, 1982) in representing only computation neutral aspects that are not influenced by possible implementation solutions. Violation of this principle results in a higher complexity of diagrams.

We believe that analyzing interplay of interactive, behavioral and structural aspects helps to introduce a new approach to conceptual modeling. One of the goals of this paper is to question the conventional way of system analysis and design. We will demonstrate a different way of reasoning in modeling of static and dynamic relations among concepts. A small case study on a conference review management system will be used as a running example, which is important for the demonstration of modeling constructs and their expressive power. Five interactions of a conference review management system are defined by the sequence diagram, which is presented in figure 1.

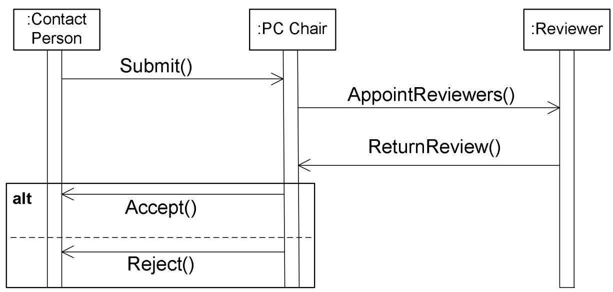

Fig. 1. Main interactions of a conference review management system

This diagram corresponds to our initial description of a conference management system. It is as follows: *One of the authors plays the role of* ***Contact person*** *who* ***submit****s a paper to a conference. The responsibility of a conference program committee* ***(PC) chair*** *is to* ***appoint reviewers*** *for every submission. The* ***Reviewer*** *is obliged to* ***return review*** *of the paper to the* ***PC chair*** *on time. Depending on the reviewing outcome, the* ***PC Chair*** *is authorized to* ***accept*** *or* ***reject*** *a submitted paper. If paper is accepted, then revision instructions are sent to the corresponding* ***Contact person****. Otherwise, reviewer comments are included in the rejection letter.*

There are few unconventional features related to the diagram, which is presented in figure 1:

1) The boxes such as :*Contact Person*, :*PC Chair* and :*Reviewer* are roles, not objects in traditional understanding of object-oriented design. Contact Person, PC Chair and Reviewer are organizational components, which are called actors in UML. Actors are typically placed outside technical system boundary in use case diagram.

2) Designers typically decompose higher level business related actions into more detail activities, use cases or bottom level operations, which fit well one of the UML diagram types. Each presented interaction cannot be placed precisely on a single swimlane of activity diagram, since it binds two responsible actors: agent and recipient. The presented actions *Submit*, *AppointReviewers*, *ReturnReview*, *Accept* and *Reject* are not use cases in the traditional understanding. A use case typically represents functionality, which is performed by the system, to yield an observable result to one particular actor (Jacobson & Ng, 2005). For instance, '*Reviewer is obliged to* ***return review*** *of paper to PC chair'* specifies that *Reviewer* is an agent, who is responsible for triggering *Return Review* action. *PC Chair* is a recipient of *Review* flow, which is delivered by using Return Review action. Another difficulty is that the return review action is multiple. It can be triggered by one or more reviewers, which were appointed by *PC chair* for a specific paper. Example of a corresponding lower granularity activity diagram with swimlanes is represented in figure 2.

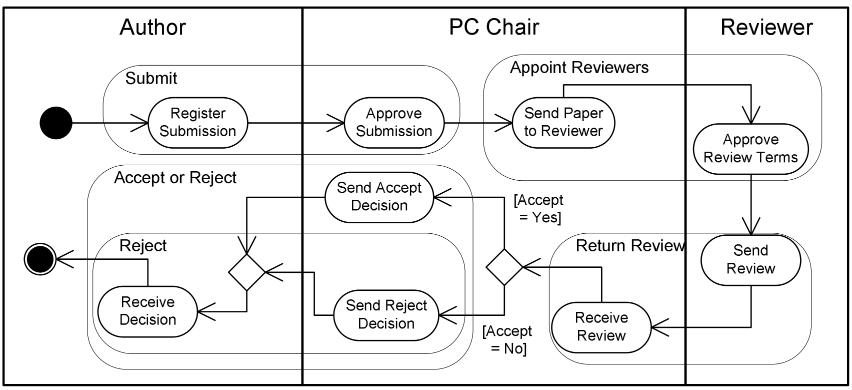

Fig. 2. Lower granularity activities in a conference review management system

As it is illustrated in this activity diagram, one business interaction between two organizational actors is broken down into two or more activities. Such finer granularity activities can be included without any problem into a use case or into activity diagram. The higher granularity business process states such as *Submit*, *Appoint Reviewers*, *Return Review*, *Accept* and *Reject* should be excluded from various UML diagrams for two major reasons:

- All these activities do not fit well a general use case definition (OMG, 2010), because they are communication actions between two actors. Designers are normally interested to focus on a use case functionality, which is performed by the system, to yield an observable result to one particular actor.
- Identification of object-oriented operations is crucial in designing class, state and sequence diagrams. The presented activities cannot be viewed as operations, which are invoked on one type of object.

Although the presented higher granularity business process states do not fit well the constructs of a use case diagram, they represent the communication actions, which are crucial for understanding crosscutting concerns of an enterprise system. The presented actions provide the natural way of decomposition of business process and therefore they must be explicitly captured in an enterprise model.

3 Conceptual Modeling of Behavioral and Structural Aspects

The presented five interactions (see figure 1) can be viewed as triggering events in various state transition diagrams. State diagrams are used to define the behavior of objects. Behavioral aspects can be represented as series of events, which may occur in one or more possible states. State transitions are triggered by events, which specify the permissible ways for changes to occur in different classes of objects. Graphical example of the corresponding state transition diagram is represented in figure 3.

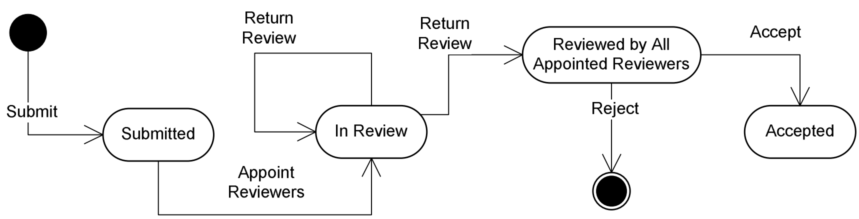

Fig. 3. Events and transitions of a paper object

This diagram represents states and state transitions of a *Paper* class object. State transitions are associated with events, which originate from five interactions in a conference managements system. Communication actions in this diagram are interpreted as triggering events: *Submit*, *AppointReviewers*, *ReturnReview*, *Accept* and *Reject*. The main problem with this diagram is that it is not representing related creation, manipulation or termination effects in various classes of objects.

Classes, associations and attributes are the most fundamental object-oriented constructs, which are used in UML class diagrams. They are stemming from the conventional conceptual modeling approaches, which were developed to capture the static aspects of concepts. Various classes of objects can be identified according to the presented initial description of conference management system such as Contact_Person, Submitted_Paper, Reviewing, Reviewer, Review, Accepted_Paper. Dependencies between concepts can be defined by associations as well as by inheritance, aggregation and composition relations. Associations between concepts are represented in terms of association ends and multiplicities in two opposite directions. The associations, classes and their attributes of a conference review management system are represented by the class diagram in figure 4.

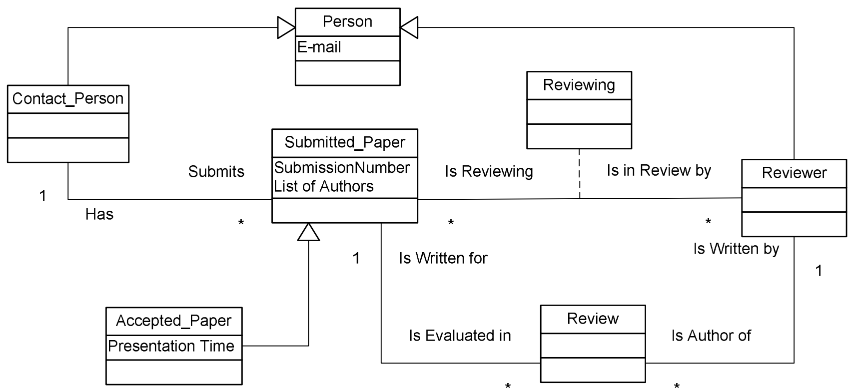

Fig. 4. Main classes and associations of a conference review management system

A state (see the previous diagram) is defined as a collection of object properties, which can be represented by attributes and associations with other classes. Three types of object transition effects can be distinguished in object-oriented design (Martin and Odell, 1998):

1) creation and termination of an object,
2) classification and declassification of an object,
3) connection and disconnection of a link between objects.

These effects are represented by various UML diagrams in a variety of ways. For instance, creation and termination of objects is visualized by the transitions from an initial state and to a final state. Connection and disconnection events can be specified by using sequence diagrams. Classification and declassification effects can be implemented by using sequences of object creation, connection, disconnection and termination operations. Classes, associations and attributes are necessary for understanding the effects of resulting structural changes from the events in a conference management system. One of the main challenges of the next section is to demonstrate the semantic integration of interaction events and related behavioral effects. The remaining part of this paper presents the modeling approach, which combines interactive, behavioral aspects and structural changes of objects in a single computation neutral diagram of a reasonable size.

4 Semantic Integration of Interactive and Behavioral Aspects

Conceptual models of interactions are not difficult to understand for business professionals as well as information system designers. Service interactions are helpful for clarifying why actors are willing to exchange flows with each other. Nevertheless, majority of conceptual modeling methods are not able to capture interaction flows between actors (Gustas & Gustiene, 2009). Actions and flows can be viewed as fundamental elements for defining business scenarios. A scenario is an excellent means for describing the order of interactions. Each interaction can be analyzed separately as it is required by the principle of separation of crosscutting concerns.

Interaction flows are special types of concepts that represent moving things. In our modeling approach, solid rectangles are be used for denotation of material flows and light boxes will indicate data flows. There are no material flows in the presented graphical description of a conference management system. An action with a missing data or material flow is understood as a decision or control. Actions, which will be represented by ellipses, are performed by actors. Actions are necessary for transferring flows between subsystems, which are represented as enterprise actors. Actors are denoted by square rectangles. They represent organizational and technical components of a system. Interaction flows among various actors of a conference management system are illustrated in figure 5.

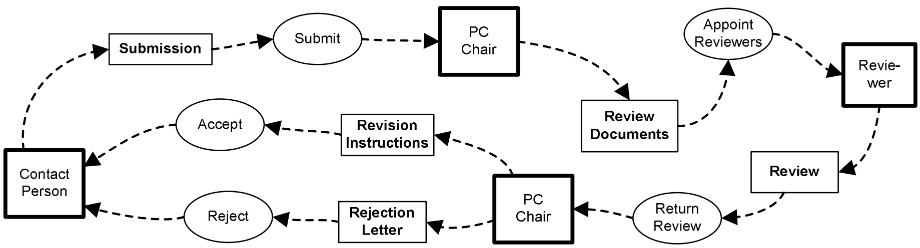

Fig. 5. Main actors and interactions of a conference management system

A contact person has possibility to *submit* a paper. If submission is accepted, the responsibility of a conference PC chair is to trigger the *appoint reviewers* action, which is used to send review documents to reviewers. Reviewer is obliged to deliver review to PC chair by triggering the *return review* action. PC Chair is authorized to *accept* or *reject* a submitted paper by informing a contact person with a special letter.

The main difference of this diagram in comparison with the presented sequence diagram (see figure 1) is that additionally it represents data flows between various actors. There are many other fundamental differences between these two kinds of diagrams, which will be discussed below.

In general, two kinds of interactions between actors can be distinguished (Dietz, 2006) such as production and coordination actions. Service requests are normally coordination actions, which are initiated by service requesters. Coordination actions are necessary to make commitment regarding the corresponding production action, which is supposed to bring a value to service requester. Production acts are normally performed by service providers. For example, the appoint reviewers can be viewed as coordination action, which corresponds to service request. The return review can be considered as a production action, because *Review* flow brings value to PC chair. Service requesters and providers are viewed as subjects or as active concepts (Gustas, 2010), which represent enterprise system components.

The behavioral and structural aspects of interactions can be analyzed in terms of reclassification, creation or termination effects. When two subsystems interact one may affect the state of each other (Evermann & Wand, 2009). Structural changes of objects can be defined in terms of object properties. Interaction dependency R(A┅▶B) between two active concepts A and B indicates that A subsystem can perform action R on one or more B subsystems. An action typically manipulates properties of some objects (Gustas & Gustiene, 2009). Otherwise, this action is not useful. Property changes may trigger objects transitions from one class to another. The internal changes of objects can be expressed by using transition links (——▶) between two object classes, which represent passive concepts. Graphical notation of reclassification construct is graphically represented in figure 6.

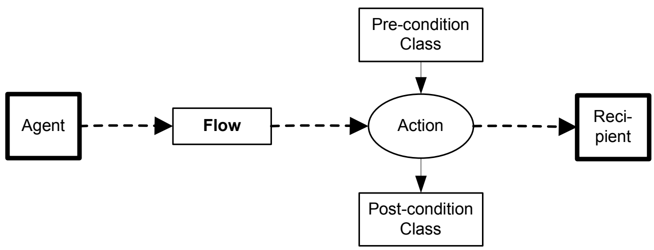

Fig. 6. Construct for representation of reclassification event

Two kinds of fundamental changes occur in reclassification action: removal of an object from a pre-condition class and creation of an object in a post-condition class. Reclassification construct with a missing post-condition class is used for representation of termination of object in a precondition class. A construct without a pre-condition class represents object creation in a post-condition class. For example, *Submit* action can be defined as creation event and *Reject* action can be viewed as termination. *Appoint Reviewers*, *Return Review* and *Accept* are reclassification events (see figure 8). Object creation or reclassification without any properties does not make any sense. So, various types of static and dynamic dependencies between classes are used to define mandatory properties of objects. The lack of noteworthy difference between pre-condition and post-condition class indicates that the specification of a communication action is either incomplete or a communication action is not purposeful. Pre-condition and post-condition classes are typically characterized by two different sets of mandatory attributes, which are sufficient for representing the permissible ways in which changes may occur. Static dependencies such as inheritance, composition, single-valued and multi-valued mandatory attributes are sufficient to visually recognize and comprehend the details of various interaction effects. Graphical notation of concept dependencies is presented in figure 7.

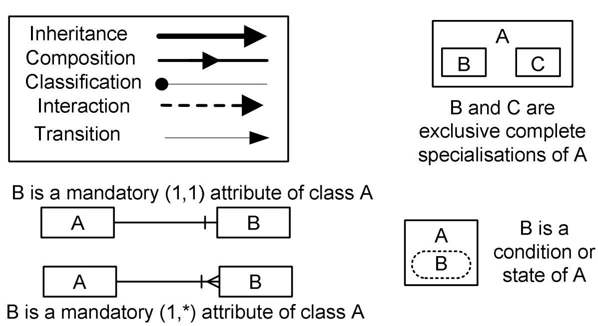

Fig. 7. Graphical notation of dependencies between concepts

One significant difference of our integrated modelling approach from traditional methods is that all dependencies are nameless. Concepts can be specialized by using special conditions or states. Any concept can be also defined as an exclusive complete generalization of other concepts. Actors are also specialized or decomposed by using inheritance, classification and composition dependencies. Inheritance dependency (➡) and composition dependencies (–▶–) can be used for reasoning about sharing static and dynamic dependencies between concepts. For example, the diagram, which is represented in Figure 5, can be extended by using the following dependencies:

Contact Person ➡ Author, Reviewer ➡ Person, PC chair ➡ Reviewer,
PC chair –▶– Conference, Reviewer –▶– Conference.

Composition dependency is a strict form of aggregation, which allows just either 1 or 1..* multiplicities between wholes and parts. Other cases of conventional composition are not legal in the presented modeling approach (Gustas, 2010). Composition and inheritance dependencies can be used for detection of inconsistent interaction dependencies on various levels of abstraction. The presented set of semantic dependencies is sufficient for unambiguous specification of creation, reclassification or termination effects in various classes of objects. These effects are fundamental for integration of behavioural and structural aspects of interactions. Analysis of creation, termination and reclassification effects can be performed by using a special set of rules, which are presented in the next section.

5 Semantic Integration of Behavioral and Structural Aspects of Objects

One of the most general ontological definitions of a system is provided by Bunge (Bunge, 1979). It served as a theoretical basis for understanding the notions of organization and enterprise ontology (Dietz, 2006). Bunge's ontological foundation is important for motivation of our semantic integration principles. They are as follows:

- Enterprise system can be decomposed into subsystems, which are represented as interacting actors,
- Every subsystem can be loosely coupled by interactions to other subsystems,
- When subsystems interact, they cause certain things to change. Changes are manifested via properties.

A transition arrow from and to action represents a control flow, which defines correspondingly termination and creation of various types of objects. A diagram showing object transitions and flows with states has most of the advantages of activity, state, sequence and class diagrams without most of their disadvantages when analyzed in isolation. Each action is used to superimpose interaction and object transition effects in a single diagram. Various combinations of the presented dependencies are fundamental for understanding sequences, alternatives, synchronizations or iterations of object creation, reclassification and termination effects. Behavioural, interactive and structural aspects of previously analyzed diagrams can be integrated into a single conceptual representation, which is presented in figure 8. Note that this conceptualization represents the semantic details of sequence, state transition, class diagrams (see figure 1, 2, 3, 4) including all interactions (see figure 5) with related creation and termination effects.

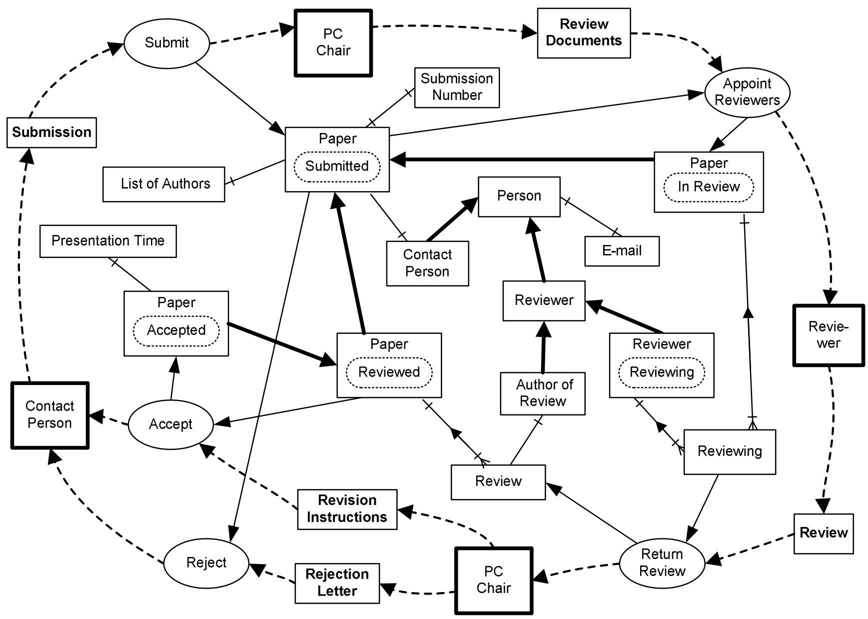

Fig. 8. Interactive, behavioral and structural aspects of a conference management system

1) The first event is *Submit.* According to the presented integrated diagram, the Submit action creates the following effects:

1.1) Creation of the Submitted Paper object, which is characterized by Submission Number and List of Authors. We assume that creation of an object in Contact Person class can be triggered by the Register Person action, which is not included in our initial description (see figure 9). We also assume that designers are interested just in one property of a contact person object. It is represented by the E-mail attribute.

1.2) Creation of the link between a Submitted Paper and one Contact Person object.

2) *Appoint Reviewers* is the second event, which is triggering reclassification of an object from the Submitted Paper into the Paper[In Review] class. Each Paper[In Review] object must be composed of one or more Reviewing objects (reified process). Reviewing is characterized by exactly one Reviewer. We assume that the objects of Reviewer class are created by other communication actions, which are outside of our initial description (a possibility for PC Chair to register a new reviewer is shown figure 9).

3) Changes of the *Return Review* event include creation and termination effects, which can be described as follows:

3.1) Creation of the object in Review class.

3.2) Creation of the mutual property links between the Reviewed Paper and Review object in two opposite directions.

3.3) Creation of the Author of Review property link for the Review object.

3.4) Termination of the corresponding Reviewing object with all its associated properties.

4) Triggering effects of the *Accept* event includes the reclassification of an object from the Reviewed Paper class into Accepted Paper class with an additional property of Presentation Time.

5) Triggering effects of the *Reject* event are defined by the termination of a Submitted Paper object. We assume that, in the case of a paper rejection, all related properties of the submitted paper object are removed. Each set of listed effects must be synchronized. Otherwise, the situation of data inconsistency may arise.

Inheritance dependencies (➨) are useful for reasoning about alternatives of communication action effects. The main rule for understanding of creation and termination effects is as follows:

Rule 1: Termination of an object in an inheritance hierarchy requires termination of all its specializations. For instance, termination of a person object is causing termination of a Reviewer and/or termination of a Contact Person objects (see the diagram in figure 8). Object of the more specific class requires creation of a more generic object. For example, a Paper[Accepted] object cannot be created by *Accept* action prior to Paper[Reviewed] object is created by the *Return Review* action.

Composition dependencies (−►−) are useful for reasoning about synchronization and iteration of object creation, reclassification and termination effects. Four rules (2, 3, 4 and 5) can be applied for understanding existential dependencies effects between wholes and parts. They are as follows:

Rule 2: Creation of object requires creation of all its compositional parts. For instance, the Appoint Reviewers action requires synchronous creation of Paper [in Review] together with at least one associated object of Reviewing. Note that reviewing object represents reified review process, which links one Paper and one Reviewer object in the state of Reviewing.

Rule 3: Termination of object requires termination of all its compositional parts. For example, the Reject action requires termination of a Paper in Submitted as well as in Reviewed states together with all Review objects as compositional parts.

Rule 4: Creation of the first part requires creation of a compositional whole. For example, if the Return Review action creates the first Review of some Paper object, it is necessary to synchronously create the Paper object in Reviewed state.

Rule 5: Termination of the last part requires termination of a compositional whole. For instance, if Return Review action terminates the last Reviewing object, it is necessary to terminate the compositional whole, which is represented by the Paper object in Review state.

Attribute dependencies are useful for reasoning about sequences of object creation, reclassification and termination effects. Rule 6 and 7 are useful for understanding manipulation effects of objects and their properties:

Rule 6: A property, which is viewed as an object on its own cannot be created prior to the creation of the object itself.

Rule 7: Removal of a mandatory object property causes termination of the object.

Any interaction can be used for instantiation or removal of objects and their properties. Some properties can be interpreted as objects on their own. Property can play a role of object, if it is characterized by its own properties. Such objects must be created prior to be connected as properties of other objects. For example, a Contact Person is a property of Paper [Submitted]. It is also an object, because each Contact Person object is characterized by E-mail, which is inherited from a Person. Instantiation of this object property of Submitted Paper cannot be done prior creation of a Contact Person object itself. The conceptualization, which is presented in figure 9, can be considered as a consistent extension of the previous diagram. It represents possible ways for creation and termination of various kinds of Person objects in a conference management system.

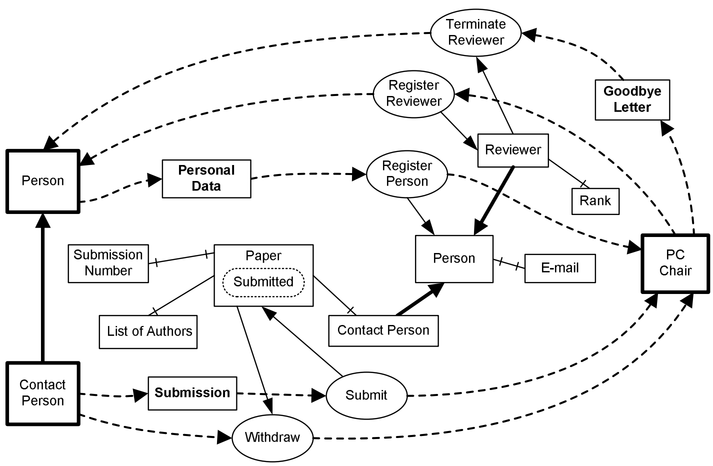

Fig. 9. Complementary interactions in a conference management system

Note that Submit communication action requires first creation of a Person object by using Register Person action. Thus, this action is considered as decomposition (like a use case inclusion) of Submit action. If the Register Person action is missing, then effects of Submit action can be described as follows: 1) Create a Person with E-mail property, 2) Create Paper [Submitted] with such properties as Submission Number, List of Authors and Contact Person, which is reclassified by using exactly one object of a Person class. The Register Person action can be also viewed as an extension of the Register Reviewer action.

Termination of object requires removal of object properties that are defined by the mandatory attributes of a pre-condition class. For example, termination of Reviewer will cause removal of Rank, but it will not cause removal of E-mail property, because an object of Person is not terminated. Sometimes, termination of object may cause removal of it as a property of other objects. Removal of one essential property may be sufficient for the violation of some mandatory dependency links, which would cause termination of other objects. Note that termination of an object (and removal all its properties) does not mean termination of the dependent objects. For example, Contact Person is a property of Paper [Submitted]. Termination of Paper [Submitted] with its contact person property does not mean termination of a Contact Person object, which is characterized by E-mail property. Nevertheless, the removal of a Person or a

Contact Person object would cause termination of the associated Paper[Submitted] objects. In some cases, post-condition class constraints may override termination of a pre-condition class object. One such case is reclassification action with the post-condition class, which is either specialization of the pre-condition class or the pre-condition class is viewed as a mandatory attribute of the post-condition class. Quite often objects are not preserved (see Reviewing in figure 8). They may pass several classes and then are terminated.

6 Concluding Remarks

Conceptual modeling methods, which put into foreground active concepts, typically focus on analyzing interactivity between subjects. The starting point of the enterprise modeling language ArchiMate (Lankhorst et al., 2010) as well as the DEMO approach (Dietz, 2006) is stemming from this modeling tradition. On the contrary, most of the conventional system analysis and design approaches put into the foreground modeling of passive concepts. Only few emerging approaches make attempts to express deep semantics of interplay (Dori, 2002) between active and passive structures of concepts. Analyzing various diagrams in isolation create difficulties in detecting requirement conflicts by business experts, who determine the organizational strategies. Consequently, information system methodologies are not able to bridge a communication gap among business experts and IT-system designers. The presented modeling approach for semantic integration for static and dynamic aspects provides several advantages. It is based on a single diagram type and therefore semantic integrity rules can be introduced directly into the model. Particular views, which define structural, behavioral or interactive aspects, can be generated by producing projections of an integrated model.

The lack of conceptual modeling approach, which can be used for the detection of semantic integrity problems among various types of diagrams, is the cornerstone of frustration for enterprise architects. The basic underlying principle in UML is to provide separate models for different modeling dimensions. According to the ontological principles, which are developed by Bunge (Bunge, 1979), the structural changes of objects are manifested via object properties. Properties in our modeling approach are expressed as mandatory attribute values. If diagrams are used to communicate unambiguously the semantic details of a conceptualized system, then optional properties should be proscribed (Gemino, 1998). The decline in cognitive processing performance, that occurs when optional attributes and relationships are used, appears to be substantial (Bodart et al., 2001).

The possibility to conceptualize interactions, behavior and effects of structural changes in a single diagram is not the only benefit of the presented modeling approach. Another important advantage is stability and flexibility of diagrams in dealing with evolutionary changes. Our initial studies demonstrate that separating and merging crosscutting concerns in the presented modeling approach is more efficient in comparison to the conventional modeling techniques. Typically, semantically equivalent changes that are required to be introduced in activity, state transition and sequence diagrams are quite substantial. We should focus on this topic in our future research. Conceptual modeling of service interactions is useful for separation crosscutting concerns among organizational and technical components. A new modeling approach was demonstrated on small scale example. Let us extend the initial description of a conference management system by introducing the following requirement: '*contact person should have a possibility to* ***withdraw*** *a submitted paper from a conference at any time'*. Such new requirement would cause a very simple extension of the diagram, which is represented in figure 8 (with withdraw communication action between Contact Person and PC chair). Termination of a Paper [Submitted] by Withdraw action is represented in figure 9. Related triggering effects of withdraw action can be visually recognized from the previous conceptualization (see figure 8) by using the rules, which are presented in this paper. There are four sets of effects, which can be identified by analyzing states of a Paper object such as Submitted, In Review, Reviewed and Accepted. Unfortunately, defining semantics of Withdraw action would require significant extensions of UML diagrams. Four new sequence diagrams must be introduced for specification of related effects. The presented state transition diagram (see figure 3) should be extended by four new state transitions with their associated events and effects. The complexity of the activity diagram would also increase dramatically. The problem is that four new variations of withdraw activities must be introduced for evaluation of withdrawal conditions of *Submit*, *Appoint Reviewers*, *Return Review* and *Accept* events.

Traditional information system analysis and design methods are projecting the structural and behavioral aspects of conceptualizations into totally different types of diagrams. The UML individual diagram types are clear, but integrated semantics among models is missing. That is why object-oriented diagrams are difficult to apply for business logics alignment with implementation specific design for making both organizational and technical system parts more effective. The presented modeling approach is capable to capture, in concise form, semantics of structural changes, which are motivated by interaction flows. Similarity of conceptual models before and after adding a complimentary requirement, demonstrates stability of the presented modeling approach.

References

Blaha, M., Rumbaugh, J.: Object-Oriented Modelling and Design with UML. Pearson, London (2005)

Bodart, F., Patel, A., Sim, M., Weber, R.: Should Optional Properties be Used in Conceptual Modelling? A Theory and three Empirical Tests. Information Systems Research 12(4), 384–405 (2001)

Bunge, M.A.: Treatise on Basic Philosophy Ontology II: A World of Systems, vol. 4. Reidel Publishing Company, Dordrecht (1979)

Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, Berlin (2006)

Dori, D.: Object-Process Methodology: A Holistic System Paradigm. Springer, Berlin (2002)

Evermann, J., Wand, Y.: Ontology Based Object-Oriented Domain Modeling: Representing Behavior. Journal of Database Management 20(1), 48–77 (2009)

Gane, C., Sarson, T.: Structured System Analysis. Prentice Hall, NJ (1979)

Gemino, A.: To be or maybe to be: An empirical comparison of mandatory and optional properties in conceptual modeling. In: Proc. Ann. Conf. Admin. Sci. Assoc. of Canada, Information Systems Division, Saskatoon, pp. 33–44 (1998)

Glinz, M.: Problems and Deficiencies of UML as a Requirements Specification Language. In: Proc. of the 10-th International Workshop on Software Specification and Design, San Diego, pp. 11–22 (2000)

van Griethuisen, J.J.: Concepts and Terminology for the Conceptual Schema and Information Base, Report ISO TC97/SC5/WG5, No 695 (1982)

Gustas, R., Gustiene, P.: Pragmatic – Driven Approach for Service-Oriented Analysis and Design. In: Information Systems Engineering - from Data Analysis to Process Networks. IGI Global, USA (2008)

Gustas, R., Gustiene, P.: Service-Oriented Foundation and Analysis Patterns for Conceptual Modelling of Information Systems. In: Information System Development: Challenges in Practice, Theory and Education, vol. 1. Springer, Heidelberg (2009)

Gustas, R.: A Look behind Conceptual Modeling Constructs in Information System Analysis and Design. International Journal of Information System Modeling and Design 1(1), 79–108 (2010)

Gordijn, J., Akkermans, H., van Vliet, H.: Business Process Modelling is not Process Modelling. In: Mayr, H.C., Liddle, S.W., Thalheim, B. (eds.) ER Workshops 2000. LNCS, vol. 1921, pp. 40–51. Springer, Heidelberg (2000)

Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of 'Semantics'? IEEE Computer, 64–72 (October 2004)

Jacobson, I.: NG, P-W. Aspect-Oriented Software Development with Use Cases. Pearson Education, Pennsylvania (2005)

Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Pearson Education, NJ (2009)

Lankhorst, M.M., Proper, H.A., Jonkers, H.: The Anatomy of the ArchiMate Language. International Journal of Information System Modeling and Design 1(1), 1–32 (2010)

Martin, J., Odell, J.J.: Object-Oriented Methods: A Foundation (UML edn.) Prentice-Hall, Englewood Cliffs (1998)

OMG. Unified Modeling Language Superstructure, version 2.2. (2010), `http://www.omg.org/spec/UML/2.2/` (retrieved January 19, 2010)

Yourdon, E., Constantine, L.L.: Structured Design. Prentice Hall, NJ (1979)

Towards Defining a Competence Profile for the Enterprise Modeling Practitioner

Anne Persson[1] and Janis Stirna[2]

[1] University of Skövde, Informatics Research Centre, P.O. Box 408,
SE-541 28 Skövde, Sweden
anne.persson@his.se
[2] Department of Computer and Systems Sciences, Stockholm University, Forum 100,
SE-1644 0, Kista, Sweden
js@dsv.su.se

Abstract. Enterprise Modeling (EM) has established itself as a valuable instrument for various purposes related to organizational development, such as designing or redesigning the business, eliciting requirements for information systems, capturing and reasoning about organizational knowledge. A notable characteristic of EM is its collaborative way of stakeholder involvement in modeling. Much of the success of projects using EM depends on how the EM process is organized and on the competence level of the expert responsible for the EM approach. This paper analyses what are the competence needs for the method expert and what competences are needed in the different steps in the EM process. The EM process described consists of activities for project inception and planning, conducting modeling sessions, and delivering a result that can be taken up by a subsequent implementation project. Two main competence areas are discussed in relation to the EM process – competences related to modeling and competences related to managing EM projects.

Keywords: Enterprise modeling, modeling practitioner, competence profile.

1 Introduction

Enterprise Modeling (EM) is a process where an integrated and negotiated model describing different aspects of an enterprise is created. An Enterprise Model consists of a number of related "sub-models", each describing the enterprise from a particular perspective, e.g. processes, business rules, goals, actors and concepts. EM has for some years been a central theme in information systems (IS) engineering research. There are two main reasons for using EM [1]:

Developing the business – this entails developing business vision, strategies, redesigning the way the business operates, developing the supporting information systems, capturing IS requirements, etc.

Ensuring the quality of the business – here the focus is on two issues: 1) sharing the knowledge about the business, its vision, the way it operates, and 2) ensuring the acceptance of business decisions through committing the stakeholders to the decisions made.

P. van Bommel et al. (Eds.): PoEM 2010, LNBIP 68, pp. 232–245, 2010.

Examples of EM methods can be found in [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12]. Examples of application domains for EM can be found in [13, 14, 15, 16, 17, 18, 19 and 20].

Some method developers have advocated a participatory way of working (see e.g. [7, 8, and 21]). In facilitated group modeling, participation is *consensus-driven* in the sense that it is the domain stakeholders who "own" the model and govern its contents. In contrast, *consultative* participation means that analysts create models and domain stakeholders are then consulted in order to validate the models.

In the participatory approach to EM, stakeholders meet in modeling sessions, led by a facilitator, to create models collaboratively. In the modeling sessions, models are often documented on large plastic sheets using paper cards. The "plastic wall" is viewed as the official "minutes" of the session, for which every participant is responsible. Establishing effective and consensus-driven participation requires:

- achieving active communication and lively discussion between individuals and between groups of individuals. This increases the chances of identifying different views on the problem to be discussed.
- creating a group, i.e. to make people feel that they work towards the same goal which increases the chances of achieving a good modelling result.

More on the participative approach to EM can be found e.g. in [8 and 21]. The EM process described in this paper is based on the view that a participatory way of working is the main approach to EM.

A large amount of research has been dedicated to the development of new modeling languages and to the refinement of existing EM languages, while their use in practice has attracted much less attention. EM practice can be discussed from a number of alternative perspectives, such as e.g.:

- the ability of modeling languages to express aspects of the domain being modeled,
- the usability of modeling languages,
- the role of enterprise models in information systems engineering,
- the effect on the systems development process of using enterprise models, and
- the applicability of modeling languages in different contexts.

This paper addresses a perspective on EM that has been more or less neglected in the scientific literature: the competency of the modeling practitioner. Some references exist (see e.g. [7, 8 and 21]), which mainly focus on the modeling expert in her/his capacity as facilitator. We have not found any previous research that takes a broader view on the competency required throughout a whole EM project.

Therefore, the goal of the paper is to define a set of core competencies for the modeling practitioner and to relate these core competencies to a detailed stereotype EM project process. A modeling practitioner is defined as someone who is responsible for running part of or the whole EM project process towards effectively achieving its goals.

The remainder of the paper is organized as follows. Section 2 discusses the concept of EM competence. In Section 3 the research approach is presented. The process of a stereotypical EM project is defined in Section 4, while the core competences related to the defined EM process are included in Section 5. The paper ends with some concluding remarks in Section 6.

2 EM Competence – A Critical Resource to Achieve the Goals of EM

Human knowledge and competence is a critical resource for achieving the goals of EM. There are two reasons for this:

- Models contain human knowledge about an organization in its current or perceived future state. We need domain experts who contribute this knowledge.
- The knowledge that domain experts have needs to be captured and structured in enterprise models that contribute to the EM goals. We need modeling practitioners who are able to do this.

The concept of competency is complex and can be defined in a number of ways. For the purposes of this paper we state that competency has four main aspects:

1) *Knowledge* - a person's factual knowledge about a specific subject matter, as a result of e.g. education.
2) *Skills*- a person's ability to actually use the knowledge to achieve goals.
3) *Individual properties*- a wide range of personal characteristics e.g. social skills, intelligence, flexibility, integrity, ability to co-operate, courage etc.
4) *Willingness to contribute competency* - a person's attitude towards actually contributing her/his knowledge and skills to the achievement of goals other than her/his own.

In this paper we concentrate on skills and individual properties, and we particularly target the competence of the modeling practitioner. In the following we give an overview of the roles in EM.

An Enterprise Model comprises knowledge about different aspects of some organization in its current or perceived future state. *Domain experts* provide this knowledge and are responsible for the correctness and relevance of that knowledge in the context of the EM project.

To create the Enterprise Models, using a participative approach, a *modeling practitioner* is needed. This is the role that the professional EM practitioner plays in an EM project. They can take on a number of sub-roles, e.g. EM project leader, facilitator of a modeling session, and tool expert. As *project leader*, the modeling practitioner negotiates and plans the modeling project together with the project or problem owner. A *facilitator* moderates each modeling session. In a session there can be more than one facilitator and also a *tool expert*. A larger modeling project will typically have several facilitators and tool experts forming a *modeling practitioner team*, which is headed by *project leader*. The team leader should be an experienced facilitator.

Figure 1 shows the main difference in responsibilities of domain experts and modeling practitioners. Arguments for this separation of responsibilities are given in [22].

The main responsibility of the modeling practitioner is that the models produced have good enough quality to accomplish the project goals. It is also to ensure that the chosen EM method is suitable for modeling the problem at hand and that the method is effectively used to accomplish the project goals. This means not only to use the

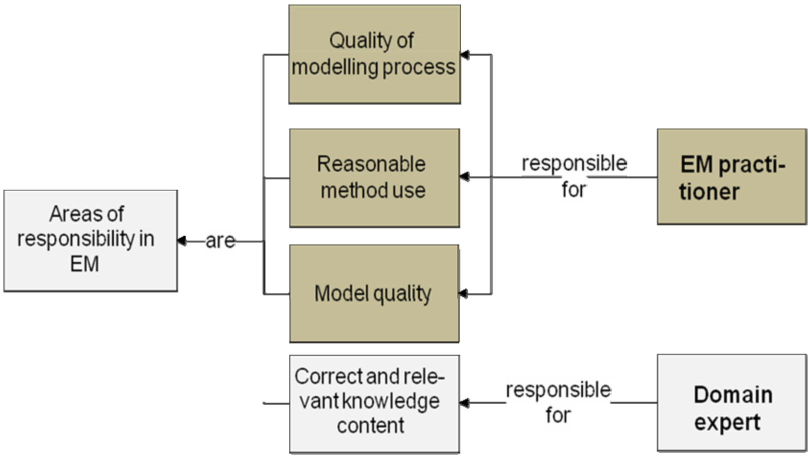

Fig. 1. Actors in EM and their responsibilities

method's notation in a reasonable way but to also construct and to run a modeling process that makes the best of available resources, e.g. the knowledge and abilities of domain experts. The modeling practitioner is also responsible for making sure that the project resources are used in a way that enables the modeling project to be completed on time and in such a way that the goals are achieved.

The main challenge of the EM process is to ensure that the quality of its outcome is fit for the intended use. Potential outcomes of EM are, e.g., models, decisions and enhanced knowledge among those involved in the EM process. Since the EM process is highly intellectual, it is dependent on the competence of its participants. If the proper EM competence is not available in the EM process, the effects of EM will not appear. In the following we will focus on the competence of the EM practitioner.

3 Research Approach

The research approach taken in this paper follows the principles of design science [23]. This section discusses how the seven guidelines [ibid] of design science have been addressed in this research.

Guidelines 1: Design as an Artifact. We consider the competence profile for EM practitioners and its alignment to the EM process a design artifact. It contributes to making EM approaches more operational, i.e. easy to use in practice.

Guideline 2: Problem Relevance. Many organizations struggle to adopt EM and to carry out EM projects. Currently the knowledge of what competences and skills are needed in a successful project is to a large extent tacit possessed by only a few experienced EM practitioners. Hence, this knowledge should be captured and presented in such a way it can be applied by a broad range of practitioners. The need for addressing the competence issue in EM is supported by interview studies on the practice of EM reported in [22] and on the EM tool usage in [24].

Guideline 3: Design Evaluation. The proposed design artifact has been validated by informed arguments and its internal consistency has been ensured by linking the proposed competence profile to the EM process. In addition the proposed competence profile has been applied in field studies of staffing and carrying out real life EM projects. More about some of these projects is available [25].

Guideline 4: Research Contribution. To the best of our knowledge the competence needs for EM is relatively unexplored and similar competence profile linked to the steps of the EM process does not currently exist.

Guideline 5: Research Rigor. The initial theoretical constructs of how EM application projects should be organized and carried out emerged from grounded theory studies reported in [22 and 24], projects such as F3 [8], ELEKTRA [26] and HyperKnowledge [27 and 28]. The findings were further extended by participating in and analysis experiences from EM activities in projects such as Mapper [29, 30] and InfoFlow [31]. All together the findings are based on more than 100 modeling sessions during the years from 1993 to present.

Guideline 6: Design as a Search Process. We have developed the proposed artifact iteratively and incrementally over numerous cycles going from generating alternatives to then validating them against real life requirements for EM projects and back.

Guideline 7: Communication of Research. [23] suggest that "design science research should be presented to technology-oriented as well as management audiences". The interpretation of this in our case is that the proposed competence profile for EM experts should be presented both to researchers developing new EM approaches and tools as well as to practitioners using these EM technologies. Hence, we choose PoEM as the most suitable forum to present our research.

4 Enterprise Modeling Projects – A Process View

This section presents the EM process, shown in figure 2. We consider this a stereotyped process, because in real life projects the actual steps and information sets might differ slightly. It is also possible that additional steps are needed, e.g. to ensure integration with another development project or to involve a broad group of stakeholders.

The EM process follows generic principles of carrying out projects for various purposes. This is because we strongly believe that aligning EM activities with the general project activities improves stakeholder acceptance of the modeling way of working.

Table 1 shows how different actors are involved in the steps of the EM process.

Process 1: Define scope and objectives of the EM project. We assume that the EM project is commissioned either as a result of selling consulting services or another in-house development project has decided to address a specific problem area by a modeling approach. In either case there usually exists an initial problem statement (inf. set 1) and an organizational actor that will benefit from solving the problem–problem owner. At this stage the problem owner and the EM project leader should discuss the problem to find its boundaries, what the likely ways of solving it might be, and what the expected outcomes are. This would form a project definition (inf. set 3). In this process model we assume that the organization has already assessed its suitability for using the participative approach to EM, but if has not been done, or

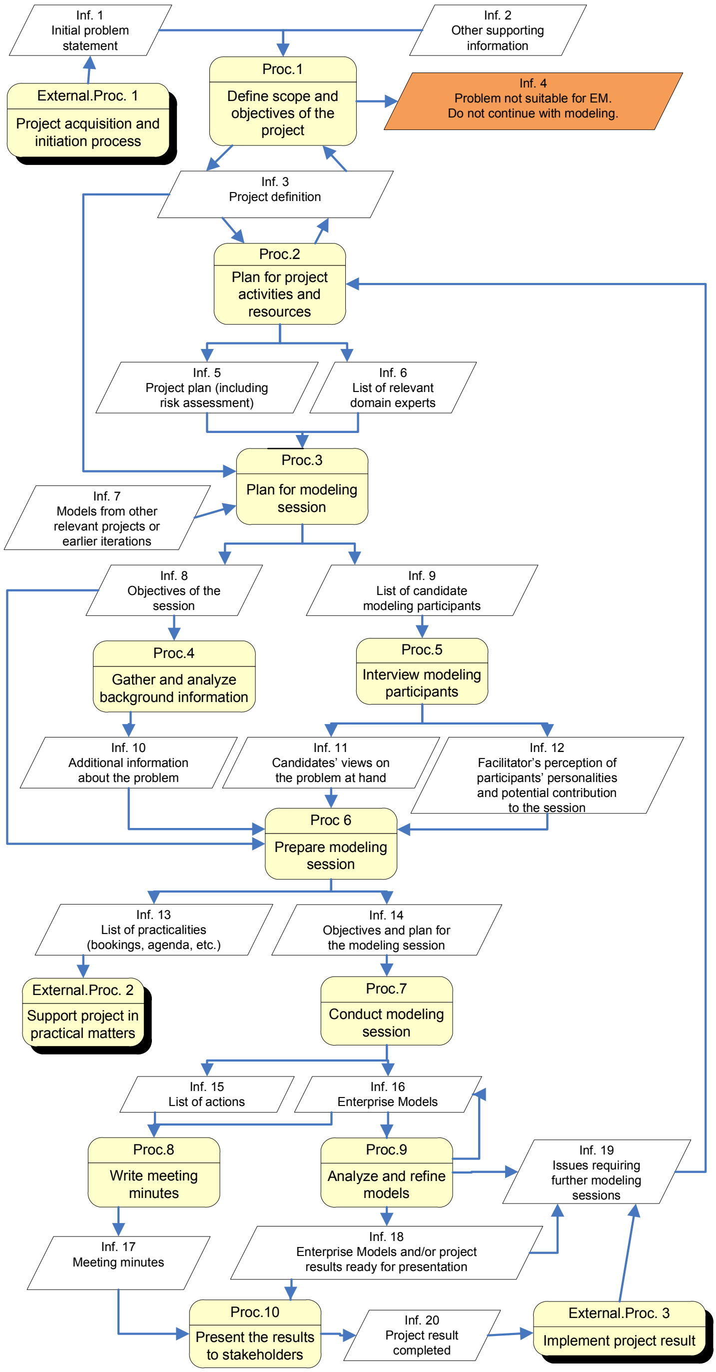

Fig. 2. The EM process model showing processes and information sets

Table 1. Actor involvement in the EM process steps (R- responsible, P- participates)

EM Process Step	Problem owner	Domain expert	EM project leader	EM facilitator	Tool expert
P1 Define scope and objectives of the project	R		P		
P2 Plan for project activities and resources	R		P	P	
P3 Plan for modeling session	P		R	P	
P4 Gather and analyze background information			P	R	
P5 Interview modeling participants		P		R	
P6 Prepare modeling session	P		P	R	
P7 Conduct modeling session		P		R	P
P8 Write meeting minutes			P	R	P
P9 Analyze and refine models	P		P	R	P
P10 Present the results to stakeholders	R	P	P	P	

some doubts arise (e.g. a strong sense of hidden agendas) then the EM project leader should assess the situation in the organization. The problem should also be assessed for being suitable for EM. More about assessing the organization and the problem at hand is available in, e.g. [21, 22, and 29]. If the organization or the problem is found to be unsuited for EM, then the problem owner and the project leader should choose other ways of solving the problem, e.g. by the consultative approach or by brainstorming. When dealing with complex and/or wicked problems [32] it might be difficult to formulate a clear problem definition. In such cases the project might organize a modeling session with an objective to find out what the real problem is and how to tackle it.

Process 2. Plan for project activities and resources. At this stage the EM project leader, problem owner and facilitator plan specific activities to be carried out. This includes the overall number and schedule of modeling sessions, the issues addressed in them (inf. set 5), as well as indicating relevant domain experts to be involved in the modeling sessions later (inf. set 6). Additional issues to pay attention at this stage are risk assessment, resource allocation, both for the method provider team and for the domain experts, and establishing project groups' overall authority, i.e. mandate to solve the problem.

Process 3. Plan for modeling session. The objective is to plan a specific modeling session, i.e. to set its overall objective and questions to be addressed (inf. set 8). Existing models produced in previous modeling session of the project or earlier projects in the organization and/or other supporting information might also be analyzed. The initial list of relevant domain experts (inf. set 6) should be analyzed and candidates for involving in the modeling session should be selected (inf. set 9).

Process 4 Gather and analyze background information. The modeling facilitator usually needs to obtain additional information to learn more about the organization and the background of the problem at hand.

Process 5. Interview modeling participants. The candidates for involving in the modeling session (inf. set 9) are interviewed individually in order to learn more about their views on the problem at hand (inf. set 11) and to assess the participant's potential contribution at the modeling session (inf. set 12). A benefit for the candidate is that he/she is able to learn about the project and the upcoming modeling session in advance. In some projects it is beneficial to interview more participants that are going to be used in the modeling sessions, because this allows the project team to learn more about the organization and, indirectly, to spread the word about the project and the coming change in the organization.

Process 6. Prepare modeling session. At this stage a detailed plan for the modeling session (inf. set 14) is elaborated by analyzing the background material and findings from the interviews. This plan should include specific objectives the modeling session, specific questions to be addressed, preliminary set of enterprise models to be developed (e.g. goal models, concepts models, actor models), a set of driving questions for starting the discussion, the expected level of model quality. The modeling facilitator should also assess various risks and scenarios of how the modeling session might develop. E.g. what are the topics that the participants will not talk willingly, what are the topics that might lead the discussion astray, what can cause conflicts, how to act in case of a conflict. This should be done in collaboration with the problem owner and project leader. The practicalities of the meeting (inf. set 13) should also be organized, which includes location, agenda, travel plans, etc.

Process 7. Conduct modeling session. The objective of this paper is not to describe details of how a modeling session is conducted. Recommendations of what to do and what not to are available, for example, in [29, 30, 33, 34, and 35]. The tangible outcome of the modeling session is the models produced (inf. set 16) and an additional list of actions for implementing the decisions made during the modeling session (inf. set 15). Additional intangible outcomes of modeling are participants' improved understanding of the problem area and a firmer commitment to the decisions made [22 and 36].

Process 8. Write meeting minutes. After the modeling session it is recommended to write minutes of the meeting (inf. set 17) which includes the models as in the state were produced at the modeling seminar and action list. At this stage the models should not be more refined because the main purpose of this activity is to send notes to the participants which might also serve as a reminder of the actions that they have agreed to be responsible for.

Process 9. Analyze and refine models. Enterprise Models created at a modeling session usually need further refinement in terms of presentation and layout, as well as content. The result of the modeling session should also be analyzed with respect to the objectives of the session and the project. This either leads the project team to a conclusion that the expected result is achieved and can be presented to the organization (inf. set 18). Otherwise the team indentifies a set of issues for further development and modeling (inf. set 19) and proceeds with planning subsequent project activities (process 2). In many cases information sets 18 and 19 are reports of the project activities.

Process 10. Present the results to stakeholders. The modeling project ends with presenting the results to the problem owner and relevant stakeholders. A part of this presentation is decision making on how the results should be implemented or taken-up by the organization. It might also be that the stakeholders indentify issues that are not resolved and require further development (inf. set 19).

The EM process we have outlined ends when the problem owner and the involved stakeholders feel that they have a result that can be implemented. In practice the EM project results will most likely serve as input for another development project, including an IT or IS development project.

The EM process described in this section may appear easy to conduct on the outset. In reality however there are many challenges to succeed and pitfalls to avoid, particularly in the project preparation phase (processes 1 to 6). Much of this knowledge is related to organizational and social issues and hence is not easily formalizable. For example, in [34] we have proposed to capture some of this knowledge in the form or anti-patterns. But in addition to capturing and sharing knowledge about best and worst practices, much of the success depends also on the competence of the modeling facilitator, which we will discuss in the next section.

5 Core Competences in Enterprise Modeling Projects

Previous research [22] has identified three levels of EM practitioner competence:

- Ability to model, which means that a person is able to construct an Enterprise Model which is syntactically correct according to the used EM language and that the model in a reasonable way reflects the domain and problem in question.
- Ability to facilitate modeling sessions, which means that a person is able to lead a group of domain experts in creating/refining an Enterprise Model and doing it in such a way that the group's knowledge and abilities work together to create a high quality model.
- Ability to lead EM projects towards fulfilling their goals and making the best of the project resources.

The list of relevant competences that are useful for acting at each level can potentially be very long. We claim that in order to target the *main* challenge of the EM process, which is to produce an EM outcome that is fit for its intended use, we need to define a set of essential core competences that target the quality of the outcome of EM. In the following we describe the core competences that our research has yielded so far. They fall into two distinct categories: 1) those related to modeling itself, i.e. the ability to model and the ability to facilitate participatory modeling session. These competences are at the heart of modeling and 2) those related to setting up and managing EM projects.

5.1 Competences Related to Modeling

The *ability to model* involves making use of the chosen EM language to create and refine enterprise models. The resulting models should reflect the discussion in the modeling session and focus on the problem at hand. Knowing how to use modeling tools for documenting and analyzing the modeling result is also included in this ability. One important, and sometimes neglected, aspect is the ability to create a readable model, because they tend to become large and graphically complex.

Since we advocate a participatory approach to EM, the *ability to facilitate a modeling session* is essential. Facilitation is a general technique used in group processes for a wide variety of purposes, also within EM (see further e.g. [37] and International Association for Facilitators (IAF) http://www.iaf-world.org). This ability is very much based on knowledge about the effects of modeling, the principles of human communication and socialization (especially in groups), as well as the conditions of human learning and problem solving (cognition). For EM, some of the more important aspects of this competence are to condense and capture important ideas, to pose questions that trigger discussion, to listen, to summarize and generalize, as well as to drive the discussion towards fulfilling the goals of the EM session.

For both of these abilities we want to highlight the fact that the competence requirements are quite different if EM is used to capture the current situation compared to designing a future situation. In the latter case the ability of the EM practitioner will be geared towards drawing out the creativity of the domain experts and to guide that creativity towards the goals of the session.

5.2 Competences Related to Managing EM Projects

In order for the models to be fit for their intended use, the EM practitioner needs the *ability to select an appropriate EM approach and tailor it in order to fit the situation at hand.* Sometimes that choice is restricted by the requirements of the context of use, as e.g. is the case when EM is used in an IS development project that uses a particular method and tool-set. In other cases the choice of EM approach is up to the EM practitioner. Based on her/his knowledge about the problem at hand, the requirements on the EM result, the preferences and modeling skill level of the modeling group, and the context in which EM will be used the EM practitioner will choose an appropriate approach. The professional EM practitioner will have a "tool-box" of potential methods for different purposes that she/he is able to use. Independently of whether the EM practitioner has the choice of approach, the approach often needs to be tailored to fit the situation at hand and she/he will then need to be able to assess the consequences of any changes made to the approach.

In participatory EM the *ability to interview involved domain experts* before the EM session is critical. In this situation the social skills of the EM practitioner are essential, such as e.g. ability to listen, ability to read body language. In a discrete way the EM practitioner needs to ask the domain expert what should be talked about in the modeling session and also try to find out what topics should be avoided and why.

For EM to have effect in its context of use, it needs to be focused towards a particular goal or problem. This pertains both to the overall EM project level and to each EM session. The *ability to define a relevant problem* that is feasible to model based on the information that the EM practitioner can obtain is, therefore, important. This ability is very much related to the ability to interview domain experts. In this ability the capacities to conceptualize, generalize and to assess the relationships between different problems are included. An essential aspect of defining the relevant problems is the ability to spot hidden agendas, which builds both on the practitioner's previous experience but also on her/his social skills and ability to "read between the lines" in a conversation. Unidentified hidden agendas can potentially cause problems later on in the EM project. Assessing the complexity of a problem is also part of defining a problem. Problem complexity is a heavy influence on the planning of the project both in terms of activities and resources. It can be argued that it is impossible to define a clear problem on the outset and that it will change as the project proceeds. This is true, but in order for the project to become operative at least a "working problem" is needed.

In planning an EM project and an EM session the *ability to define requirements on the results* are essential in order for project/session goals to be achieved. These requirements relate to the models that are to be produced as well as what is to be achieved by these models. Sometimes the requirements have to do with the process itself. E.g. by involving certain stakeholders and having them listen to what other stakeholders have to say in a participatory EM session, certain change decisions can be made less dramatic for the organization. The EM practitioner should also keep in mind that the models produced is the tangible result of modeling, but equally important is the intangible result – participants' changed thinking and understanding of the problem.

The *ability to establish a modeling project* is critical in order to create the most beneficial conditions for the EM project. Favorable conditions will increase the chances of obtaining the desirable effects of EM. Conditions involve resources in terms of time and competence (domain as well as EM practitioner competence) as well as authority for EM project participants to act freely and make decisions within the project definition. This ability is essential in any project.

The result of modeling will be used for a specified purpose. In order for that purpose to be fulfilled the users of the result need to understand it and its implications. This means that the modeling practitioner will have to present it in oral and/or written form to them. Depending on the target audience, certain aspects of the result will need to be emphasized or toned down. E.g. presenting project results to a group of managers the detailed data structure of the supporting IS can be omitted. This requires an ability *to adjust a presentation of project results and issues related to them to various stakeholders.*

An EM project is a signal to the organization that change of some kind is imminent. This means that various stakeholders will try to influence the EM practitioner so that their own goals will be those of the EM project. To *navigate between the wishes of various stakeholders while upholding the EM project goal* is, therefore, a critical competence. More about the challenges involved in tackling this problem can be found in [38].

EM projects typically deliver a solution to a business problem. The solution usually consists of an organizational design proposal (which might include an IT solution) reflected in Enterprise Models. A partially intangible outcome of the EM project is the supporting set of decisions and commitment to implement the solution.

Table 2. Matching of EM process steps to core competences

Process / Ability	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
to model							X		X	
to facilitate modeling sessions							X			
to interview involved domain experts					X					
to define a problem	X		X			X			X	
to define requirements on the results	X	X	X							
to establish a modeling project	X									
to adjust presentation of project results						X				X
to navigate between the wishes of stakeholders while upholding a defined project strategy	X	X			X		X			X
to assess the impact of the modeling result and the modeling process in the organization	X	X				X			X	X

An example issues to consider are: would the solution appear to be inappropriately bureaucratic, democratic, authoritative; what kind of implementation activities are needed, etc. An ability *to assess the impact of the modeling result and the modeling process in the organization* is therefore needed to drive the modeling effort towards a solution that has a high probability of being implemented within the organization.

In Table 2 the core competences are summarized and mapped to the process steps defined in Figure 1.

6 Concluding Remarks

In this paper we have identified a set of core competencies based on a large number of EM projects and EM sessions conducted during more than 10 years. One important message in our findings is that the quality of the outcome of EM does not only depend on the practitioner's ability to model and to lead modeling sessions. At least as important is the EM practitioner's ability to manage an EM project. This means that being a professional EM practitioner involves integrating core modeling activities into an overall EM project management process that is geared towards effectively fulfilling the goals of the EM project. As a consequence, the role of managing EM projects is not something for the inexperienced EM practitioner.

Looking at EM education and training, particularly in the university context, it is mostly focused on the ability to model. Often it is assumed that with that ability comes, automatically, the ability to facilitate modeling sessions and particularly to manage EM projects. Based on our research we find that this is clearly not the case. In consequence, the education and training of EM practitioners should focus more on aspects related to effectively managing EM in its context of use.

References

1. Persson, A., Stirna, J.: An explorative study into the influence of business goals on the practical use of Enterprise Modelling methods and tools. In: Tenth International Conference on Information Systems Development (ISD 2001), September 5-7, University of London Royal Holloway (2001)
2. Bajec, M., Krisper, M.: A methodology and tool support for managing business rules in organizations. Information Systems 30(6), 423–443 (2005)
3. Castro, J., Kolp, M., Mylopoulos, J.: A Requirements-Driven Software Development Methodology. In: Dittrich, K.R., Geppert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 108–123. Springer, Heidelberg (2001)
4. Dobson, J., Blyth, J., Strens, R.: Organisational Requirements Definition for Information Technology. In: Proceedings of the International Conference on Requirements Engineering 1994, Denver/CO (1994)
5. Johannesson, P., Boman, M., Bubenko, J., Wangler, B.: Conceptual Modelling. Prentice Hall International Series in Computer Science. Prentice Hall, Englewood Cliffs (1997)
6. Bubenko Jr., J.A.: Extending the Scope of Information Modelling. In: Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Lloret, Costa Brava Catalonia, September 20-22 Department de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, Report de Recerca LSI/93-25, Barcelona (1993)

7. Bubenko Jr., J.A., Persson, A., Stirna, J.: User Guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns. In: Deliverable D3, IST Programme project Hypermedia and Pattern Based Knowledge Management for Smart Organisations, project no. IST-2000-28401, Royal Institute of Technology, Sweden (2001)
8. F3 Consortium, F3 Reference Manual. ESPRIT III Project 6612, SISU, Stockholm (1994)
9. Fox, M.S., Chionglo, J.F., Fadel, F.G.: A common-sense model of the enterprise. In: Proceedings of the 2nd Industrial Engineering Research Conference, Institute for Industrial Engineers, Norcross,GA (1993)
10. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises. Springer, Heidelberg (2008) ISBN: 978-3-540-79415-8
11. Loucopoulos, P., Kavakli, V., Prekas, N., Rolland, C., Grosz, G., Nurcan, S.: Using the EKD Approach: The Modelling Component, UMIST, Manchester, UK (1997)
12. Yu, E.S.K., Mylopoulos, J.: From E-R to ”A-R-” Modelling Strategic Actor Relationships for Business Process Reengineering. In: Proceedings of the 13th International Conference on the Entity-Relationship Approach, Manchester, England (1994)
13. Wangler, B., Persson, A., Johannesson, P., Ekenberg, L.: Bridging High-level Enterprise Models to Implemenation-Oriented Models. In: Fujita, H., Johannesson, P. (eds.) New Trends in Software Methodologies, Tools and Techniques. IOS Press, Amsterdam (2003)
14. Niehaves, B., Stirna, J.: Participative Enterprise Modelling for Balanced Scorecard Implementation. In: 14th European Conference on Information Systems (ECIS 2006), Gothberg, Sweden (2006)
15. Stirna, J., Persson, A., Aggestam, L.: Building Knowledge Repositories with Enterprise Modelling and Patterns - from Theory to Practice. In: Proceedings of the 14th European Conference on Information Systems (ECIS 2006), Gothenburg, Sweden (June 2006)
16. Wangler, B., Persson, A.: Capturing Collective Intentionality in Software Development. In: Fujita, H., Johannesson, P. (eds.) New Trends in Software Methodologies, Tools and Techniques, pp. 262–270. IOS Press, Amsterdam (2002)
17. Wangler, B., Persson, A., Söderström, E.: Enterprise Modeling for B2B integration. In: CD-ROM proceedings International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, , L’Aquila, Italy, August 6-12 (2001)
18. Gustas, R., Bubenko Jr., J.A., Wangler, B.: Goal Driven Enterprise Modelling: Bridging Pragmatic and Semantic Descriptions of Information Systems. In: 5th European - Japanese Seminar on Information Modelling and Knowledge Bases, Sapphoro, May 30-June 3 (1995)
19. Kardasis, P., Loucopoulos, P., Scott, B., Filippidou, D., Clarke, R., Wangler, B., Xini, G.: The use of Business Knowledge Modelling for Knowledge Discovery in the Banking Sector. In: IMACS-CSC 1998, Athens, Greece (October 1998)
20. Carstensen, A., Holmberg, L., Sandkuhl, K.: Supporting Collaboration in an Extended Enterprise with the Connector View on Enterprise Models. In: Proc. of PoEM 2008. LNBIP, vol. 15, pp. 111–126. Springer, Heidelberg (2008)
21. Nilsson, A.G., Tolis, C., Nellborn, C. (eds.): Perspectives on Business Modelling: Understanding and Changing Organisations. Springer, Heidelberg (1999) ISBN 978-3540652496
22. Persson, A.: Enterprise Modelling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Ph.D. thesis, Department of Computer and Systems Sciences, Stockholm University, ISSN 1101-8526 (2001)
23. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)

24. Stirna, J.: The Influence of Intentional and Situational Factors on EM Tool Acquisition in Organisations. Ph.D. Thesis, Department of Computer and Systems Sciences, Royal Institute of Technology and Stockholm University, Stockholm, Sweden (2001)
25. Stirna, J., Persson, A.: An Enterprise Modeling Approach to Support Creativity and Quality in Information Systems and Business Development. In: Halpin, T., Krogstie, J., Proper, E. (eds.) Innovations in Information Systems Modeling: Methods and Best Practices, IGI Global (December 2008),ISBN 978-1-60566-278-7 `http://www.igi-global.com/reference/details.asp?ID=33232`
26. Bergenheim, A., Persson, A., Brash, D., Bubenko, J.A.J., Burman, P., Nellborn, C., Stirna, J.: CAROLUS-System Design Specification for Vattenfall, Vattenfall AB, Stockholm, Sweden (1998)
27. Mikelsons, J., Stirna, J., Kalnins, J.R., Kapenieks, A., Kazakovs, M., Vanaga, I., Sinka, A., Persson, A., Kaindl, H.: Trial Application in the Riga City Council, deliverable D6, project no. IST-2000-28401 Hypermedia and Pattern Based Knowledge Management for Smart Organizations, Riga, Latvia (2002)
28. Dulle, H.: Trial Application in Verbundplan, deliverable D5, IST Programme project IST-2000-28401 Hypermedia and Pattern Based Knowledge Management for Smart Organizations, Verbundplan, Austria (2002)
29. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modelling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007) ISBN 978 -3-540-72987-7
30. Sandkuhl, K., Lillehagen, F.M.: The Early Phases of Enterprise Knowledge Modelling: Practices and Experiences from Scaffolding and Scoping. In: Proc. of PoEM 2008. LNBIP, vol. 15, pp. 1–14. Springer, Heidelberg (2008)
31. Lundqvist, M., Holmquist, E., Sandkuhl, K., Seigerroth, U., Strandesjö, J.: Information Demand Context Modelling for Improved Information Flow: Experiences and Practices. In: Poc. of PoEM 2009. LNBIP, vol. 39, Springer, Heidelberg (2009) ISBN 978-3-642-05351-1
32. Rittel, H.W.J., Webber, M.M. (eds.): Planning Problems are Wicked Problems, Developments in Design Methodology, Cross. John Wiley & Sons, Chichester (1984)
33. Jørgensen, H.D.: Enterprise Modeling - What We Have Learned, and What We Have Not. In: Proc. of PoEM 2009. LNBIP, vol. 39, pp. 3–7. Springer, Heidelberg (2009)
34. Stirna, J., Persson, A.: Anti-patterns as a Means of Focusing on Critical Quality Aspects in Enterprise Modeling. In: Proc. of BMMDS/EMMSAD 2009. LNBIP, vol. 29, pp. 407–418. Springer, Heidelberg (2009) ISBN 978-3-642-01861-9
35. Willars, H.: Business Modeller's Checklist: "Dos" and "Don'ts" in Hands-on Practice. In: Nilsson, A.G., Tolis, C., Nellborn, C. (eds.) Perspectives on Business Modelling: Understanding and Changing Organisations, Springer, Heidelberg (1999)
36. Lindström, C.-G.: Lesson Learned from Applying Business Modelling: Exploring Opportunities and Avoiding Pitfalls. In: Nilsson, A.G., Tolis, C., Nellborn, C. (eds.) Perspectives on Business Modelling: Understanding and Changing Organisations, Springer, Heidelberg (1999)
37. Zavala, A., Hass, B.H.: The Art and Power of Facilitation: Running Powerful Meetings, Management Concepts. Inc., Vienna (2008) ISBN 9781567262124
38. Kaarst-Brown, M.L.: Five symbolic roles of the external consultant – Integrating change, power and symbolism. Journal of Organizational Change Management 12(6), 540–561 (1999)

Author Index