Paul Johannesson
John Krogstie
Andreas L. Opdahl (Eds.)

LNBIP 92

The Practice of Enterprise Modeling

4th IFIP WG 8.1 Working Conference, PoEM 2011
Oslo, Norway, November 2011
Proceedings

ifip

Springer

Lecture Notes in Business Information Processing 92

Paul Johannesson
John Krogstie
Andreas L. Opdahl (Eds.)

The Practice of Enterprise Modeling

4th IFIP WG 8.1 Working Conference, PoEM 2011
Oslo, Norway, November 2-3, 2011
Proceedings

Springer

Volume Editors

Paul Johannesson
Stockholm university
DSV
Forum 100
164 40 Kista, Sweden
E-mail: pajo@dsv.su.se

John Krogstie
Norwegian University of Science and Technology (NTNU)
Department of Computer and Information Science
Sem Sælandsvei 7-9
7491 Trondheim, Norway
E-mail: krogstie@idi.ntnu.no

Andreas L. Opdahl
University of Bergen
Department of Information Science and Media Studies
P.O. Box 7802
5020 Bergen, Norway
E-mail: andreas.opdahl@uib.no

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-24848-1 e-ISBN 978-3-642-24849-8
DOI 10.1007/978-3-642-24849-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011938793

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Enterprise modeling (EM) has gained substantial popularity both in the academic community and among practitioners. A variety of EM methods, approaches, and tools are being developed and offered on the market. In practice they are used for various purposes such as business strategy development, process restructuring, as well as business and IT architecture alignment and governance.

PoEM 2011—the 4th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling—took place in November 2011 in Oslo, Norway. The conference series is a dedicated forum where the use of EM in practice is addressed by bringing together researchers, users, and practitioners in order to develop a better understanding of the practice of EM, to contribute to improved EM practice, as well as to share knowledge and experiences. PoEM 2011 attracted 38 submissions with authors from 17 different countries (Australia, Belgium, Brazil, Canada, France, Germany, India, Italy, Latvia, Luxembourg, The Netherlands, Norway, Russia, Spain, Sweden, Switzerland, USA), out of which the Program Committee selected 18 high-quality papers. Among the authors of these papers we find both researchers and practitioners. The resulting program reflects the fact that the topic of EM encompasses human, organizational issues, as well as more technical aspects related to the development of information systems. The program was organized in five thematic sessions:

- Process Modeling
- Business Modeling
- Enterprise Architecture
- Enterprise Modeling
- Model-Driven Development

The program also featured two keynotes. One by Harald Wesenberg, of Statoil, Norway, an experienced EM practitioner discussing the use of enterprise modeling in an agile world. The second keynote was by Wil van der Aalst on intra- and inter-organizational process mining, on how to discover processes within and between organizations Following the positive experiences from earlier PoEM-conference, the program also included a joint working session where researchers and practitioners had the opportunity to discuss emerging issues in the field of EM practice.

We devote a special thanks to the members of the international Program Committee for promoting the conference and for providing excellent reviews of the submitted papers. Their dedicated work was vital for putting together a

high-quality working conference. We also thank the external reviewers. Special thanks go to the Norwegian University of Science and Technology and the Norwegian Computing Society (Dataforeningen) for supporting the organization of the conference. The PoEM 2011 organizers would also like to thank the conference sponsors—Oracle Norway and the Research Council of Norway.

August 2011

Paul Johannesson
John Krogstie
Andreas L. Opdahl

Organization

PoEM Steering Committee

Anne Persson	University of Skövde, Sweden
Janis Stirna	University of Stockholm, Sweden

PoEM 2011 Organizers

John Krogstie	NTNU, Norway
Paul Johannesson	Stockholm University, Sweden
Andreas L. Opdahl	University of Bergen, Norway
Naci Akkök	Oracle, Oslo, Norway
Steinar Carlsen	Computas, Norway
Håvard D. Jørgensen	Committment, Norway

PoEM 2011 Program Committee

Marko Bajec	University of Ljubljana, Slovenia
Giuseppe Berio	University of Turin, Italy
Rimantas Butleris	Kaunas University of Technology, Lithuania
Steinar Carlsen	Computas, Norway
Wolfgang Deiters	Fraunhofer ISST, Germany
Mathias Ekstedt	KTH, Sweden
Owen Eriksson	Dalarna University, Sweden
Xavier Franch	Universitat Politécnica de Catalunya, Spain
Mounira Harzallah	University of Nantes, France
Patrick Heymans	University of Namur, Belgium
Stijn Hoppenbrouwers	Radboud University Nijmegen, The Netherlands
Jarl Höglund	Allmentor, Sweden
Jon Iden	Norges Handelshøyskole, Bergen, Norway
Håvard D. Jørgensen	Committment, Norway
Robert Lagerström	KTH, Sweden
Mikael Lind	University College Borås, Sweden
Pericles Loucopoulos	Loughborough University, UK
Raimundas Matulevicius	University of Tartu, Estonia
Graham McLeod	PROMIS Solutions AG, South Africa
Christer Nellborn	Nellborn Management Consulting AB, Sweden
Björn Nilsson	Anatés AB, Sweden
Sietse Overbeek	Delft University of Technology, The Netherlands

Oscar Pastor	Valencia University of Technology, Spain
Anne Persson	University of Skövde, Sweden
Michaël Petit	University of Namur, Belgium
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, France
Matti Rossi	Aalto University, Finland
Kurt Sandkuhl	University of Rostock, Germany
Ulf Seigerroth	Jönköping 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 & Young SIA, Latvia
Patrick van Bommel	Radboud University Nijmegen, The Netherlands
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania
Carson Woo	University of British Columbia, Canada

PoEM 2011 Additional Reviewers

Quentin Boucher
Ulrik Franke
Nicolas Genon
Kestutis Kapocius
Johan Knig
Algirdas Laukaitis
Liv Marcks Von Würtemberg
Waldo Rocha Flores
Sam Solaimani

Table of Contents

Enterprise Architecture

Enterprise Modeling

Model-Driven Development

Intra- and Inter-Organizational Process Mining: Discovering Processes within and between Organizations

Wil M.P. van der Aalst

Eindhoven University of Technology, P.O. Box 513,
NL-5600 MB, Eindhoven, The Netherlands
w.m.p.v.d.aalst@tue.nl,
vdaalst.com

Abstract. Due to the availability of more and more event data and mature process mining techniques, it has become possible to discover the actual processes within an organization. Process mining techniques use event logs to automatically construct process models that explain the behavior observed. Existing process models can be validated using conformance checking techniques. Moreover, the link between real-life events and model elements allows for the projection of additional information onto process models (e.g., showing bottlenecks and the flow of work within an organization). Although process mining has been mainly used within individual organizations, this new technology can also be applied in cross-organizational settings. In this paper, we identify such settings and highlight some of the challenges and opportunities. In particular, we show that cross-organizational processes can be partitioned along two orthogonal dimensions. This helps us to identify relevant process mining challenges involving multiple organizations.

Keywords: process mining, cross-organizational mining, business process management.

1 Process Mining

We have applied process mining in over 100 organizations [4]. Our experiences show that process mining is a new and exiting technology that can be applied in a variety of domains (healthcare, governments, banking, insurance, education, retail, production, transportation, high-tech systems, etc.). However, lion's share of today's process mining projects are conducted within a single organization, whereas many processes are distributed over multiple organizations and different organizations are executing similar processes. Therefore, this paper aims to describe the various cross-organizational settings where process mining can be used. Before doing so, we provide a brief overview of the state-of-the-art in process mining.

Process mining provides a new means to improve processes in a variety of application domains. There are two main drivers for this new technology. On

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 1–11, 2011.

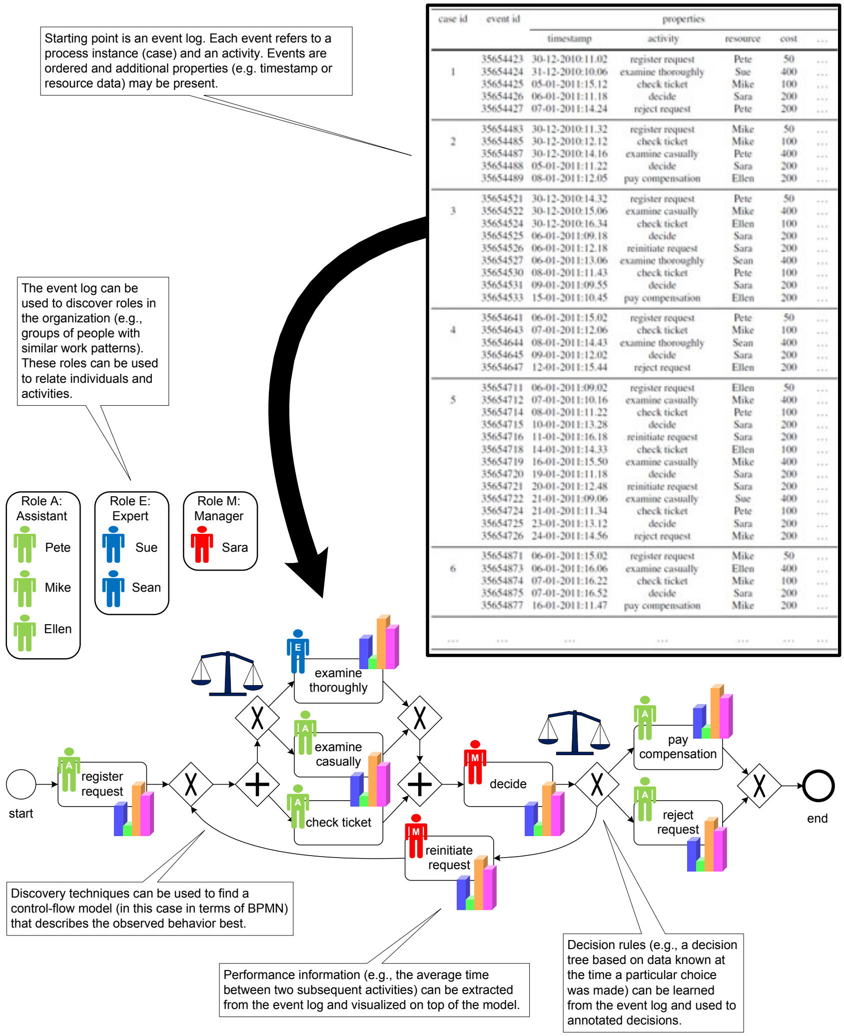


Fig. 1. Process mining techniques extract knowledge from event logs in order to discover, monitor and improve processes [4]

the one hand, more and more events are being recorded thus providing detailed information about the history of processes. On the other hand, there is a need to improve and support business processes in competitive and rapidly changing environments.

Process mining is a relative young research discipline that sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand. The idea of process mining is to *discover, monitor and improve real processes* (i.e., not assumed processes) *by extracting knowledge from event logs* readily available in today's systems (see Fig. 1). Note that process mining includes (automated) process discovery (extracting process models from an event log), conformance checking (monitoring deviations by comparing model and log), social network/organizational mining, automated construction of simulation models, case prediction, and history-based recommendations. Process mining provides an important bridge between data mining and business process modeling and analysis. Over the last decade, event data have become readily available and process mining techniques have matured. Moreover, process mining algorithms have been implemented in various academic and commercial systems. Today, there is an active group of researchers working on process mining and it has become one of the "hot topics" in Business Process Management (BPM) research. Moreover, there is a huge interest from industry in process mining. More and more software vendors started adding process mining functionality to their tools. Examples of software products with process mining capabilities are: ARIS Process Performance Manager, Enterprise Visualization Suite, Interstage BPME, OKT Process Mining suite, Process Discovery Focus, ProcessAnalyzer, ProM, Rbminer/Dbminer, Reflect|one, and Reflect.

Starting point for process mining is an *event log*. All process mining techniques assume that it is possible to *sequentially* record *events* such that each event refers to an *activity* (i.e., a well-defined step in the process) and is related to a particular *case* (i.e., a process instance). Event logs may store additional information about events. In fact, whenever possible, process mining techniques use extra information such as the *resource* (i.e., person or device) executing or initiating the activity, the *timestamp* of the event, or *data elements* recorded with the event (e.g., the size of an order).

Basically, there are three types of process mining. The first type of process mining is *discovery*. A discovery technique takes an event log and produces a model without using any a-priori information. Process discovery is the best-known process mining technique. For many organizations it is surprising that existing techniques are able to discover the real process based on the example executions in the event log. The second type of process mining is *conformance*. Here, an existing process model is compared with an event log of the same process. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. The third type of process mining is *enhancement*. Here, the idea is to extend or improve an existing process model using information about the actual process recorded in some event log. Whereas conformance checking measures the alignment between model and reality, this third type of process mining aims at changing or extending the a-priori model. For instance, by using timestamps in the event log one can extend the model to show bottlenecks, service levels, throughput times, and frequencies.

Process mining may cover different perspectives. The *control-flow perspective* focuses on the control-flow, i.e., the ordering of activities. The goal of mining this perspective is to find a good characterization of all possible paths, e.g., expressed in terms of a Petri net or some other notation (e.g., EPCs, BPMN, and UML ADs). The *organizational perspective* focuses on information about resources hidden in the log, i.e., which actors (e.g., people, systems, roles, and departments) are involved and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show the social network. The *case perspective* focuses on properties of cases. Obviously, a case can be characterized by its path in the process or by the originators working on it. However, cases can also be characterized by the values of the corresponding data elements. For example, if a case represents a replenishment order, it may be interesting to know the supplier or the number of products ordered. The *time perspective* is concerned with the timing and frequency of events. When events bear timestamps it is possible to discover bottlenecks, measure service levels, monitor the utilization of resources, and predict the remaining processing time of running cases.

Moreover, process mining can be used in *online* and *offline* settings. The results of process mining may be used to reason *about* processes (redesign) and to make decisions *inside* processes (operational support).

For a more comprehensive introduction to process mining, we refer to [4].

2 Intra- and Inter-Organizational Processes

Although most applications of process mining have been conducted inside a particular organization, there is no foundational reason why the technology cannot be applied across different organizations. Of course there may be issues related to confidentially, privacy, and data heterogeneity. In this paper we abstract from such problems and simply explore the possibilities of intra- and inter-organizational process mining. For this purpose, we consider two basic settings: (a) *collaboration* and (b) *exploiting commonality.*

In a collaborative setting, different organizations work together to handle process instances. A *process instance*, often referred to as *case*, corresponds to the "thing" that needs to be handled (e.g., a customer placing an order, a patient having a decease that needs to be treated, or a citizen applying for a building permit). The work associated to a case may be distributed over different organizations in a collaborative setting.

In the other basic setting (i.e., exploiting commonality) there are different organizations essentially doing the same thing. For example, there are 430 Dutch municipalities handing out building permits. Here the goal is not to distribute the work associated to a case as different organizations can do (more-or-less) the same thing. Organizations that have processes in common may be in competition, however, they can also learn from one another and share experiences and infrastructures. For example, Dutch municipalities are not competing with respect to handing out building permits. Although they may be competing for new citizens, they can still share a common IT infrastructure and share experiences to better (e.g. faster or more efficient) handle requests for building permits.

2.1 Collaboration: Distributing the Work among Different Organizations

First of all, we consider the collaborative setting where different organizations work together to handle process instances. This requires that the different parties are able to inter-operate, i.e., coordinate their activities. In [1], we identified five forms of *interoperability*. These are depicted in Fig. 2 and described next.

- The first form of interoperability is *capacity sharing*. This form of interoperability assumes centralized control, i.e., the routing of cases is under the control of a single organization. The execution of tasks is distributed, i.e., resources of different organizations may execute tasks.
- The second form of interoperability is *chained execution*: the process is split into a number of disjoint subprocesses which are executed by organizations in a sequential order. This form of interoperability requires that a partner transfers or initiates the flow for a case after completing all the work. In contrast to capacity sharing, the control of the workflow is distributed over the different organizations.
- The third form of routing is *subcontracting*. In this setting, one organization subcontracts subprocesses to other organizations. Consider for example Fig. 2(c) where two subprocesses are subcontracted. For the top-level organization the two subcontracted subprocesses appear to be atomic. For the two organizations executing subcontracted work, the subprocesses can be very complex. Note that the control is hierarchical, i.e., although there is a top-level actor, control is distributed in a tree-like fashion.
- The fourth form of interoperability is *case transfer*. Each organization has a copy of the same process description, i.e., the process specification is replicated. However, at any time, each case resides at exactly one location. Cases (i.e., process instances) can be transferred from one organization to another. A case can be transferred to balance the workload or because tasks are not implemented at all organizations. Note that in Fig. 2(d) it is essentially assumed that each of the organizations uses the same process definition (although some may implement only a part of it).
- The last form of interoperability is shown in Fig. 2(e): *loosely coupled*. For this form of interoperability the process is cut in pieces which may be active concurrently. Moreover, the definition of each of the subprocesses is local, i.e., the environment does not need to know the process. Only the protocol which is used to communicate is public for the other parties involved.

Note that chained execution and subcontracting can be seen as loosely coupled processes. One can think of such processes as "jigsaw puzzles", i.e., the overall process is cut into parts that fit well together. Case transfer uses a different kind of partitioning: cases rather than process fragments are partitioned. Capacity sharing is the only form of interoperability which does not require some partitioning of the process and its instances. We will not consider this for of interoperability as conventional process mining techniques can be used.

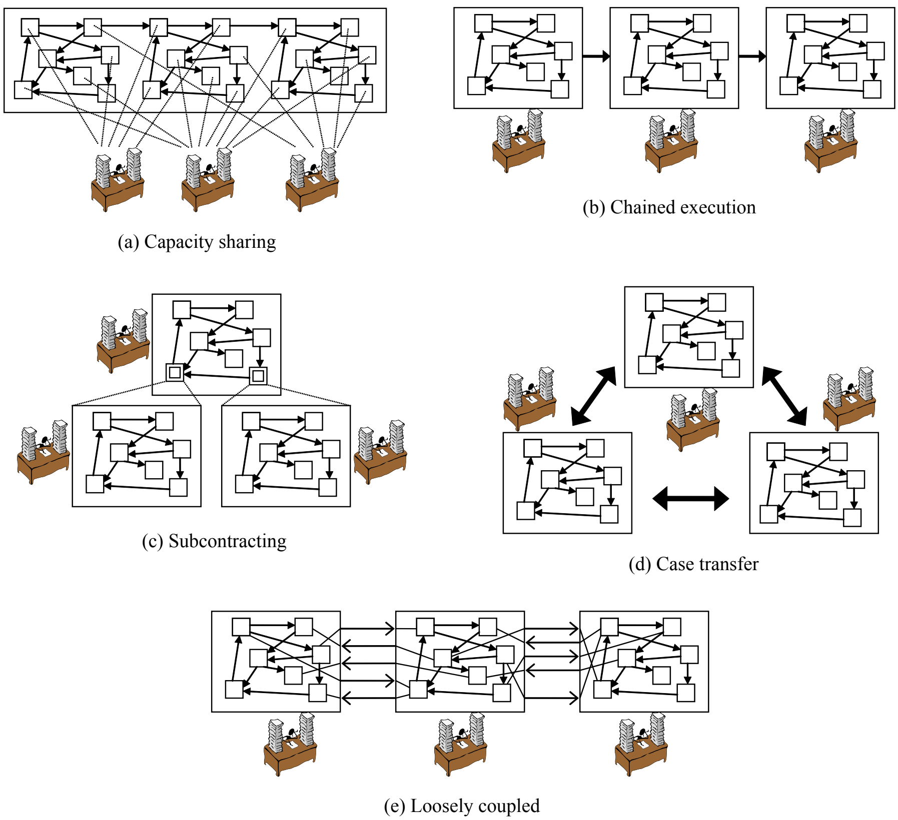


Fig. 2. Different ways of distributing work in a collaborative setting [1]

2.2 Exploiting Commonality: Sharing Knowledge and Infrastructures

As indicated earlier we consider two basic settings: (a) collaboration (cf. Fig. 2) and (b) exploiting commonality. Now we focus on the latter one. This type of cross-organizational processes does not involve interoperability, i.e., there is no explicit distribution of work. Instead, *organizations are executing essentially the same process while sharing experiences, knowledge or a common infrastructure*. To better understand such cross-organizational processes, we consider some examples taken from [2,3,7].

- There are about 430 *municipalities* in The Netherlands. In principle, they all execute variants of the same set of processes. For example, they all support processes related to building permits, such as the process handling applications for permits and the process for handling objections against such permits.
- *Suncorp* is the largest Australian insurance group. The Suncorp Group offers various types of insurance using brands such as Suncorp, AAMI, APIA, GIO,

Just Car, Bingle, Vero, etc. There are insurance processes related to different types of risks (home, motor, commercial, liability, etc.) and these processes exist for the different Suncorp brands. Hence, there are up to 30 different variants of the process of handling an insurance claim at Suncorp.
– *Hertz* is the largest car rental company in the world with more than 8,000 locations in 146 countries. All offices of Hertz need to support the same set of processes, e.g., how to process a reservation. However, there are subtle differences among the processes at different locations due to regional or national variations. For example, the law in one country or the culture in a particular region forces Hertz to customize the standard process for different locations.
– The sales processes of many organizations are managed and supported by *Salesforce*. On the one hand, these organizations share an infrastructure (processes, databases, etc.). On the other hand, they are not forced to follow a strict process model as the system can be configured to support variants of the same process.
– *Easychair* supports the review processes of many conferences. On the one hand, conferences share common functionality and processes. On the other hand, many variations are possible.

Organizations such as Suncorp and Hertz need to support many variants of the same process (*intra-organizational variation*). Different municipalities in a country need to offer the same set of services to their citizens, and, hence, need to manage similar collections of processes. However, due to demographics and political choices, municipalities are handling things differently. Sometimes these differences are unintentional; however, often these differences can be easily justified by the desired "Couleur Locale" (*inter-organizational variation*).

The cross-organizational processes mentioned above refer to a different type of cooperation than the different ways of distributing work depicted in Fig. 2. Organizations can learn from one another. For example, one municipality may improve its processes by learning from experiences of a better performing municipality. Moreover, if there a sufficient commonalities, organizations may want to share *configurable* processes and infrastructures [2,3,7].

2.3 Horizontal and Vertical Partitioning

After discussing the two basic forms of cross-organizational processes (collaboration and exploiting commonality), we conclude that there are two partitioning dimensions: the *case* dimension and the *process* dimension. *Vertical partitioning* uses the case dimension to partition work, i.e., the cases are distributed over several organizations but the process is not cut into pieces. *Horizontal partitioning* is based on the process dimension, i.e., the process is cut into pieces and organizations are responsible for specific parts of the jigsaw puzzle. The partitioning dimensions are in principle orthogonal but combinations are possible.

Chained execution, subcontracting, and loosely coupled, as described using Fig. 2, correspond to horizontal partitioning. Case transfer (Fig. 2(d)) and exploiting commonality (Section 2.2) correspond to vertical partitioning. Figure 3

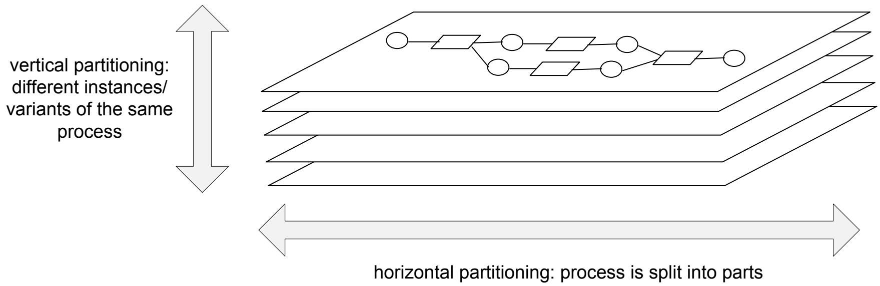


Fig. 3. Two partitioning dimensions: (a) horizontal partitioning and (b) vertical partitioning

illustrates these two partitioning dimensions. Traditionally, process mining has been focusing on processes that are not partitioned, i.e., all process instances belong to the same monolithic process. As will be shown in the reminder, the two dimensions shown in Fig. 3 can be used to structure the different process mining challenges.

3 Challenges for Process Mining

In Section 1 we introduced process mining as a new technology to analyze operational processes based on the footprints they leave in event logs. Subsequently, we provided a classification of intra- and inter-organizational processes in Section 2. Based on this we identified two main partitioning dimensions as shown in Fig. 3. These two partitioning dimensions serve as the basis for discussing various process mining challenges.

3.1 Horizontal Partitioning

A process that is partitioned horizontally can be seen as a jigsaw puzzle. Each "puzzle piece" corresponds to a fragment of the overall process and is under the control of one organization. In the classical setting (i.e., single process/organization) it is possible to capture all events relevant for a particular process, e.g., one can extract data from an SAP system clearly showing all steps taken in a particular process. However, when a process is partitioned horizontally one cannot assume this. An organization can only see some of the "puzzle pieces". It can see all events related to the puzzle pieces it is responsible for. Moreover, it can see the interactions with other puzzle pieces.

In a horizontally partitioned process there needs to be interaction between the different pieces. Typically, messages are exchanged between the puzzle pieces controlled by different organizations. Consider for example SOAP or EDI messages. SOAP (Simple Object Access Protocol) is a protocol specification for exchanging messages between web services. It uses a rather generic XML format which specifies a SOAP envelope consisting of a header and body.

Electronic Data Interchange (EDI) standards such as UN/EDIFACT (United Nations/Electronic Data Interchange For Administration, Commerce and Transport) impose more constraints on the messages being exchanged. The different fields in an EDI message have a predefined meaning. Note that messages can be exchanged in a synchronous or asynchronous manner. In some cases there may also be a party that is able to observe all message exchanges without being able to look "inside the puzzle pieces".

Hence, the main challenge is to conduct process mining *while only seeing a part of the overall process* [5]. Typical questions are:

- How to discover a process model when only seeing message exchanges and/or local events?
- How to check conformance when only seeing message exchanges and/or local events?
- How to identify bottlenecks when only seeing message exchanges and/or local events?
- How to correlate messages to process instances? When sending a message from one organization to another it needs to be routed to the appropriate instance of the process. This is a problem that is often underestimated and most researchers simply abstract from it [6].
- How to deal with many-to-many relationships across different organizations? One customer order may correspond to many order lines that may or may not be combined in different deliveries. Besides the problem of correlating messages there is the problem that one instance in one organization may refer to multiple instances in another organization and vice versa [4].

Since more and more processes are distributed over multiple organizations, it is important to address the above questions.

3.2 Vertical Partitioning

When a process is partitioned vertically, cases are distributed over several organizations each using their own variant of the process. These organizations may collaborate (see case transfer style of interoperability illustrated by Fig. 2(d)) or simply share knowledge and infrastructures. The metaphor of the jigsaw puzzle is not applicable anymore. A better metaphor is the "spot the difference game" children like to play (i.e., looking at two figures to find the differences between both). The bottom line is that there are different events logs all referring to some variant of the same process. The challenge is to *analyze commonalities and differences* between these processes based on the different event logs [2,3,7].

Besides a pair-wise comparison of logs and models, we can also use supervised learning to explain differences. For example, we can use classification techniques such as decision tree learning. For this purpose we need to label the data at the level of cases or at the level of event logs. Classification is based on a selected *response variable* and a set of *predictor variables*. For example, the response variable could be the (average) flow time or costs of a case or log. The fitness of an event log or case with respect to some reference model can also be taken as a response variable. Predictor variables are other properties of cases, event logs, or process models. For example, the complexity of the process model and the number or resources involved. Based on such information one can construct a decision tree that aims to *explain the response variable in terms of predictor variables.* This assists in understanding the essential differences between different organizations. For example, classification based on logs of different municipalities may reveal that (a) larger municipalities tend to have fewer deviations, (b) allowing for more concurrency results in shorter flow times but more deviations, and (c) a pre-check of building permits results in shorter flow times and a higher acceptance rate.

In [7], we provide some initial results obtained in the CoSeLoG project. In this project, 10 of the 430 Dutch municipalities are participating to investigate how process mining, configurable process models, and cloud technology can be used to reduce costs and improve service. All Dutch municipalities need to offer the same services to their citizens, and need to manage similar collections of processes. However, due to demographics and political choices, municipalities are handling things differently. The ten municipalities involved in CoSeLoG are eager to learn "proven best practices" from one another. This can be operationalized using cross-organizational process mining.

4 Conclusion

Although process mining is often applied within the boundaries of individual organizations, there are many process management questions that transcend the level of a single organization. Different organizations need to cooperate to realize a process, share an infrastructure, or may want to learn from one another. However, very few applications of process mining have been documented in literature. Therefore, this paper aims to structure the different cross-organizational settings in which process mining can be applied. Based on this, we highlighted some of the key questions. Currently, we are involved in several research projects that aim to address these questions:

- The EDImine project (`http://edimine.ec.tuwien.ac.at`) seeks to extend current process mining approaches in order to apply them to inter-organizational business processes while building on the additional information provided by traditional Electronic Data Interchange (EDI) standards. Advantages of using EDI technology are its widespread use and standardization of message content.
- The CoSeLoG project (`http://www.win.tue.nl/coselog/wiki/start`) focuses on a particular application domain: Dutch municipalities. Since all of these municipalities need to execute the same collection of processes, it is interesting to analyze differences and commonalities. The goal is to let these municipalities learn from one another and share a common (configurable) infrastructure.
- The ACSI project (`http://www.acsi-project.eu/`) uses artifact-centric modeling approaches to support service collaborations in open business networks. Process mining is used to understand such collaborations and to improve performance.

References

1. van der Aalst, W.M.P.: Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow. Information Systems 24(8), 639–671 (2000)
2. van der Aalst, W.M.P.: Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 8–25. Springer, Heidelberg (2010)
3. van der Aalst, W.M.P.: Business Process Configuration in The Cloud: How to Support and Analyze Multi-Tenant Processes? In: Zavattaro, G., Schreier, U. (eds.) Proceedings of the 9th IEEE European Conference on Web Services (ECOWS 2011). IEEE Computer Society Press, Los Alamitos (2011)
4. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin (2011)
5. van der Aalst, W.M.P., Dumas, M., Ouyang, C., Rozinat, A., Verbeek, H.M.W.: Conformance Checking of Service Behavior. ACM Transactions on Internet Technology 8(3), 29–59 (2008)
6. van der Aalst, W.M.P., Mooij, A.J., Stahl, C., Wolf, K.: Service Interaction: Patterns, Formalization, and Analysis. In: Bernardo, M., Padovani, L., Zavattaro, G. (eds.) SFM 2009. LNCS, vol. 5569, pp. 42–88. Springer, Heidelberg (2009)
7. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards Cross-Organizational Process Mining in Collections of Process Models and their Executions. In: International Workshop on Process Model Collections (PMC 2011). Springer, Berlin (2011)

Business Process Model Similarity as a Proxy for Group Consensus

Peter Rittgen

Vlerick Leuven Gent Management School, Reep1, 9000 Gent, Belgium, and
University of Borås, 501 90 Borås, Sweden
peter.rittgen@vlerick.com

Abstract. Consensus is an important measure for the success of any business process modeling effort. Although intensively studied in the general literature on group processes, consensus has hardly been considered in business process modeling and never seriously measured. We define consensus as the level of agreement of group members' views on the process and introduce business process similarity as a proxy. We validate the measure by comparing it to an existing self-reported measure of consensus.

Keywords: Business process modeling, model similarity, group consensus, mental model, view, visualization.

1 Introduction

The literature on business process modeling is vast and the importance of measuring the success of process modeling projects and sessions has been widely recognized [1-6]. But prevalent success measures for individual modeling sessions primarily involve some form of model quality measure [7-10]. While it is undisputed that the quality of a business process model is relevant to modeling success it is not the only and perhaps not even the most important success factor.

The reason for this is twofold: the process model itself is a social construction, and its purpose is again to support some social process, e.g. a change project or system development project. In other words: the model documents the results of one social process (modeling) and serves as a point of departure for another one.

If the model were to be processed by a computer its quality would be of prime importance to ensure correct interpretation by the machine. But the results that are documented in the model are primarily the mutual knowledge that has been developed in the modeling session, the conflicts that had to be solved on the way, and the consensus that has been achieved among the group members as a result.

It is precisely this consensus that is a prerequisite for people's commitment to the ensuing change project, for example. Often a poor model with high consensus goes further than a good model with little consensus. Hence consensus is a major result that needs to be achieved in business process modeling sessions much like in many other forms of group work.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 12–24, 2011.

But while there is considerable research on consensus in other areas [11-13] the topic received little attention in business process modeling with researchers barely mentioning the issue [14-17] and, to the best of our knowledge, not researching it in a systematic way, let alone measuring consensus.

The purpose of this paper is to develop such a measure. To do so we first define the concept of consensus in the next section, Group consensus in process modeling. For this purpose we rely on cognitive theories of modeling.

Based on the cognitive concept of a view and the model as its externalization we can interpret consensus as "view agreement" and hence as "model similarity". The section Business process model similarity therefore introduces a measure for the latter.

A proper evaluation of a new measure typically establishes validity by comparison to an existing measure of the same concept. The section Other group consensus measures therefore introduces an established measure for group consensus. The actual validation of the new measure was done in field experiments. The set-up of these experiments is described in the section Comparing model similarity and consensus in field experiments.

The section Data analysis reports on the analysis of the data that we collected in the experiments. The results and implications of this analysis are treated in the section Discussion. The paper concludes with a summary of the findings and an outlook on future work.

2 Group Consensus in Process Modeling

Group modeling is a cognitive as well as a group process. We therefore define group consensus in business process modeling as the extent to which the group members' views on the process agree with each other. The problem with this definition is that the views that are entertained by the group members are not directly accessible so the measure of consensus needs to be based on some external representation of these views.

For this purpose we need to resort to the cognitive theory on the modeling process. The foundations for our understanding of model cognition were laid by Johnson-Laird [18, 19] who introduced the idea of so-called mental models that the mind constructs when it imagines a situation. A mental model consists of a mental system of relations that has a structure similar to the system that is imagined.

When the mind engages in the process of deduction it performs the following three steps: comprehension, description, and validation. [20] found that the individual part of the modeling process can be described well in terms of Johnson-Laird's deduction. For our purposes the relevant step is that of description, which proceeds as follows:

1. Build *mental model*
2. Extract *view* from mental model
3. Transcribe view to *visualization*
4. Conduct within-model testing
5. If any test fails, go back to step 2 or 3, possibly modifying the mental model

In short: *mental models* are stored in long-term memory but not directly accessible; to work on them the mind needs to create *views* in working memory that correspond to these mental models. If the work on the views becomes too complex, which is often the case for modeling, external representations have to be created in order to extend the internal working memory. These external representations are called *visualizations* and in business process modeling they typically take the form of process diagrams.

We now come back to the original problem: if group consensus in modeling is the mutual agreement of the views (and thereby of the underlying mental models), then consensus can be measured also in terms of the agreement of the visualizations of the views, which correspond to the views.

So comparing the views of the group members can be replaced by comparing the externalizations of these views, i.e. the business process models. If we want to assess the degree to which two people agree on their views on a process we need to assess how similar the business process models are that visualize their views.

3 Business Process Model Similarity

Measures for business process model similarity have been investigated thoroughly [21-26]. A comprehensive overview can be found in [24]. We use the measure suggested in [24] because it covers most prevalent modeling languages as well as all aspects of model similarity that have been introduced so far in the literature: node matching, structural and behavioral similarity.

The selected measure works for the Business Process Modeling Notation (BPMN), the language which was used by the organizations where we conducted the study. In the following we describe the 3 aspects of model similarity.

Node matching tries to map nodes from the one model to nodes of the other model by comparing the labels, attributes and types of nodes. Types in BPMN are task, start event, end event, parallel gateway, exclusive gateway and so on. Attributes are the swim lanes where the nodes are located, for example, and labels are the names of e.g. activities (“Send invoice”).

Node matching can be effected with semantic or syntactic measures. The latter is based on the string-edit distance, i.e. the number of letters that need to be added, replaced or deleted to transform the label of an activity in one model to that of an activity in the other model. It is useful in the case of spelling mistakes as it is able to identify two words where one is spelled wrong (e.g. confirmation and confrimation), whereas semantic measures consider that as a 0% match. But syntactic measures will also (wrongly) identify words with similar spellings but different meanings, such as e.g. plane and plate.

To avoid such misclassifications we have decided to drop syntactic measures in favor of semantic matching. To ensure that wrong spellings do not introduce faulty mismatches we have spell-checked all models manually.

Semantic matching is based on a database of synonyms. We have used Wordnet for this purpose which is freely available online and contains 147,278 unique words. It will identify labels and attributes with similar meanings (e.g. “Send invoice” and

"Send bill"), which is exactly our purpose as participants in modeling sessions often phrase the same activity in different terms.

Based on the node matching the two graphs can be compared now with the help of structural or behavioral similarity. The former uses only structural information on the graph, i.e. the way in which activities are connected with "arrows" but does not look at their meaning in terms of control flow. Two models are considered structurally equivalent if two nodes are always connected in the same way in one model as their matching counterparts in the other.

Behavioral equivalence looks at the actual execution of the processes described by the models, e.g. bisimulation equivalence. Here two models are considered equivalent if, at any time during process execution, an activity that can be performed in one process can also be performed in the other, and vice versa.

One would expect the latter measure to give a more realistic and reliable account of the similarity of process models as it considers actual behavior. But [24] found that the structural measure performs equally well, i.e. it correctly identifies behaviorally similar and dissimilar processes without looking at behavior.

This is perhaps not surprising as the behavior of a process is largely determined by the structure of the control flow graph and the semantics of the gateways whose similarity is checked on the node level already, and both are accounted for in the structural measure, too.

As structural similarity is equally accurate but computationally more advantageous we have used it in our study. It is based on the concept of graph-edit distance, i.e. the number of nodes and edges that need to be added, replaced, or deleted in order to turn one model into the other.

The overall similarity measure is the inverse of the number of edit steps in relation to the graph size where nodes are matched by type and semantic similarity of their labels and attributes. It is a number that ranges from 0 to 1 where 0 refers to two process models that have nothing in common and 1 to a perfect match.

More details on the used algorithms can be found in the section Comparing model similarity and consensus in field experiments.

4 Other Group Consensus Measures

Group consensus is considered as an important outcome of a group process and as a consequence there is a large body of literature that provides measurements for it. The majority of these publications are in the area of decision-making where consensus is usually defined as the mutual agreement of decision-makers' preferences concerning a number of decision alternatives. Preferences can be binary, i.e. an alternative is either preferred or not, or linguistic (qualitative), i.e. the preference values are labeled "none", "very low", "low", "medium", "high", "very high", and "perfect". In addition the membership in a category can be strict, i.e. in exactly one category, or fuzzy, i.e. to a certain extent [0…1] in several categories.

It is common to all these approaches that they require both explicit alternatives and preferences. These measures do therefore not easily carry over to modeling where the decision alternatives are rather implicit.

In our search for a suitable and established measure of consensus in business process modeling we have therefore turned our attention to another stream of research where group consensus is viewed in terms of solution satisfaction [27-32].

The purpose of business process modeling is to find a solution either to the problem of accurately representing an existing process, the knowledge about which is distributed over a number of domain experts (AS-IS modeling), or to the problem of creating a new process that overcomes problems of the old one or provides new products or services (TO-BE modeling).

A modeling group exhibits consensus if they are satisfied with the solution that was generated in the modeling session. Solution satisfaction is therefore a suitable measure of group consensus in the context of business process modeling.

We used the measure described in [33] for group decision processes that uses five items rated on a 5-point Likert scale with the anchors “Not at all”, “To a little extent”, “To some extent”, “To a great extent”, and “To a very great extent”. The items run:

1. How satisfied are you with the quality of your group’s solution?
2. To what extent does the final solution reflect your inputs?
3. To what extent do you feel committed to the group solution?
4. To what extent are you confident that the group solution is correct?
5. To what extent do you feel personally responsible for the correctness of the group solution?

Using a measure for group decision processes in modeling is appropriate because the group modeling process can be conceptualized as a process where decisions on the development of the model are negotiated [34]. Moreover, the measure has been validated by its authors.

The group consensus measure is also a self-reported measure and therefore suited for comparison with the model similarity measure, which is based on observation. This means that we can compare the agreement as perceived by the participants with the actual agreements of their views as “objectively” measured according to the previous section.

5 Comparing Model Similarity and Consensus in Field Experiments

We set up field experiments to test the validity of business process model similarity as a proxy for group consensus in process modeling. For this purpose we asked members of process modeling groups to draw up a model of the business process **after** the modeling session.

Participants had to prepare the after models without access to the group model. In half of the cases the after model was done immediately after the session, in the other half one week later to control for group model bias (see section Data analysis).

We also varied the order of drawing the after models and applying the consensus measure. In both cases, immediate and one week later, half of the participants were drawing the models after and half of them before the consensus measure was applied. Again a more detailed discussion can be found in the Data analysis section.

Each after model was compared to the group model by computing the business process model similarity measure introduced in section Group consensus in process modeling. The formulae we used are (all from [24]):

- Definition 14 (Graph-edit distance similarity)
- Definition 13 (Graph-edit distance)
- Definition 12 (Node matching similarity)
- Definition 7 (Attribute similarity, *Simattr*)
- Definition 6 (Semantic similarity, *Simsem*)

The similarity of two nodes n_1 and n_2 is defined as (τ = type of node):

$$Sim(n_1, n_2) = \begin{cases} 0, & \tau_1 \neq \tau_2 \\ \frac{1}{2} \cdot Simsem(n_1, n_2) + \frac{1}{2} \cdot Simattr(n_1, n_2), & \tau_1 = \tau_2 \end{cases}$$

We also asked participants to fill in the questionnaire related to group consensus. Participants had to rate the five questions with respect to the final model that was adopted by the group. For this purpose they were shown the group model to facilitate the answering of the questions.

We conducted three half-day modeling workshops at each of the following five organizations: the insurance branch of a large bank, an IT solution provider in the automobile industry, a psychiatric hospital, a bio-engineering laboratory, and a city administration. All workshops revolved around the development of a TO-BE model for a business process or a substantial part of it.

In total we conducted 15 modeling sessions with 6-12 group members and a total of 122 participants.

6 Data Analysis

In order to assess whether business process model similarity is an acceptable proxy for consensus on the group model we compared the consensus measure to the model similarity measure. We assumed that agreement with the group model is a result of the agreement of an individual's view on the process with the group process model and that the individual view is accurately represented in the externalization of this view as manifested in the visualization created by this individual (the after model).

If this assumption is true then we should see a high correlation between the items C1 to C5 of the self-reported consensus measure and the similarity degree objectively measured between the after model and the group model.

We have first subjected the data to a Kolmorov-Smirnov test which was significant on the 1% level for all six variables (C1-C5 and Similarity). We can therefore assume a normal distribution of the data and make use of parametric correlation analysis based on bivariate Pearson correlation coefficients.

The results of the correlation analysis are shown in Table 1. All correlations are significant on the 1% level.

Table 1. Pearson correlations for consensus items and business process model similarity

	Consensus				
	C1	C2	C3	C4	C5
Similarity	0.822	0.720	0.734	0.777	0.758

On a side note it should be mentioned that the inter-item correlations are also both significant and strong for all item pairs which provides some further validation of the consensus instrument.

The coefficients are reasonably high which confirms that the self-reported level of consensus (individual agreement with the group result) is indeed closely linked to the agreement of the individual view on the process and the group view as measured by the similarity between the individual model and the group model.

A potential threat to validity might be that the knowledge about the group model might have influenced people in drawing the after models. This phenomenon is called "group thinking" and people exhibit this behavior to conform to group pressure either knowingly or unknowingly.

We took three measures to counteract this possible bias. First we told participants explicitly that the model they draw should reflect their own ideas about a good TO-BE model. We warned them not to try and copy the group model, or what they remembered of it.

A second measure was that we asked people in half of the sessions to draw their diagram a week after the session so that the chances of remembering the group model were quite low. Table 2 shows the correlations and Table 3 the averages and standard deviations in both cases.

Table 2. Correlations of similarity with consensus items after session and 1 week later

Time of drawing	C1	C2	C3	C4	C5
After session	0.835	0.762	0.715	0.775	0.775
1 week later	0.793	0.650	0.751	0.753	0.752

Table 3. Averages and standard deviations after session and 1 week later

Item	After session		1 week later	
	Mean	Std. deviation	Mean	Std. deviation
C1	3.31	1.05	2.90	1.13
C2	3.19	1.03	2.83	1.07
C3	3.06	1.09	2.73	1.28
C4	3.16	1.00	2.70	1.16
C5	3.18	1.14	2.95	1.03
Similarity	0.55	0.22	0.43	0.34

The correlations are slightly higher for all items except C3 when measurements are made immediately after the sessions. This means that the link between similarity and consensus is stronger in the beginning and weakens over time. An explanation can be found in the increased standard deviations, especially in the case of similarity, that indicate a greater variety in the data that was measured after one week. This shows that time did indeed have an impact on memory as the drawing used for the similarity measure requires an accurate recall of the view on the business process.

The consensus values have decreased by only 11% on average but the similarity values have gone down by 22%. This means that people can apparently remember the level of agreement that was achieved better than the details of the models, both their own and that of the group.

Another bias could be the fact that the group model is shown to the participants to facilitate the answering of the consensus questions. If the drawing is done after the consensus measure is applied, they might use parts of the recently seen group model in their own drawing.

To see whether this is the case we have let people do the drawing both before and after the consensus measure. This is the third measure we used to control for group thinking bias. Table 4 shows the results.

Table 4. Average consensus and similarity depending on order of measure application

Consensus measure	Average consensus	Average similarity
After drawing	2.8	0.49
Before drawing	3.2	0.49

Group thinking is clearly not at play here as it would have increased the similarity values in the groups that did the consensus measure before the drawing because they might have used some information from the group model shown during questioning in the drawing.

On the other side, the consensus values are lower in the groups that answered the consensus questions after the drawing. This is probably due to the fact that they had better knowledge of their own view after expressing it explicitly in a model and could therefore remember more distinctions between their own view and the group model.

7 Discussion

The fact that there is a strong correlation between the similarity measure and all items of the consensus measure shows that the perceived agreement with the group model is closely related to the degree in which the group model coincides with the individual perception of the process.

Note that all process models were TO-BE models so agreement is much harder to achieve than in AS-IS modeling because different personal interests are at stake. As a consequence the average consensus value was not very high after only one half-day session with an average 3 on the 1-5 Likert scale and 0.49 on model similarity.

But nevertheless the correlation was strong both in the case of little consensus and in the case of good consensus. The maximum consensus was a 4.6 on the Likert scale and 0.89 on similarity, the minimum 1.6 and 0.08, respectively.

As a consequence of our results business process model similarity is a valid proxy for measuring individual agreement with the group result in business modeling of processes. The similarity measure can also be extended to capture the overall group consensus.

For this purpose it is useful to know that model similarity is actually measured as the inverse of model distance, i.e. the effort required to turn one model into the other. If we have *n* models that reflect the views of *n* group members we first compute all pairwise distances and then determine the model M_j with the lowest sum of incident distances. The effort to turn all other models into this one is the lowest possible effort to make all models agree. It can be calculated as:

$$dist(M_1,\ldots,M_n)=\frac{\sum_{i,i\neq j} dist(M_i,M_j)}{(n-1)\cdot size(M_j)+\sum_{i,i\neq j} size(M_i)}$$

The similarity is then just the distance subtracted from one. This measure delivers a number between zero and one for any number of models and can be seen as the overall similarity between them, i.e. the inverse of the minimum effort required to make all the models equal.

This figure also tells us how far or close the individual views are from each other and is therefore a measure of the group consensus. Please observe that this measure does not require a group model and can therefore be used at any stage of modeling. This is important as consensus can otherwise only be measured after a group model is available.

To use model similarity as a proxy for consensus also has the advantage that it can be measured in a much more objective manner. Other consensus measures have to rely on self-reported perceptions whose measurement is notoriously subjective and inexact.

Model similarity, on the other hand, is an exact measure that accurately portrays in how far the view of an individual as expressed in the individual model, really agrees with the group model, or in how far the group members' views agree with each other. The only drawback is that the drawing of a model takes more time than the answering of a few questions.

But this disadvantage becomes obsolete when modeling is done in a collaborative way where individual models are produced as a by-product. This form of modeling is increasingly becoming the rule as can be seen in the growing number of publications that advocate collaborative modeling, participatory modeling, or end-user modeling [35-38]. In these scenarios the participants in group modeling take on an active role in model creation.

They draw models themselves, e.g. with the help of a computerized tool and the models, or rather proposals are then viewed and commented by other group members which leads to revisions of these proposals and hence a second round of drawing. This process is not only a revision process but also a collective learning process [39] where participants learn to understand others' views and adapt their own views accordingly.

But beyond this modeling is also a negotiation process [34] where individuals have to make concessions to move towards an agreement. In collaborative modeling they can do so by incorporating features that are of interest to others into their proposals thereby making them more attractive and acceptable for others.

Over time initially different proposals will converge and a consensus group model will emerge. In the meantime the similarity between the proposals at each stage of model development will give us a precise understanding of the level of consensus that has already been achieved without the necessity for instruments that require additional inputs from the participants and that rely on perceptions only.

Each modeling session can be evaluated by comparing the similarity of models before and after the session to see whether consensus has been increased during the session. Model similarity can also be used as a tool of modeling project management by specifying a minimum consensus level that is required. A modeling project is only considered closed after this level has been reached. This makes sense because the success of follow-up projects often depends on the achievement of a satisfactory level of agreement in modeling.

If a session cannot increase consensus the facilitator can use conflict resolution techniques to resolve the issues before proceeding with the constructive part of the modeling session. Similarity measures can hence also be used to detect situations of serious conflict and guide session management.

8 Conclusion and Outlook

We have suggested the use of business process model similarity as a proxy for group consensus in business process modeling. We have shown that this proxy is a reliable measure of group consensus as compared to an established self-reported consensus measure. The proxy is also a more objective and accurate measure and has therefore the potential to outperform self-reported measures because it eliminates the group thinking bias usually associated with self-reported measures of group performance. It does also not rely on the existence of a group model and can therefore be used at any stage in the modeling process.

The scenarios for using this proxy are manifold and the Discussion section has shown some of them: evaluation of modeling session performance, modeling project management, conflict detection, and so on. But the real advantage lies in integrating this measure into new methods for collaborative business process modeling where the progress towards a consensus model can actually be measured and controlled.

This will allow us to organize modeling in such a way that the steps in the method really lead to an improvement in consensus so that the success of modeling sessions can actually be planned. Because of the collaborative nature of these sessions, the individual group members, or perhaps small teams, will generate model proposals anyway so that additional drawing of models (as in our experiments) will no longer be necessary.

A continuous assessment of the status quo can therefore be made at any time in the modeling project as the computation of model similarity can be done automatically once the views of individuals or teams are available as models. An evaluation of the similarity between views on the business process can hence be the driver for the whole modeling effort.

This opens up possibilities for developing a new range of consensus-driven, or consensus-oriented business process modeling methods and supporting tools for the creation, maintenance, review, revision, and integration of model proposals and their convergence to a consensus model for the group.

Beyond this the further evolution of the model after implementation can also be supported in a consensus-oriented and decentralized way.

While all this is still hypothetical the assessment and control of consensus in business process modeling is a relevant issue already today. We believe that the measurement of consensus in an objective way is an important key to solving the consensus-related issues in process modeling.

References

1. Dennis, A.R., Carte, T.A., Kelly, G.G.: Breaking the rules: success and failure in groupware-supported business process reengineering. Decision Support Systems 36, 31–47 (2003)
2. Lu, R., Sadiq, S.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
3. Rosemann, M.: Potential pitfalls of process modeling: part A. Business Process Management Journal 12, 249–254 (2006)
4. Sedera, W., Gable, G.G., Rosemann, M., Smyth, R.W.: A success model for business process modeling: findings from a multiple case study. In: Eighth Pacific Asia Conference on Information Systems. Association for Information System (2004)
5. Luo, W., Tung, Y.A.: A framework for selecting business process modeling methods. Industrial Management & Data Systems 99, 312–319 (1999)
6. Recker, J.C., Rosemann, M., Indulska, M., Green, P.: Business process modeling: a comparative analysis. Journal of the Association for Information Systems 10, 333–363 (2009)
7. Dean, D.L., Orwig, R.E., Lee, J.D., Vogel, D.R.: Modeling with a group modeling tool: Group support, model quality, and validation. In: Proceedings of the 27th Hawaii International Conference on System Sciences, Information systems: Collaboration technology, organizational systems and technology, vol. 4, pp. 214–223. IEEE Computer Society Press, Los Alamitos (1994)
8. Mendling, J., Recker, J.: Extending the Discussion of Model Quality: Why Clarity and Completeness may not always be enough. In: 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), pp. 109–121 (2007)
9. Moody, D.L., Sindre, G., Brasethvik, T., Sølvberg, A.: Evaluating the quality of information models: empirical analysis of a conceptual model quality framework. In: Dillon, L., Tichy, W. (eds.) Proceedings of the International Conference on Software Engineering, ICSE 2003, Portland, USA (2003)

10. Sánchez-González, L., García, F., Mendling, J., Ruiz, F., Piattini, M.: Prediction of Business Process Model Quality Based on Structural Metrics. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 458–463. Springer, Heidelberg (2010)
11. DeStephen, R., Hirokawa, R.Y.: Small Group Consensus. Small Group Research 19, 227–239 (1988)
12. Priem, R.L.: Top management team group factors, consensus, and firm performance. Strategic Management Journal 11, 469–478 (1990)
13. Yoo, Y., Alavi, M.: Media and Group Cohesion: Relative Influences on Social Presence, Task Participation, and Group Consensus. MIS Quarterly 25, 371–390 (2001)
14. Clegg, B.T.: Building a Holarchy Using Business Process-Oriented Holonic (PrOH) Modeling. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 37, 23–40 (2007)
15. Decker, B., Rech, J., Althoff, K.-D., Klotz, A., Leopold, E., Voss, A.: eParticipative Process Learning–process-oriented experience management and conflict solving. Data & Knowledge Engineering 52, 5–31 (2005)
16. Kumarapeli, P., De Lusignan, S., Ellis, T., Jones, B.: Using Unified Modelling Language (UML) as a process-modelling technique for clinical-research process improvement. Informatics for Health and Social Care 32, 51–64 (2007)
17. Rittgen, P.: Quality and perceived usefulness of process models. In: Proceedings of the 2010 ACM Symposium on Applied Computing. ACM, Sierre (2010)
18. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Harvard University Press, Cambridge (1983)
19. Johnson-Laird, P.N.: The Computer and the Mind. Harvard University Press, Cambridge (1988)
20. Waisel, L.B., Wallace, W.A., Willemain, T.R.: Visualizing modeling heuristics: an exploratory study. In: Proceedings of the 20th international conference on Information Systems. Association for Information Systems, Charlotte (1999)
21. Dijkman, R.: Diagnosing Differences between Business Process Models. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 261–277. Springer, Heidelberg (2008)
22. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
23. Dijkman, R., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning Business Process Models. In: 2009 IEEE International Enterprise Distributed Object Computing Conference (EDOC 2009), pp. 45–53 (2009)
24. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36, 498–516 (2010)
25. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling, vol. 67, Australian Computer Society, Inc., Ballarat (2007)
26. van Dongen, B., Dijkman, R., Mendling, J.: Measuring Similarity between Business Process Models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
27. Dennis, A.R., Aronson, J.E., Heninger, W.G., Walker II, E.D.: Structuring Time and Task in Electronic Brainstorming. MIS Quarterly 23, 95–108 (1999)

28. Dennis, A.R., Hilmer, K.M., Taylor, N.J.: Information Exchange and Use in GSS and Verbal Group Decision-Making: Effects of Minority Influence. Journal of MIS 14, 61–88 (1997)
29. Dennis, A.R., Wixom, B.H.: Investigating the Moderators of the Group Support Systems Use with Meta-Analysis. Journal of MIS 18, 235–258 (2001)
30. Ocker, R., Fjermestad, J., Hiltz, S.R., Johnson, K.: Effects of Four Modes of Group Communication on Outcomes of Software Requirements Determination. Journal of Management Information Systems 15, 99–118 (1998)
31. Reinig, B.A.: Toward an Understanding of Satisfaction with the Process and Outcomes of Teamwork. Journal of Management Information Systems 19, 65–83 (2003)
32. Mennecke, B.E., Valacich, J.S.: Information Is What You Make of It: The Influence of Group History and Computer Support on Information Sharing, Decision Quality, and Member Perceptions. Journal of MIS 15, 173–197 (1998)
33. Green, S.G., Taber, T.D.: The effects of three social decision schemes on decision group process. Organizational Behavior and Human Performance 25, 97–106 (1980)
34. Rittgen, P.: Negotiating Models. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 561–573. Springer, Heidelberg (2007)
35. Dean, D.L., Orwig, R.E., Vogel, D.R.: Facilitation Methods for Collaborative Modeling Tools. Group Decision and Negotiation 9, 109–127 (2000)
36. Gjersvik, R., Krogstie, J., Følstad, A.: Participatory Development of Enterprise Process Models. In: Krogstie, J., Halpin, T.A., Siau, K. (eds.) Information Modeling Methods and Methodologies, pp. 195–215. IGI Global, Hershey (2005)
37. Rittgen, P.: Collaborative Modeling - Roles, Activities and Team Organization. International Journal of Information System Modeling and Design 1, 1–19 (2010)
38. Ssebuggwawo, D., Hoppenbrouwers, S., Proper, E.: Assessing Collaborative Modeling Quality Based on Modeling Artifacts. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 76–90. Springer, Heidelberg (2010)
39. Gronau, N., Müller, C., Uslar, M.: The KMDL Knowledge Management Approach: Integrating Knowledge Conversions and Business Process Modeling. In: Karagiannis, D., Reimer, U. (eds.) PAKM 2004. LNCS (LNAI), vol. 3336, pp. 1–10. Springer, Heidelberg (2004)

Instances over Algorithms: A Different Approach to Business Process Modeling

Stefan Hofer

University of Hamburg, Department of Informatics,
Software Engineering Group and
C1 WPS GmbH

Abstract. Most of today's approaches to business process modeling interpret processes as algorithms. As a result, modelers often try to cover every possible outcome of a process. This algorithmic view increases the amount of information that has to be captured and reviewed. As an alternative, we propose to focus on exemplary process instances instead. We call this concept *Exemplary Modeling* and have built a modeling method around it. In this paper we explain its merits and illustrate its feasibility by reporting on our practical experience.

Keywords: business process modeling, requirements engineering, instance modeling, CSCW.

1 Introduction

No matter if you optimize business processes, analyze requirements for the development of a new software system or aim to improve IT-business alignment: Business process models are commonly used to capture the necessary information to achieve these goals. Usually, business process modeling can be divided into two major activities:

– Modelers gather and condense information into graphical models.
– Domain experts review and approve the models.

These steps become increasingly difficult with the amount of information that has to be considered. The more information they comprise, the larger and more complex the models become. Also, the total number of models typically increases with the amount of information to capture. In professional environments modelers and domain experts have to cope with this situation. Thus they could draw significant support from concepts that address gathering of extensive information, condensing it into models, and reviewing these models.

In this paper we describe a concept which provides that support – *exemplary modeling*:

– We will describe how modeling of exemplary instances of processes differs from classical approaches. We will point out how the concept helps modelers to gather and condense information and domain experts to review and approve models (see Sect. 2).

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 25–37, 2011.

– We demonstrate the applicability of exemplary modeling by presenting a modeling method that is based on this concept. The method was developed cooperatively at the University of Hamburg and C1 WPS GmbH (see Sect. 3).
– We reflect on the concept by sharing our experience gained by using the modeling method in commercial projects (see Sect. 4). Drawbacks and limitations will also be discussed.
– A comparison of the concept with related work concludes this article.

2 Exemplary Modeling

2.1 Motivation

Being involved with business process modeling in academic and professional contexts, we noticed how hard it is to get domain experts to participate in modeling. Shipman and McCall provide a possible explanation:

> "The difficulties that users have in formalizing information are not just interface problems. More effort is required of users in part because formal representations require the explicit statement of information that might have been left implicit [. . .] in a less formal representation. In addition, substantial effort is typically required to learn a formal representation. This requires both talent and interest in formalization that users are unlikely to have." [13, p. 286]

We compared this situation with a similar problem domain – requirements engineering, and particularly user participation during software development. Our software engineering process is heavily influenced by the *Tools and Materials Approach* [16]. This approach fosters the use of prototypes and pilot system to incorporate user feedback into the development process. Prototypes are very tangible and allow for the discussion of concrete workflows and boundary conditions. Thus, they provide a proper basis for discussion about the system.

In our opinion, most approaches to business process modeling lack such a foundation because models are usually rather abstract and algorithmic descriptions of processes. Typically they are produced by decomposition. Processes are broken down into subprocesses until the level of activities is reached. Graphically, they are depicted as a sequence of boxes and arrows – much like diagrams used in *Structured Analyses and Design Technique* [11] or similar engineering techniques. Two prominent examples of this kind of business modeling are the *Business Process Modeling Notation* (*BPMN*) and *Event-driven Process Chains* (*EPC*) (for both see [15]). In fact, EPCs were originally conceived as a modeling technique for software engineers, not for business process modelers and domain experts. The underlying mindset aims at describing precisely how a process is executed. Thus, activities and decisions are modeled in an "algorithmic" way; a way suitable for automating processes with software. However, the behavior of interactive systems such as typical business applications cannot be reduced to algorithms as Wegner explains in [14]. Jacobson et al. criticize that algorithmic modeling techniques are used for business processes:

> “All these techniques come from the computer world. It is as though we learned to think in a way that works for computer systems, and we realized we could apply the same way of thinking to describe an organization [. . .]. We believe information systems should be described so they are easy for people to understand, with abstractions that people can comprehend. We think it is bizarre to apply the way of thinking that governs computer systems to business process.”[6, p. 36]

A main feature of algorithmic modeling techniques is that case switches (called *gateways* in BPMN [2]) like “exclusive or” are an important element of the models. Each switch increases the number of paths from start point to end point. Therefore, a model displays all these possible paths – a graphical representation of the process’s transitive hull of states and state transitions. But from a domain expert’s view, different paths may represent very different outcomes. That means that their graphical proximity in the model does not necessarily correspond to their relationship in the real world. In a similar way, case switches do not express that some paths are more important or more likely than others.

Additionally, the use of case switches is suggestive of depicting *complete* models, meaning that every possible path is covered. This blends in with the underlying logic of algorithmic modeling techniques: The execution of an incomplete algorithm would result in an error or an undefined state. In our experience, this ambition for completeness leads either to an increase in the number of models or the use of more case switches. In fact, completeness is unachievable for interactive business applications: “Incompleteness is a necessary price for modeling independent domains of discourse whose semantic properties are richer than the syntactic notation by which they are modeled [. . .].” [14, p. 98]

In conclusion, the common practice of modeling business processes with methods that have an “algorithmic” nature does provide plenty of room for improvement.

2.2 Exemplary Modeling as a Different Approach to Modeling

Our approach to business process modeling is based on the idea that concrete instances of a process are easier to understand than abstract descriptions of all possible outcomes. These exemplary instances of a process can be modeled without case switches – every model represents one path from process start to end. As a consequence, modelers and domain experts do not have to deal with the complete set of possible states that a process may have during its execution. Consider this example:

A sales process for an insurance takes different paths depending on the client being already customer of the insurance company or not. In an usual process model, this would be modeled as a “exclusive or” switch, splitting the process model into two paths[1] (see Fig. 1). With our approach, we would create one model for each significant path through the model (so, two for this example – see Fig. 2).

[1] Omitting the possibly negative outcome of the second switch for the sake of brevity.

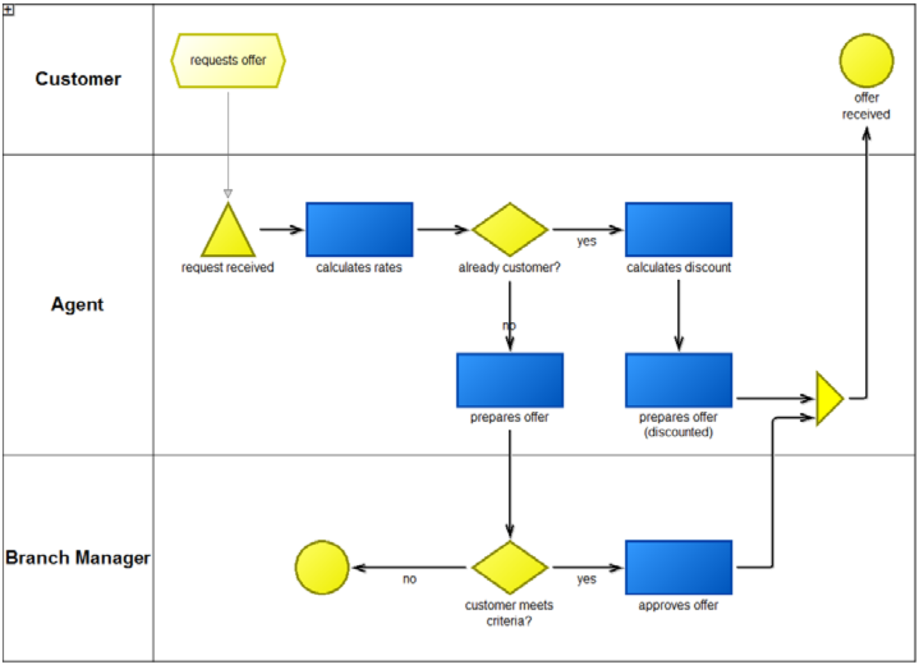


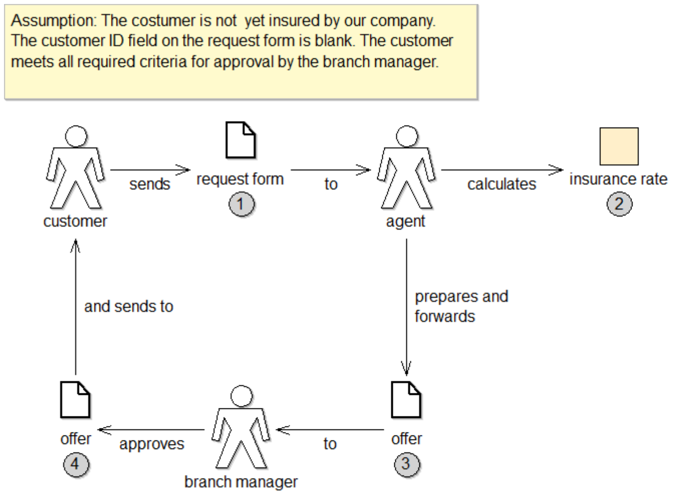
Fig. 1. An insurance sales process modeled with BPMN

Every exemplary process instance is considered a *scenario*. In [4], John M. Carroll describes scenarios as follows:

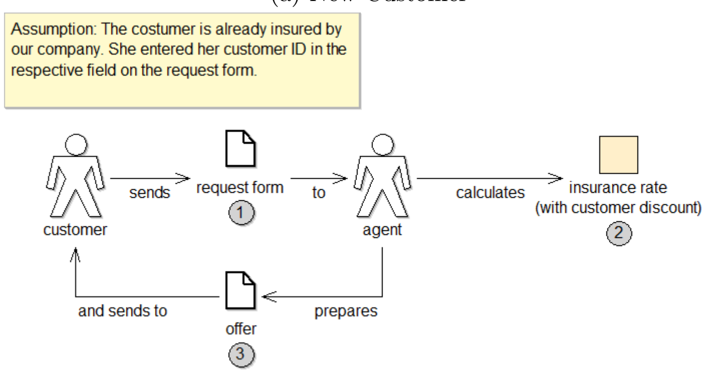
– "Scenarios are stories – stories about people and their activities." [4, p.46]
– Scenarios "mention or presuppose a setting", meaning they are set in a specific context. [4, p.47]
– "Scenarios include agents or actors [...] each typically with goals or objectives."[4, p.47]
– "Scenarios have a plot; they include sequences of actions and events" [4, p.47]

In contrary to other modeling approaches, a scenario-oriented approach does not describe actions in an algorithmic way. Instead, it focuses on the actors (human or IT systems) that are involved and order their activities in an exemplary sequence. That way, activities and their IT support can be discussed without consideration of an algorithm that could automate these activities.

Obviously, modeling every unique path of a process may take would increase the number of models dramatically. However, in our experience it is usually sufficient to look at some characteristic paths that a process may take. This is why the number of models for each process can be limited by focusing on the most important scenarios. Smaller variations and optional steps can easily be annotated in textual form. Reconsider the previous example:



(a) New Customer



(b) Existing Customer

Fig. 2. Exemplary insurance sales process models

Let us assume that a new customer may be treated like an existing customer if his or her spouse is already an existing customer. The only difference is that some additional fields have to be filled out on the application form. Instead of creating additional models for this variation, a textual annotation is sufficient to incorporate it into the insurance sales process for existing customers. Figure 3 depicts this variation of Fig. 2b.

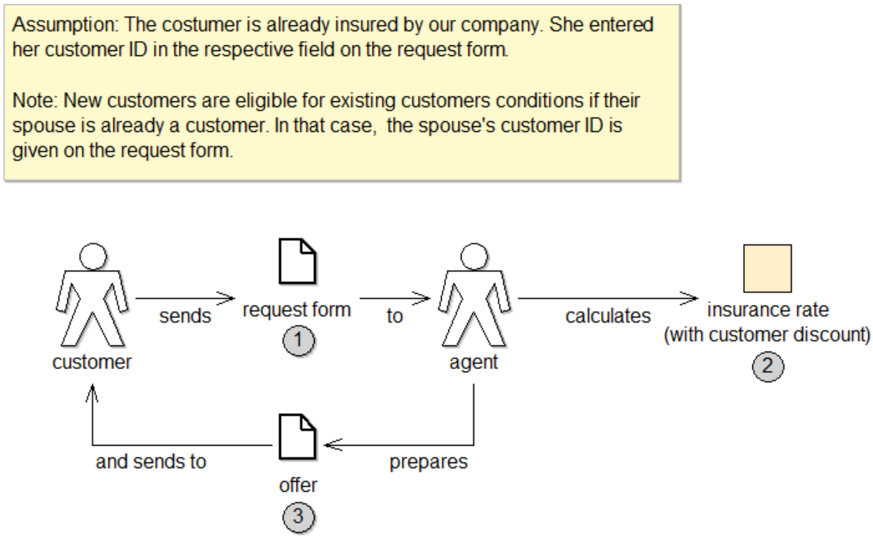


Fig. 3. Small variations are covered by annotation rather than separate models

3 The Exemplary Business Process Modeling Method

Exemplary modeling is merely a concept. To turn it into a modeling method for professional use, one needs:

- a modeling notation,
- a modeling process, and
- a modeling tool.

This section describes how these were built around the idea of exemplary modeling. The result was a tool-supported method called *exemplary Business Process Modeling (eBPM)* (see [3]).

3.1 Cooperation Scenarios

The eBPM notation consists of several types of diagrams with *cooperation scenarios* being the most important one. It becomes obvious that cooperation scenarios implement the concept of exemplary modeling when looking at some of the graphical elements that can be used in this diagram type:

- *Actors* are people or IT systems that perform actions. They may interact with each other by exchanging Information (see *Work Objects*).
- *Work Objects* can take the form of documents, data, tools and so on.
- *Relationships* express interaction of actors, creation or editing of things, use of IT systems and so forth.

Consequentially, there are no case switches available. It should also be noted that modelers can choose whether they use actors to represent a concrete actor (e.g. "Mr. Smith") or a role (e.g. "agent").

eBPM's cooperation scenarios were derived from a diagram type called *cooperation pictures*, proposed by Krabbel et al. in [8]. Cooperation pictures were created as a requirements engineering method to capture cooperation. Thus both diagrams have a strong focus on depicting cooperation. For example, they both use pictograms to create a more tangible view on cooperation.

Compared to cooperation pictures, cooperation scenarios describe a business processes as a sequence of steps. However, in contrast to other methods like BPMN this sequence is not depicted as a control flow. Instead, an order is established by numbering the steps (see Fig. 3). A more detailed look on cooperation scenarios shows that every step has to be carried out by an human or a software system. To do that, an actor performs an activity which is modeled as relation between actor and a work object. Activities are depicted as arrows and annotated with verbs. Thus, every step can easily be transcribed in textual form in a subject-predicate-object pattern. As a result, cooperation scenarios can be "read" like stories – just like suggested by Carroll (cf. Sect. 2.2).

Although the notation is not the focus of this paper, we find it noteworthy that its tangibility complements exemplary modeling. As stated above, work objects are depicted as icons that match the real world objects they represent: A document icon symbolizes a document, a telephone icon symbolizes communication via telephone, ...

The cooperation scenario and the other diagram types of eBPM are embedded in a meta model. Other important diagrams include:

– *Model of Terms* for collecting and structuring work objects.
– *Model of Roles* for collecting and structuring roles that actors can assume.
– *IT-Landscape* for collecting and structuring IT systems.

Since these diagram types are not necessarily required to comprehend the idea of exemplary modeling, we will not explain them any further in this article. Additional information can be found in [3].

3.2 Modeling Process

Typically, eBPM models are developed in workshops. Participants include modelers, domain experts and, if needed, technically oriented experts. During the workshop, the model is visible for all participants (e.g. projected on a screen). Since only concrete cases are modeled, the resulting models can easily be validated and agreed upon by the participants. Thus, the workshop covers both of the modeling steps described in Sect. 1: Information gathering and condensing as well as reviewing.

To ensure the models are scenario-based, the modeler has to guide the modeling process. She does so by discussing the boundary conditions of the exemplary processes and includes the description into the graphical model. The examples in Fig. 2 each contain such a description in a yellow textbox. Since every modeled action is performed by an actor, the modeling process advances by asking questions like:

– Who performs this action?
– What things does the actor use for that step?

3.3 Tool Support

Although eBPM may be used with a flip chart or generic drawing tools like Microsoft's Visio, tool support was implemented using the BOC Group's *Ado* modeling platform. A specific software tool offers various advantages:

– References between models and/or model elements allow for a hypertext-like navigation.
– The enumerated actions of a model can be visualized one step at a time. This "walkthrough" feature helps to read large models by telling a "visual story".
– Models can be evaluated by certain criteria – for example, which IT systems are used by which actors.

A free version (without any restriction to non-commercial use) is available via the University of Vienna's *Open Model Initiative* website (see [9])[2].

4 Reflection

In this section, we reflect upon the use of the eBPM method in general and upon the concept of exemplary modeling in particular. To do so, we begin with an overview of the fields where eBPM was used successfully. Then, we compare our experience with the initial claim that exemplary modeling supports modelers and domain experts to perform their modeling activities (see Sect. 1).

4.1 Field of Application

eBPM has been in use for several years now in so different domains as banking, insurance, federal government, logistics, and health care. It has mainly been adopted to support requirements engineering rather than for "classical" business process modeling. The following list presents some fields in which the method has been used so far:

Software Development. Current business processes and their IT support were modeled using eBPM. From this starting point, the models for the to-be processes were derived. These models had a strong focus on how the new systems should support the user's activities.

[2] So far, the software is available in German language only.

Evaluation of *Commercial Off-the-shelf* (*COTS*) Software. Organizations that wanted to buy COTS software used eBPM to describe how IT support for their business processes looked like ideally. These models were used in the acquisition process. The vendors of possibly fitting COTS software had to describe how the modeled activities would look like if their product was used. This way, it became evident for the organization how well each possible solution fitted and how they compared against each other.

Bringing Software into Service. Complex software solutions are often introduced into organizations incrementally. eBPM was used to model how each step affects the business processes. Also, the use of interim solutions (like adapters between systems or organizational workarounds) were planned with help of eBPM models. Thus, the transition to the new software system was smoothened.

Quality Assurance. Software migration projects require a quality assurance process that helps ensuring the success of the changes. eBPM was used to describe core business processes that were of crucial importance to the organizations. Then, to-be models of the processes were derived to describe the planned outcome of the migration. In a third step, these descriptions were turned into process-oriented test cases.

Organizational Changes. Organizations that wanted to change their structure used eBPM to describe their current work places. After analyzing the models for improvement or reorganization, eBPM was used to outline how the to-be work places should look like and how IT could support them.

4.2 Experience

Comprehension. In our experience, eBPM models are easy to understand, regardless of one's background or domain. An indication of that ease of use is that we do not explain the eBPM notation to the participants of modeling workshops. These workshops can be successful even if the modeler is the only one of the participants who has experience with the method. However, we usually have to explain the exemplary character of the models. Especially participants who used other modeling approaches before are used to distinguishing cases.

Concreteness of Scenarios. Although a cooperation picture depicts an instance of a process, modelers have some freedom in choosing its level of concreteness. As described in Sect. 3.1, actors may represent individuals or roles. We recommend the latter as often several representatives of a role participate in a modeling workshop. Modeling roles as actors allows all representatives to identify with the actor. Simultaneously, it helps them to abstract from little details that differ from individual to individual.

Variations. The power of scenario based modeling becomes evident when participants are constrained to separate important from less important variations of business processes, thus facilitating information gathering as well as reviewing of models. However, the pragmatic approach of handling small, but not negligible variations as textual annotations comes with a pitfall. It may lead to lots of descriptive text accompanied by a weak model instead of a rather rigorous model with some additional textual comments.

Responsibilities of Modelers. Naturally, the concept of exemplary modeling and the freedom of choice that eBPM provides call for a skilled modeler. We learned from training prospective eBPM modelers that the usefulness of models increases dramatically with experience. That experience is of particular importance when conducting a modeling workshop (see Sect. 3.2). For example, the modeler has to guide the process of finding a sufficient amount of scenarios for a given purpose.

Feedback Loop. Notation, modeling process and tool support allow for "live modeling". Breitling et al. conclude: " The modeler immediately translates the contributions of the participants into eBPM models that are visible to everyone. This procedure shortens the feedback loop significantly." [3, p. 189] This effect is amplified by the "walkthrough" feature provided by the modeling tool (see Sect. 3.3).

We noticed that approving of models is supported especially well by exemplary modeling. Domain experts can easily dismiss incorrect models because of their concreteness. It is also interesting to see that domain experts identify with the actors in the model. They recognize their work objects and the actors they interact with. This too helps them to evaluate process models.

Notation. Both the symbols of the modeling language as well as the layout have a profound impact on model comprehension [12]. Hence, it is difficult to distinguish to what extent the applicability of eBPM can be attributed to its notation, to the concept of exemplary modeling, and to the skills of the modeler.

Limitations. Processes that mainly consist of complex state transitions proved to be difficult to capture solely with exemplary models. Again, we found it useful to focus on a small number of typical chains of transitions so that one could gain an idea of the general purpose of the process. But in addition we used state transition diagrams to clarify the lifecycle of work objects. This combination was valuable when dealing with processes that have a legal dimension like for example dunning processes of insurance companies.

It should be noted that all this experience was gained in industries that rely on both people and IT systems to execute their business processes (see Sect. 4.1 for fields of application). eBPM was neither developed as a language for process automation nor as modeling method for embedded systems or similar areas.

In summary, we noticed that exemplary modeling does indeed support modelers and domain experts in their respective modeling activities.

5 Related Work

5.1 Instance Modeling

Since exemplary modeling deals with instances of business processes a discussion of *instance modeling* seems appropriate. Within the area of information modeling, there are several approaches that propose modeling instances as an alternative or an extension to some sort of a "class" model or schema. For example, Agarwal et al. suggest modeling schemas that describe the building blocks and relations that can be used to define business processes (see [1]). Although the resulting process models are instances of the process schema in terms of object-orientation, they are not scenarios. Thus, they are not instances in terms of exemplary modeling.

A different instance modeling approach proposed by Parsons and Wand in [10] aims for a semantically richer database design by making classification of data instances optional (i.e. allowing decoupling from a relational database schema). Their design consists of two layers of models, one for instances and one for classes. In a somewhat comparable way, cooperation pictures (containing "instances") may refer to models of terms and models of roles (see Sect. 3.1). The latter model types can be used to define structured collections of work objects and actor's roles, respectively.

5.2 Use Cases

Finally, we want to address a well known modeling approach that is also scenario based: *Use Cases*. Since there are several variants of use cases (see for example [7]) we will limit this discussion to Cockburn's prominent version as described in [5].

Apparently, use cases are text-based whereas eBPM is a graphical modeling method. Although there are so-called *Use Case Diagrams* in UML, Cockburn makes clear that they are "not a notation for capturing use cases"[5, p. 128]. Use cases and eBPM both center around actors and use scenarios to describe business processes. Yet, there are differences in focus. In [5], Cockburn distinguishes three levels of use cases:

- *User Goals* correspond to elementary business processes.
- "*Summary-level* goals involve multiple user goals. [...] They show the context in which the user goals operate. They show life-cycle sequencing of related goals. They provide a table of contents for the lower-level use cases [...]." [5, p. 64]
- "*Subfunction-level* goals are those required to carry out user goals." [5, p. 66]

We do not make this distinction as exemplary modeling works on all three levels. However, eBPM is not just based on exemplary modeling but also on modeling cooperation. Thus, we usually use eBPM to capture a level between user goals and summary-level goals. In summary, eBPM has a strong focus on how several actors work together to achieve a goal.

Still, the similarities between the two approaches may indicate that eBPM is merely a graphical addition to use cases. However, this paper only covers one of several of eBPM's model types, neglecting valuable modeling capabilities that are not related to use cases (see Sect. 3.1).

All in all, it should not be surprising that we have incorporated eBPM in use cases several times. Cooperation pictures are well suited as a use case's "main success scenario" [5, p. 28]. Extensions and variations of use cases blend well with the way we deal with variations in exemplary modeling (see Sect. 2).

6 Summary

In summary, we think that an algorithmic view on process modeling adds to the problem of how to handle extensive information:

- The quest for completeness increases the amount of information that needs to be gathered.
- Subsequently, more information needs to be condensed into models. This leads to either more complex models and/or an increased number of models, organized as hierarchies of processes and sub-processes.
- Due to their abstract nature und their extensiveness, these models are usually hard to review and approve.

In this paper, we proposed a different approach to process modeling. This approach is based on the idea that it is easier to capture and evaluate exemplary instances of processes rather than an abstract description of all their possible outcomes. We explained how scenarios provide a proper basis for exemplary modeling. By introducing our Exemplary Business Process Modeling Method (eBPM) we showed that the concept of exemplary modeling is suitable for a professional modeling technique. This was illustrated by presenting several fields of application that eBPM has proven successful in. Using exemplary modeling in these fields of applications allowed us to gain experience and reflect on the concept. We observed, that exemplary modeling (as implemented by eBPM) indeed helped to avert the disadvantages we experienced with other modeling techniques. Finally, we compared our approach to Use Cases and pointed out how these two approaches can be combined.

References

1. Agarwal, R., Bruno, G., Torchiano, M.: Instance Modelling – Beyond Object-Oriented Modelling. In: Proceedings of the 3rd International Conference on Information Technology (2000)
2. Object Management Group: BPMN Graphical Elements, http://www.bpmn.org/Samples/Elements/Gateways.htm (last visited 2011-06-30)
3. Breitling, H., Kornstädt, A., Sauer, J.: Design Rationale in Exemplary Business Process Modeling. In: Dutoit, A., McCall, R., Mistrik, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 191–208. Springer, Heidelberg (2006)

4. Carroll, J.: Making Use. Scenario-Based Design of Human-Computer Interaction. MIT Press, Cambridge (2000)
5. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley Longman, Amsterdam (2000)
6. Jacobson, I., Ericsson, M., Jacobson, A.: The Object Advantage: Business Process Reengineering with Object Technology. Addison-Wesley Longman, Amsterdam (1995)
7. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Reading (1992)
8. Krabbel, A., Wetzel, I., Ratuski, S.: Participation of Heterogeneous User Groups: Providing an Integrated Hospital Information System. In: Proceedings of the Participatory Design Conference PDC 1996, pp. 241–249 (1996)
9. Open Models Initiative Website, http://www.openmodels.at/web/bpm (last visited 2011-06-30)
10. Parsons, J., Wand, Y.: Emancipating Instances from the Tyranny of Classes in Information Modeling. ACM Transactions on Database Systems 25(2), 228–268 (2000)
11. Ross, D.T., Schoman, K.E.: Structured Analysis for Requirements Definition. IEEE Transactions on Software Engineering SE-3(1), 6–15 (1977)
12. Schrepfer, M., Wolf, J., Mendling, J., Reijers, H.A.: The Impact of Secondary Notation on Process Model Understanding. In: Persson, A., Stirna, J. (eds.) PoEM 2009. Lecture Notes in Business Information Processing, vol. 39, pp. 161–175. Springer, Heidelberg (2009)
13. Shipman, F.M., McCall, R.J.: Supporting Knowledge-Base Evolution With Incremental Formalization. In: Proceedings of the ACM Human Factors in Computing Systems Conference 1994, pp. 285–291 (1994)
14. Wegner, P.: Why Interaction is More Powerful Than Algorithms. Communications of the ACM 40(5), 80–91 (1997)
15. Weske, M.: Business Process Management. Springer, Berlin (2007)
16. Züllighoven, H.: Object-Oriented Construction Handbook. dpunkt.verlag, Heidelberg (2005)

Specifying Flexible Business Processes Using Pre and Post Conditions

Jeroen C. van Grondelle and Menno Gülpers

Be Informed BV, Apeldoorn, The Netherlands
{j.vangrondelle,m.gulpers}@beinformed.com
http://www.beinformed.com/

Abstract. Today's business processes have to address many, complex requirements. Mass customization leads to personalized, contextual products being offered by governments and enterprises and, as a result, the business processes for selling and offering these products are divers and contextual as well. At the same time, regulations in the area of compliance and a growing rate of change introduce additional complexity. These developments pose major challenges to the field of business process modeling. Conventional process modeling, in terms of activities and the flow they are executed in, has proven to lead to complex and often rigid business processes.

In this paper, we present our experiences with specifying business processes based on activities and their pre and post conditions instead of flow. The resulting business processes are flexible: they allow knowledge workers to influence their own process and they do not require the explicit modeling of flows to deal with exceptions and switching between straight through and human processing.

Our formalism facilitates an agile modeling process. The formalism helps involving business users in modeling as it can be expressed well into natural language. Furthermore, it allows for separation of concerns in modeling by having an algorithm consolidate the different areas of requirements into an executable business process. Analysts can focus on modeling the different concerns and are no longer required to manually consolidate all the requirements into a business process that is believed to address all of them.

Keywords: Business Processes, Adaptive Processes, Goal Orientation, Business Rules, Complexity, Natural Language Generation.

1 Introduction

Today's organizations are dealing with a number of trends that increase the complexity of their business. Most governments and enterprises offer products that have many variants and options, depending on the customers' context and his individual choices. Enterprises use this as a marketing tool, increasing their revenue by addressing multiple target groups with specific products. Governments offer products, like grants, taxes and permits, that are the result of complex

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 38–51, 2011.

policies. The business processes that are introduced to produce, sell or apply for these products are often equally individual and contextual of nature. They need to offer the knowledge workers that perform them the flexibility that matches their experience, while at the same time guaranteeing consistency and quality of the result.

At the same time, organizations need to deal with a growing rate of change. Enterprises want to react to changing market circumstances ever faster, resulting in changes to both their products and the supporting business processes. As a result of regulation, the rules they must comply with change or new constraints are introduced.

This agility typically poses requirements to the modeling process. Changes have to be implemented often and fast. This is specifically a challenge in the area of validation, as conventional testing of all possible scenarios is no longer feasible or at least takes too long. Reviewing by business experts and the ability to trace models to their source are becoming more important. Also, the typical IT approach of handing over a specification between disciplines like analysis, functional design and engineering consecutively has become a bottleneck. Equally important, agility introduces requirements of its own to the business process itself. When changes occur frequently, business processes will have to deal with active processes for old versions of products. Either by migrating transparently to the newest product definitions or by completing processes against the policy that was in place when the process started.

Classically, IT has defined business processes in terms of the activities that are performed in an organization and the order in which this is done. Conventional process modeling standards are available, like OMG's Business Process Modeling Notation [1] and the Business Process Execution Language [2]. More recently, the terms Dynamic Case Management [3] and Adaptive Case Management [4] were coined for more dynamic, rule oriented approaches, that address the fact that the complexity we described earlier has proven to pose a challenge when using the metaphor of flow. It typically results in processes with a lot of forks to accommodate process variants or exception flows. Alternatively, a large number of processes is often created that match the large number of product variations, even when these process variants essentially perform the same task.

Using declarative techniques in process modeling is broadly seen as a way to overcome limitations of conventional, imperative approaches, both in industry and academics. In for instance [5], Goedertier and Vanthienen describe the differences between declarative and imperative process modeling and how a declarative approach helps to model the actual business concerns instead of, with an imperative approach, model a process flow that meets these constraints implicitly. In [6], Pesic and Van der Aalst introduce a ConDec language that attempts to reduce over specification and introduce flexibility by using declarative techniques to specify business processes. In [7], Schonenberg et al. describe different aspects of flexibility required in today's business processes. Also in this paper, replacing flow by declarative (precedence) constraints is presented as a source of flexibility. Andersson et al. address flexibility, traceability and business orientation in [8] by introducing an activity dependency model that, like our formalism, focusses on the type of dependencies that exist between activities.

This paper introduces the formalism we use in Be Informed's Business Process Platform[1] to specify business processes in a declarative, goal oriented way. It leaves consolidation, and the combinatorics and order that follow from it, to be performed by an algorithm, so that enterprise modeling professionals can focus on modeling the underlying business aspects in terms of requirements. The resulting processes can be completely prescriptive, but at the same time offer great flexibility. They allow experts to influence their own work and they deal with exceptions well.

2 Specifying Flexible Goal Oriented Processes

In this section we introduce the formalism we use for specifying business processes.

2.1 Pre and Post Conditions

The formalism is based on the notion of activities, the pre conditions that have to be met for these activities to be performed and the consequences that result from these activities, expressed in terms of post conditions.

Be Informed uses a graph oriented representation consisting of concepts and relations between concepts. Concepts and relations have a type and can have properties. Multiple labels and fragments of text containing definitions, examples etc. can be associated with concepts, as can references to the underlying content that is the source of a concept or where it occurs. The types of concepts and relations that are available in a model are introduced in a meta model, that also contains type hierarchy rules and constraints about which relations and properties may occur at concepts of which type.

The central types in the meta model discussed here are the concept types that represent cases and the activities performed within cases. Furthermore, it contains the conditional relation types that capture pre conditions and an abstract relation type called Consequence which has post condition semantics. The meta model is summarized in Figure 1.

Nodes in Figure 1 introduce concept types, edges indicate that relations of the specified type may occur between the concept types it connects. For instance, the relations between Activity and Decision introduce both a pre and a post condition. The Requires Taken relation represents that instances of activity may require a specific decision to be taken before the activity may be performed. The Decides relation introduces the possibility for an activity to have the taking of a specific decision as post condition. This can be both read in a formal, post condition way and a more intuitive, procedural way. In the first case, the activity is only completed if the specified decision is taken. The informal way would be

[1] http://www.beinformed.com/

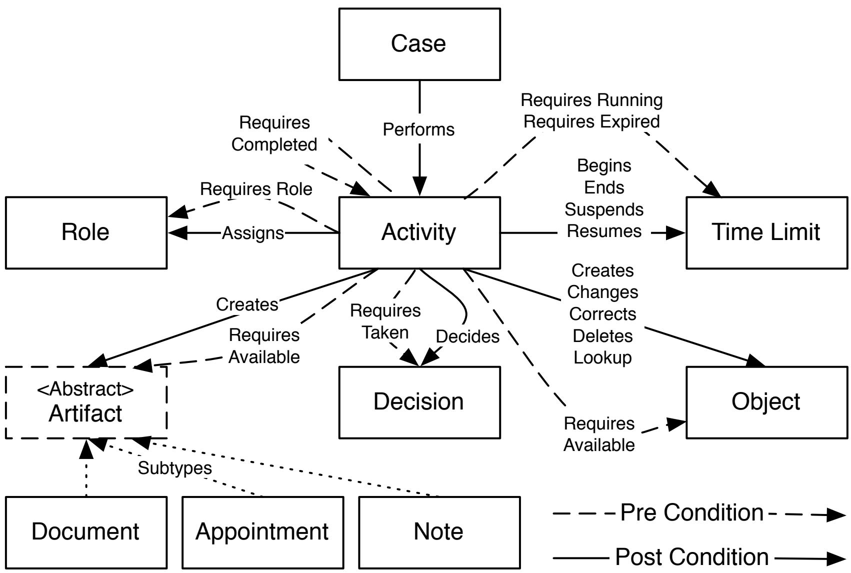


Fig. 1. Summary of the Meta Model for capturing Business Processes

to state that the decision is taken as part of the activity. The latter is correct in most cases, but overlooks the fact that the decision may have been taken in another context. The formal, correct interpretation allows the activity to be completed in such cases.

The meta model introduces **artifacts**, that represent documents produced, meetings held and notes taken during the process. Creation of artifacts is typically modeled as a consequence of an activity, their existence is often a pre condition for performing other activities. For instance, in Figure 2 an intake meeting is planned as part of the activity of registering the application form, the existence of an intake form is a pre condition for performing an assessment activity.

The meta model also enables the modeling of data **objects** used in a case. These objects can be local, case scoped objects or persistent objects in a registration.

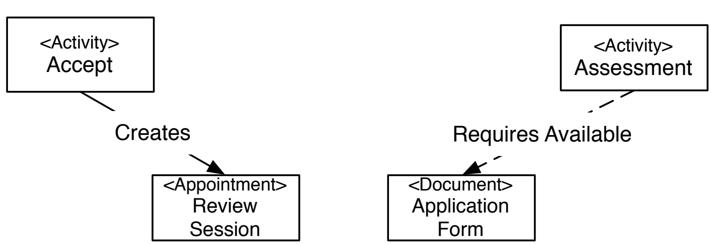


Fig. 2. Relations between Activities and Artefacts

This difference is omitted for the remainder of the paper. Storing or updating objects is typically modeled as a consequence of an activity, their availability is typically a pre condition.

The meta model identifies the **decisions** taken as part of the process. Often implicitly embedded in process flows, we believe decisions are a core concept in today's business, as they capture a major part of what people do in organizations today. Is this customer interesting to us? Is this citizen entitled to a grant? And if so, how much is he entitled to? Within the business process meta model, decisions are identified and if necessary acted upon. The definition of the decision itself is done in separate model fragments, with a different, decision oriented meta model.

In people centric processes, modeling the **people involved** is important. Our formalism has the notion of both user roles and involvement roles. User roles are typically used to represent user competences, responsibilities etc. Involvement roles encode similar aspects, but specific to the role a user plays in an individual case. Issuing/assigning these roles can be a consequence of an activity, having the appropriate roles is a typical pre condition.

An important aspect of a business process is the applicable **time limits**. In our formalism, activities may begin, end, suspend or resume a time limit. Other activities may require a certain time limit to be either still running or already expired as a pre condition. For instance, Figure 3 shows a maximum response time to a grant application that is specified as a time limit, which is suspended for the time the citizen uses to produce additional information on request. The retention period of a case starts on publication of the decision and the archiving activity requires this period to have expired before archival may be performed.

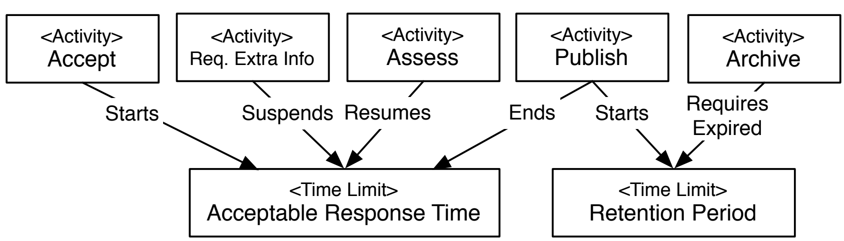


Fig. 3. Relations between Activities and Time Limits

2.2 An Example: Grant Applications

The model in Figure 4 captures the business process of applying for a grant. It introduces activities for accepting, assessing and archiving the application for the grant and publishing the decision. Assessing the grant application is done by deciding whether the grant is eligible, which can result in a confirmation or rejection letter. Time limits monitor the response time and the retention period.

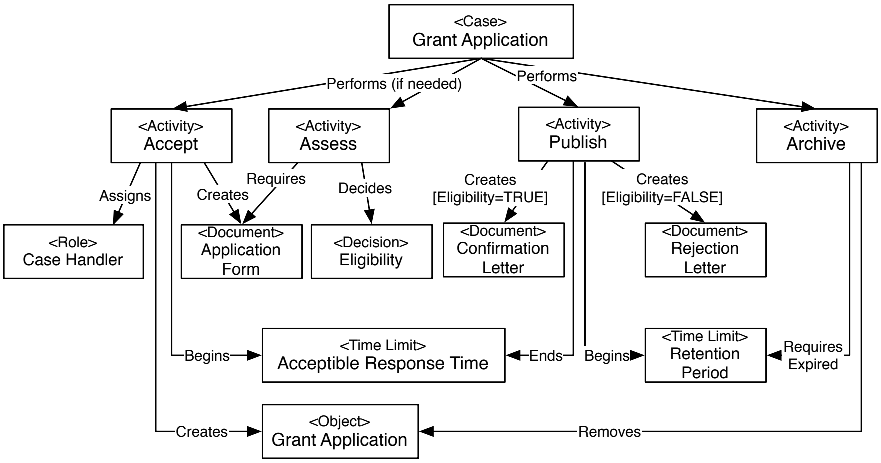


Fig. 4. A Model for Handling Grant Applications

The case Grant Application has to perform the activities Publish and Archive to be successfully completed and can perform the activities Accept and Asses. The latter two activities are not mandatory, their post conditions (and not the activities themselves) act as pre conditions for other activities. The creation of an application form is a pre condition for the assessment activity, in which a decision is taken regarding the grants eligibility. Publishing the decision requires that this decision has been taken, since the creation of a confirmation or rejection letter depends on the outcome of the eligibility test. However, the eligibility could be determined in another way, since the activity Asses itself is not mandatory, resulting in the Publish activity's pre conditions to be met. Accepting a grant application not only issues a case handler and creates an entry in the grant application registration, but also starts a time limit that monitors an acceptable response time for the application of the grant. Publishing the decision will end this time limit, while at the same time starting another time limit, defining a reasonable retention period, for instance 5 years. The archiving activity can only be performed when this period has ended and results in the removal of the grant application from the registration.

3 Goal Oriented Business Processes

Although the formalism does not explicitly describes flow, an executable process can be inferred from it. As its pre conditions are continuously evaluated, the inferred process is highly dynamic and responds well to user choices and external input.

3.1 Inferring the Process

Based on a business process described using this formalism, at any time, the activities that *may* be performed next can be determined based on case state by checking which activity's pre conditions are met. This information can be used to automate a process, by offering users only the tasks that may be performed. That way, all activities will be performed only when their constraints, in terms of order, availability of other information, competence of the actor etc., are met. When a model has many pre conditions between activities, effectively encoding order, this will lead to a conventional process, where each activity may be performed if its predecessor(s) are completed. In a more complex model, this strategy typically leads to many activities that can be performed, without being prescriptive which should be performed first.

Based on the same formalism, it is possible to infer which activities *need* to be performed in order to meeting an overall goal. In our formalism, the goal is modeled by expressing post conditions at the case level. Inferring the goal oriented process is done by (recursively) inferencing which activities contribute to meeting the pre conditions of activities that contribute to the goal. An activity can contribute to meeting the preconditions of another activity directly, if the activity is a pre condition itself, or indirectly, when post conditions of a particular activity are pre conditions of the other activity.

Post conditions can be conditional or optional. Conditional post conditions might encode activities or artifacts only required under certain conditions. For instance, a grant notification needs only be sent if eligibility for the grant is established. Optional post conditions are strictly not post conditions, as they may or may not be met. Typically, these are only met if required by some other activity's pre condition.

3.2 Flexibility through Goal Orientation

The fact that goal oriented processes focus more on the requirements that need to be met than specifying in which specific way to meet them, guarantees a large degree of flexibility.

Goal Orientation Allows Knowledge Workers to Influence Their Own Process. Both the activities that may be performed and the activities that need to be performed can be presented to users. Performing the activities that need to be performed is usually the most straightforward approach for users, but experienced professionals may choose to perform activities that may be performed, but are not inferred to be needed at this time. Reasons to do this might include availability of the required information for the activity (only) at this moment or the expert judgement that a case will for instance be accepted anyway, that way predicting that the activity will be inferred to be needed in the future.

Figure 5 shows a user interface that shows the process state to a user. It distinguishes between activities that have been completed, activities that need to be performed now, activities that are not needed but are allowed and activities that may not be performed at this time. Based on the type of pre condition violated, the user interface provides feedback on why an activity may not be performed. In cases where no activities are available to a user, this is crucial to allow him or her to assess how to progress the case: Is it my role that prevents me from performing the necessary activities? Is a lack of information blocking at this time? Is it time constraints that prevent this case from progressing?

Fig. 5. Presenting Available Activities to a User

Goal Oriented Processes Allow for (Ad Hoc) Interventions. Another form of flexibility arises when flow oriented pre conditions are replaced by pre conditions on availability of data or information. Flow oriented pre conditions make it hard to override or repair a process when exceptions occur. If a certain activity is not performed, the activities that depend on it will never be performed. If, instead, the activity depends on the post conditions of the first activity, these pre conditions might be met in an alternative, possibly ad hoc, way, without the earlier activity ever being performed.

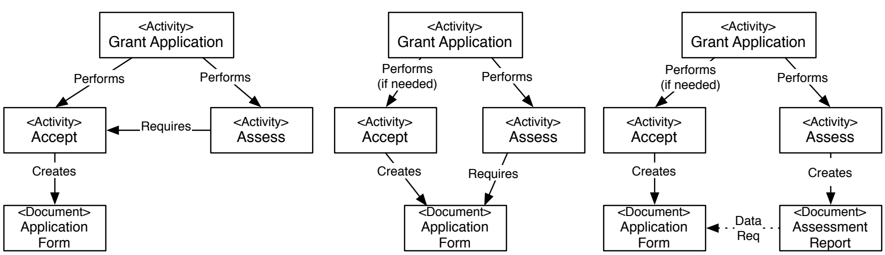


Fig. 6. Explicit and Implicit Dependencies between Activities

In Figure 6, the Application Form document required for performing an assessment is normally created in the Accept activity. However, providing the document directly in the document management system as part of an intervention would still allow the process to proceed as all pre conditions of assessment are met.

Flexible Switching between Manual and Straight through Processing. Many organizations execute their processes in a hybrid model: Part of the transactions are performed automatically and straight through, others require human intervention and are processed manually.

Typically there are two approaches used to distinguish between automatic and human work: Use explicit rules to evaluate which cases need to be processed by humans or attempt automatic processing by default and have users process the cases that terminate incompletely or lead to errors in the automated process.

In both cases, a major challenge is getting cases back into automated handling as soon as the need for human involvement is no longer present. Often, cases that fall out of the automated, straight through processing need to be processed manually from then on, even if large parts of the remaining work do not strictly require human intervention. In a flow oriented process, the only alternative is to explicitly bring cases back into the straight through processing flow, but this typically needs to be done individually for every possible reason a case might have left the straight through processing flow.

Goal oriented processes allow for far more transparent switching between manual and automated processing. For each case that terminates incompletely in straight through processing, it is known which pre or post condition failed. Instead of defining flow to deal with that, these failed conditions define classes of work in themselves, that can be assigned to human users. On resolving the pre condition, the case is available for straight through processing once again. Obviously, there might be additional issues to be solved by humans, but this approach guarantees that only tasks that really need it are processed by humans, while automated processing can always be attempted again transparently if the human task is completed.

Goal Related Feedback on Process Quality. Traditionally, feedback on the performance of business processes is collected by reporting on counts and time related aspects across groups of process instances. It is no problem to derive production reports on activities performed, reports on adherence to time limits and distribution of possible outcomes from the processes inferred from our formalism.

The fact that the process is inferred from pre and post conditions introduces additional reporting possibilities that are closely related to the goals that are to be met in the business process. The same metadata that is used to present users with the activities available to them, as depicted in Figure 5, can be used to report on the reasons why processes weren't completed in an STP fashion for instance. By reporting on which pre conditions were violated, very direct feedback is available that can be used for improving the business process. Is it the availability of information that prevents the process from being completed in a single transaction? Or are activities performed by users with insufficient expertise levels leading to reassigning of cases?

4 Verbalizing Process Specifications into Natural Language

Apart from being flexible, the business processes need to be up to date and reflect all the changes that organizations deal with on a regular basis. One of the ways to achieve this is to actively involve the business users and domain experts in the modeling of business processes. The main challenge in involving business users in enterprise modeling is the fact that most business users are not trained in formal modeling techniques. A formal, concise, visual representation can be quite intimidating to the uninitiated. One way of enabling business users to get involved in formal modeling is the use of natural language. Verbalizing graph oriented formalisms into (pseudo) natural language has been studied quite extensively, for instance by Funk et al. [11] and Kaljurand et al. [10], and we have good experiences with a similar approach based on pattern sentences [9]. Applying these techniques to the formalism introduced in Section 2 turns out to produce a very useful visualization of the business process models. Below is a summarized grammar of pattern sentences that match the presented meta model.

1. "A C case is only completed if"
   (a) "the activity A is completed." $\leftrightarrow \{Case, performs, Activity\}$

2. "The activity A may only be performed if:"
   (a) "the activity A' is completed" $\leftrightarrow \{Activity, requires, Activity\}$
   (b) "a document of type D is available" $\leftrightarrow \{Activity, requires, Document\}$
   (c) "the user is involved with role R" $\leftrightarrow \{Activity, requires, Role\}$
   (d) "the decision D was taken earlier' $\leftrightarrow \{Activity, requires, Decision\}$
   (e) "the time limit T has expired' $\leftrightarrow \{Activity, requiresExpired, TimeLimit\}$
   (f) "the time limit T is still running' $\leftrightarrow \{Activity, requiresRunning, TimeLimit\}$

3. "The activity A is only completed if:"
   (a) "a document of type D is created" $\leftrightarrow \{Activity, creates, Document\}$
   (b) "the decision D is available" $\leftrightarrow \{Activity, decides, Decision\}$
   (c) "a user was assigned role R" $\leftrightarrow \{Activity, assigns, Role\}$
   (d) "time limit T has begun" $\leftrightarrow \{Activity, begins, TimeLimit\}$
   (e) "time limit T has ended" $\leftrightarrow \{Activity, ends, TimeLimit\}$
   (f) "objects of type O have been removed" $\leftrightarrow \{Activity, removes, Object\}$

For each pattern sentence, the left hand side represents the human readable representation, the right hand side triple represents which relations in the meta model it encodes. For instance, the subsentence "the activity A is completed" is used to encode any triples of the form $\{Case, performs, Activity\}$. The fact that it is a pre condition according to the meta model is represented by prefixing it with the sentence part "A Case C is only completed if".

On verbalization, the pattern sentences are matched to the triples in the model, and the applicable parts are concatenated into complete sentences. Verbalization of a subset of the example in Section 2.2 using Be Informed Studio leads to the following sentences.

1. A Grant Request case is only completed if
   (a) the activity Publish is completed,
   (b) the activity Archive is completed
   and if needed
   (a) the activity Accept is completed,
   (b) the activity Assess is completed.
2. The activity Accept is only completed if
   (a) a document of type Application Form is available
   (b) an object of type Grant Request is available
   (c) a user is involved with role Case Handler
   (d) the time limit Acceptable Response Time has begun
3. The activity Assess may only be performed if
   (a) a document of type Application Form is available
4. The activity Assess is only completed if
   (a) a decision of type Eligibility is taken
5. The activity Publish is only completed if
   (a) if *Eligibility* = *true*, a document of type Confirmation Letter is available,
   (b) if *Eligibility* = *false* , a document of type Rejection Letter is available,
   (c) the time limit Retention Period has begun,
   (d) the time limit Acceptable Response Time has ended.
6. The activity Archive may only be performed if
   (a) the time limit Retention Period has expired.
7. The activity Archive is only completed if
   (a) objects of type Grant Application have been removed.

The pattern sentences used include feedback on the semantics of the meta model. The fact that pre conditions determine whether activities may be performed is a typical example where a modeling professional keeps in mind when interpreting the relation types used to encode pre conditions, while a business user needs to be reminded of this permanently. By including that explanation, it is automatically repeated for each activity and its pre conditions.

As we have shown in [9], pattern sentences have another important benefit: Every model can be verbalized using different pattern sentence grammars to support different expert levels and target groups. The same holds for verbalizing the models into multiple languages. Apart from translating the pattern sentences, this requires the localization of the models, as is performed in the Monnet Project[2].

5 Methodological Impact of This Formalism

We have experienced that using a formalism as introduced in Section 2 impacts the role analysts have and the way they work.

[2] http://www.monnet-project.eu/

Modeling Concerns Separately Instead of Consolidating Requirements. Classically, analysts spend a lot of time consolidating all the, possibly conflicting, requirements of the different parties involved in a business process of which they are convinced that it meets all those requirements. This takes too much time and introduces problems in the area of traceability: The analyst may be convinced that his process will violate none of the requirements, but this remains implicit in the model. The fact that it meets all requirements follows from the process of modeling, not from the model.

The formalism we introduced allows for a different approach. It is based on the fact that consolidation is left to a computer, and analysts focus on modeling business aspects, in the form of "local" model fragments that reflect a business process from an organizational unit for instance.

This also allows for separation of concerns. More applicative requirements, on how an organization for instance deals with time limits that are about to expire, can be separated from the business requirements on which time limits are to be met within the business process.

Modeling Concerns Separately Facilitates Business Ownership. The individual activities, and their pre and post conditions, are modeled in relatively modular specifications. Typically, such a local model of requirements maps well to the problem as it is perceived by its owner. For instance, an assessment department might not know or care when an intake form is filled in, but it has no problems expressing the requirement that one is available before assessment can take place. The local model on assessments will now reflect that scope and will express just the requirement.

Focus on Definitions Instead of Behavior. Classically, business modeling has had an emphasis on the flow across activities, more than on the precise definition of individual activities. The behavior is made explicit, and as a consequence, the definitions often remain implicit. This approach reverses that completely. We now focus on precise and complete definitions of activities, when they may be performed and the consequences of performing them. As a result, definitions are made explicit in the model and the behavior is left implicit. That can however be inferred from the definitions.

Apart from the benefits of flexibility and explainability, this has proven to be useful for our clients in other areas. Agreeing on terminology has helped in networked environment to really agree on the processes that are shared and communicate them to all people involved.

Refining a Single Specification Instead of Writing Discipline Oriented Deliverables. Many methods used in information analysis and design have relied on separating the analysis work into phases and writing separate documents in each phase. An initial requirement analysis might be produced, followed by a functional design. This is a basis for a technical specifications, which in turn is input for realization. Finally, test specifications are written to validate that the system built meets the requirements that were input to the initial requirement analysis. This typically requires hand over between disciplines and translations into discipline related vocabularies.

The approach we have presented allows for a method that is based more on detailing, than on more detailed deliverables replacing the more coarse grained ones. This is important, as change has large impact on the document chains described. A change impacting the requirement document might alter all documents involved, having large impact and introducing traceability challenges. Our formalism allows for expressing more coarse grained choices to be expressed in the same model as the more detailed ones. As a consequence, changes of different impact level can be dealt with in the same models, eliminating the ripple effect. At the same time, the models can be visualized into design views of different abstraction that are appropriate for different phases and responsibilities.

Consistency Checking of Constraints Instead of Runtime Processes. The fact that our processes are described declaratively impacts the ways the consistency of the processes is assessed. At modeling time, the feasibility of a process is no longer guaranteed by the predictability of an exhaustive definition of flow. Instead, the effects on the processes that may be inferred at runtime when introducing additional constraints may not always be clear up front. As the process is inferred from large numbers of pre and post conditions, (automated) consistency checking is important to detect conflicts. For instance, in cases when cycles of pre conditions occur, no feasible process can be inferred. Also, when an activity's pre conditions, or their recursive pre conditions, are mutually exclusive, an activity can never occur. As the underlying representation is based on a graph, these types of conflicts can be detected by graph traversals at modeling time.

6 Conclusions and Future Work

In this paper we have shown that alternatives to conventional process languages are available today. They address current requirements of enterprises and governments to deal with growing complexity in their product portfolios and a growing demand for agility. The formalism we presented is based on pre and post conditions of activities. The processes inferred from models using this formalism are flexible: We have shown how they allow expert knowledge workers to influence the way they perform their own work, how they can deal well with interventions and repairs and how they can switch transparently between human and straight through processing.

The formalism supports agile modeling processes with high business user involvement. We have demonstrated the verbalization into natural language, which we have experienced to be a great enabler for business user involvement in enterprise modeling. Additionally, we have presented our experiences on how this type of rule oriented process formalism impacts the methodological approach behind enterprise modeling.

As more of Be Informed's customers adopt this formalism, we would like to do additional research into quantitative aspects of this approach, such as how much it reduces specification size and complexity and how it improves modelers' productivity.

References

1. Business Process Modeling Notation 2.0, http://www.omg.org/spec/BPMN/2.0/
2. Business Process Execution Language 2.0 http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html
3. Le Clair, C., Moore, C.: Dynamic Case Management - An Old Catches New Fire, Forrester Research (2009)
4. Swenson, K.D.: Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done. Meghan-Kiffer Press, Tampa (2010)
5. Goedertier, S., Vanthienen, J.: Declarative process modeling with business vocabulary and business rules. In: Meersman, R., Tari, Z., Herrero, P., et al. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 603–612. Springer, Heidelberg (2007)
6. Pesic, M., van der Aalst, W.: A declarative approach for flexible business processes management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
7. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., Aalst, W.: Process flexibility: A survey of contemporary approaches. In: Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., Dietz, J.L.G., Albani, A., Barjis, J. (eds.) Advances in Enterprise Engineering I. LNCS, vol. 10, pp. 16–30. Springer, Heidelberg (2008)
8. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P.: A declarative foundation of process models. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 233–247. Springer, Heidelberg (2005)
9. van Grondelle, J. C., Heller, R., van Haandel, E., Verburg, T.: Involving business users in formal modeling using natural language pattern sentences. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 31–43. Springer, Heidelberg (2010)
10. Kaljurand, K., Fuchs, N.E.: Verbalizing OWL in Attempto Controlled English. In: Proceedings of Third International Workshop on OWL: Experiences and Directions, Innsbruck, Austria (2007)
11. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: CLOnE: Controlled Language for Ontology Editing. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 142–155. Springer, Heidelberg (2007)

Diagram Notations for Mobile Work Processes

Sundar Gopalakrishnan and Guttorm Sindre

Department of Computer and Information Science,
Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway
{sundar,guttors}@idi.ntnu.no

Abstract. In mobile and multi-channel information system, the geographical location and context of the user when performing some task may be important for the design of IT applications. Yet, mainstream process models seldom capture the "where" aspect, such as the location for performing some activity. In previous papers we have performed an initial analytical evaluation and two controlled experiments comparing some notation alternatives. For all these notation alternatives the underlying assumption has been that they should be achieved as fairly small adaptations of existing process notations, using UML Activity Diagrams as an example. In this paper we provide a more comprehensive analysis of the problems, using 9 principles for visual notations proposed by Moody, and considering clean-sheet design of a process notation in addition to minor adaptations of existing ones. The paper demonstrates how this would result in quite different notations, each with their pros and cons.

Keywords: Mobile, multi-channel, information system, process model, diagram notation, visual communication.

1 Introduction

Business process modelling languages tend not to capture the location of the activities performed. For instance, BPMN [1] and UML activity diagrams [2] capture what (objects), how (sequence and parallelism of activities and decisions), who (swim-lanes), when (time triggers and time events), and to a 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. For traditional office information systems, where work is performed using desktop computers, it is understandable that physical location has not been given much attention. It is much more important whether a task is performed by the Purchases or Salary department than whether the worker is sitting in office 221 or 325, and even in a distributed IS setting, it might not be so important if one of the offices is in another country either, since anyway the usage context is quite similar, thus implying similar requirements for the desktop application.

However, as emphasized in [4], the location and context of activities performed is of much higher importance in mobile and multi-channel information systems. For instance, whether a certain information processing task should be performed in the

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 52–66, 2011.

office before going out on a power line repair job, in the car while driving, in the terrain while walking and searching for the exact site of damage, while climbing in the power mast to fix the damage, while driving back from the site, or after having returned to the office - could have a large impact both on quality, efficiency, and job satisfaction, and would therefore be an important process design decision. In turn, this decision should also have a lot of impact on what ICT tools would have to be developed to support the work process, and what requirements that these tools would have to satisfy. For instance, if the task were to be performed on foot in a dense forest, this would require different equipment and imply other usability challenges than what a desktop application is normally faced with.

It is therefore interesting to consider specialized process notations that do show location and or context of the worker. This can be done either by small adaptations of existing notations like BPMN or UML activity diagrams, or by a more radical "clean sheet" design of a novel notation specifically supporting the modelling of mobile work processes. In some previous papers [5-7] we have mainly looked at the former alternative, investigating some minor adaptations of UML activity diagrams. The purpose of this paper is to make a more general evaluation of the relative merits of both alternatives, i.e. also opening for proposals of notations that are radically different from the mainstream, using Moody's 9 principles for visual notations [5]. The research questions for this paper are as follows:

RQ1: What are the most plausible visual variables to use for showing location in process models?

RQ2: What are the relative merits of designing from scratch a novel process notation for mobile information systems versus minor adaptations of existing notations?

The rest of the paper is structured as follows: Section 2 presents background and related work in the area, section 3 presents the notation alternatives. Section 4 provides an evaluation of the notation alternatives, and section 5 concludes the paper.

2 Background and Related Work

When considering the modelling of location, it is important to differentiate between space and place. The best background for conceptualization of these aspects comes from fields of cartography and CSCW, the latter distinguishing between space and place [8-9]. “Space” describes geometrical arrangements that might structure, constrain, and enable certain forms of movement and interaction; “place” denotes the ways in which settings acquire recognizable and persistent social meaning in the course of interaction. In this work we look at modelling of 'place'. Work combing modelling of space (e.g. in maps) and conceptual aspects are described in [10].

In previous papers we first performed a preliminary analytical evaluation of some notation alternatives in [5], the evaluation based on a semiotic framework [11]. Then we compared the most promising notations in experiments [6-7], which revealed that a notation using coloured nodes for capturing location fared better than using pattern-fills or UML notes. However, there are a lot of other possible notations not evaluated yet, especially radically new notations, as the previously mentioned studies focussed on minor adaptations of UML activity diagrams.

Moody [12] defines 9 principles or guidelines for effective visual notations. 1) *semiotic clarity* means that there should be a 1:1 mapping between graphical symbols and concepts. 2) *perceptual discriminability:* How easily and accurately can the various graphical symbols be differentiated from each other? It will be easier for the user to distinguish between shapes that are obviously different from a quick glimpse (e.g., squares, circles, triangles) than between shapes that are different only in subtle details (e.g., rectangles with a varying height/width ratio, with or without rounded corners, or with differences in textual font or style inside the rectangle). 3) *semantic transparency:* How well does a symbol intuitively reflect its meaning? According to Moody, symbols can be either immediate, having a nice intuitive relationship with their corresponding concepts, opaque, having only an abstract relationship, or even worse, they might be perverse, its intuitive interpretation being misleading vs. the represented concept. 4) *complexity management:* What constructs does the diagram notation have for supporting different levels of abstraction, information filtering, etc.? 5) *cognitive integration:* Does the notation provide explicit mechanisms to support navigation between different diagrams? 6) *visual expressiveness:* To what extent does the notation utilize the full range of visual variables available? 7) *dual coding:* Using text to complement graphics. 8) *graphic economy:* Avoiding a too large number of different symbols, which would make the notation hard to learn and understand. 9) *cognitive fit:* Trying to adapt the notation to the audience, i.e. possibly using different dialects with different stakeholder groups. The 9 principles of Moody have already been used in [13] for evaluating the notations of UML, and in [14] for evaluating i*. While those papers focused on evaluating existing modelling notations, our focus is to use the 9 principles to guide proposals for new notations.

There have been several efforts presenting adaptations of diagram notations. Mendling et al. [15] propose to insert small icons inside each business process activity, but not to indicate location, rather the nature of the activity, where the authors identify 25 generic labels. Some examples of concepts represented using such iconic labels are "assess" (using a weight scales icon), "complete" (using a green filled circle with a white check mark), "decide" (using a question mark), "promise" (using a handshake symbol), and "search" (using a magnifying glass symbol) [15] (p.52).

Most closely related to this paper is work by Baumeister et al. [16], where the authors propose some extensions to UML activity diagrams specifically targeting the modelling of mobile systems. In particular, it distinguishes between what the authors call a *responsibility centered notation*, where swim-lanes are used to indicate "who" performs and action, and a *location centered notation*, where swim-lanes indicate "where" an action is performed. In the responsibility centered notation location is instead indicated by the addition of textual labels in action or actor nodes, and similarly, the responsible actor is indicated by textually labelled actor objects in the location centered notation. Since responsibility centered notation is the standard usage of swim-lanes, it is location centered notation which represents the major novelty or deviation from normal practice. This resembles a notation which was briefly tried but dismissed in [5] because the need for showing responsibility by other means created a lot of extra nodes and edges in the diagram, thus creating a messy diagram that would be hard to comprehend especially for stakeholders with limited technical skills.

Decker [17] also proposes an adapted notation of UML activity diagrams, specifically targeting access control for mobile workflows. Each activity node can be linked to one or a series of location nodes, shown by parallelograms, to indicate that it is compulsory or prohibited to perform certain activities in certain locations. [18] makes a similar proposal for BPMN diagrams, again showing locations by icons with a parallelogram shape. Thus creating separate nodes for locations, this slightly resembles the usage of UML notes to indicate locations, as investigated analytically in [5] and experimentally in [6], however with poorer user performance than an alternative approach of coloured nodes.

Walderhaug et al. [4] used UML notations extensively in the MPOWER project with homecare services and conclude that UML profiles can be used as a mechanism for tool chains based on OMG's Model Driven Architecture (MDA) and UML standards. Work on mobile ontologies by Veijalainen [19] supports the idea of the 'where' aspect as essential in mobile processes, but excludes the 'what' aspect. Larsson [20] 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.

3 Notation Alternatives

There are a number of notation alternatives that have so far not been explored in our experiments [6-7], nor by the initial analytical evaluation in [5]. Bertin [21] presents 8 different visual variables that might convey meaning in diagrams:

- *planar variables*: horizontal position, vertical position
- *retinal variables*: size, brightness (="value"), texture, colour, orientation, and shape.

As Bertin points out these can be used alone or in combination to give different meanings to diagrammatic constructs, and so far we have only explored the usage of some of these variables. Using planar variables to indicate location of activities would mean that an activity node's placement in the diagram would denote the location where the activity takes place in the real world. Instead using retinal variables, variations in the visual appearance of the activity node would indicate the place where the activity is performed. Figure 1 illustrates a spectrum of possibilities. The first row indicates the planar alternative (embedding activity nodes in pools, in a practical example there would of course be several pools and several activities in each pool), the remaining rows showing retinal alternatives, either just adding text (first row), adding an extra location node (next three rows), or providing variations of the activity node itself, e.g., in the bordering line (last four rows), fill (three rows above those), shape, size, or orientation. Many of these alternatives would not provide good notations, but are included just for completeness.

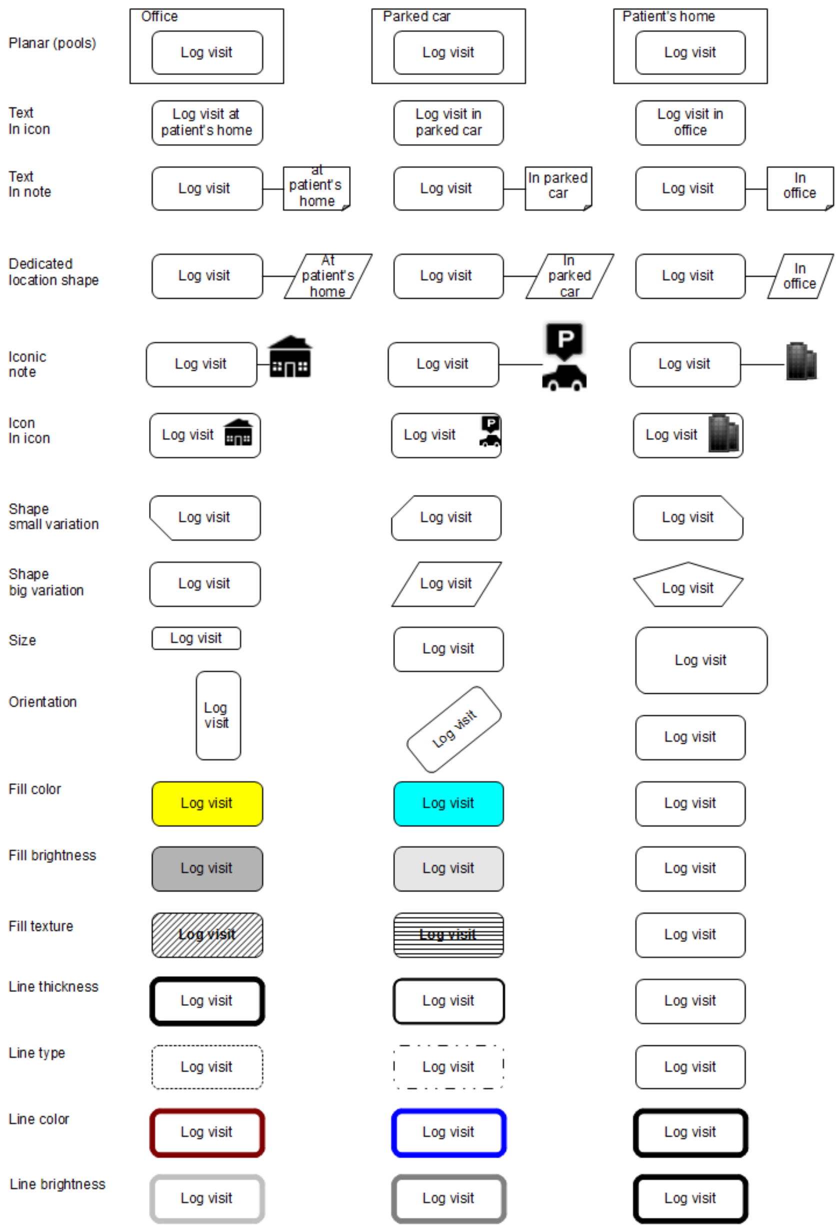


Fig. 1. Showing location by planar (top row) and retinal variables (remaining rows). Fill-colors yellow, blue, white, line colors red, blue, green (if b/w printing).

The first question is whether we should go for an entirely new notation, or a smaller adaptation of existing notation. There are of course pros and cons to both these alternatives:

- entirely new notation has the advantage that one is free to make optimal choices vs. notation design, specifically fitting the representational challenge at hand. On the other hand, more work is needed to establish the notation and supporting tools, making it less likely that the notation will be adopted by anybody.
- smaller adaptation of existing notation(s) have the advantage of leveraging the investment found in existing tools and analyst knowledge (i.e., more likely to pick up a notation which is a small adaptation of something they already know). On the other hand, there is less freedom in developing the notation since most of the choices made in existing notation must be retained, thus the notation might not be visually optimal.

In the following we will discuss these two possibilities separately, considering entirely new notations in section 3.1 and adaptations in section 3.2.

3.1 Entirely New Notations

For an entirely new notation, the usage of planar variables (horizontal and vertical placement) could be the most intuitive choice to indicate geographical location, as this the normal approach for spatial relationships in maps, which most people have somehow been accustomed to without any IS education. As discussed in [10], due to the different meta-meta model of maps and conceptual models, there are differences in how to exploit the nine principles of Moody. Cartography revolves, generally, around geographical information which is strongly reflected in the visualization used. Generally the visualization can be said to comprise three graphic primitives; point, line, and area, and relations between these. This is inherently different from meta-meta models in conceptual modelling which usually comprise only nodes and links between nodes, in addition to containment. From [10] we have the following guidelines based on the work on MAPQUAL:

1. Clearly discriminate between geographical oriented lines and conceptual lines (relationships)
2. Clearly differentiate between nodes (concept) which are often depicted by a geometric shape, and geographic areas (by texture or colour for instance)
3. Indicate topological information by positioning of conceptual nodes according to the topology where relevant.
4. Position concepts according to their temporal nearness.
5. Use visual variables where appropriate, especially the use of colour and shading for differentiation is necessary for integrated models.
6. Design the visualization based on the participants' cognitive metaphor of the most important information attribute. For instance, temporal attributes tend to be lean towards a sequential metaphor. Spatial attributes, like nearness, tend to lean towards a distance metaphor (i.e. closer is nearer).

The work in [22] illustrates that the new conceptualization might be promising, in particular when combining space and conceptual aspects. Looking on the different archetypical work-places as discussed in the start of section 2 (stationary in office, stationary visiting, travelling by own means, being transported, wandering) one often needs to differentiate between a limited number of places.

The idea of using swim-lanes for location was briefly tried out in [5] but dismissed because it created messy diagrams. If using swim-lanes pools for location, the challenge becomes how then to show which role or organizational unit is responsible for various tasks. The attempt in [5] was to use the stick-man figure, since this is anyway a well-known symbol in UML. The problem was that links then had to be drawn from every activity to some stick-man figure, and in anything beyond extremely simple process models, crossing links made the diagrams hard to read. However, considering entirely new notation, one does not have to be loyal to UML notation and use the stick man figure, instead the actors for activities can be indicated by means which do not create extra nodes or links in the diagrams, for instance:

- colour or texture. These do not have any obvious intuitive correspondence with various organizational units, thus needing a legend associated to each diagram, although one could imagine developing domain-specific conventions for various organizational units typically involved.
- iconic labels inserted in the activity nodes. These have an advantage over colour in being easier to understand intuitively, for instance HR dept = stick figure, sales = dollar, salaries = pay sack, production = gear wheel, R&D = question mark, top management = exclamation mark, etc. A downside versus colour might be if presenting huge models with limited space, when icons inside activity nodes might become small and make the diagrams appear overloaded.

Figure 2 and Figure 3 show the same simplified example of a mobile work process in municipal home care, initiated by a shift leader distributing patients on various home care assistants (e.g., depending on patient needs and available assistants, feasible geographical allocation, etc.). Each home care assistant then decides on the visiting sequence of the allocated patients and drives off on the visiting round. On the way to each patient the assistant gets preliminary information about the patient, preferably by audio to allow for sufficient attention to the driving. For each visit, the assistant takes care of the necessary task for the patient and, when returning to the parked car, immediately logs the visit, possibly also registering special needs that must be taken care of on the next visit (e.g., buying new light bulbs). In case the patient's health condition does not appear to be normal, the assistant may contact a nurse who will take further action if necessary.

In Figure 2 swim-lanes are used for denoting the different locations where this work-process is played out, while white and grey-tones for the activity nodes show the actors. Colour would be even more illustrative, but grey-tones were used here for the purpose of b/w printing. In Figure 3 bounding boxes similar to swim-lanes are again used for location, but this time the different locations are even more emphasized by using different background colour inside the bounding boxes, plus some illustrative icons (office building, driving car, parked car, homes, and red cross for the clinic) to indicate the nature of the various locations. Instead of using different grey-tones inside the activity nodes, icons are used also here, to distinguish between activities performed by the shift leader, the home care assistant, and the nurse.

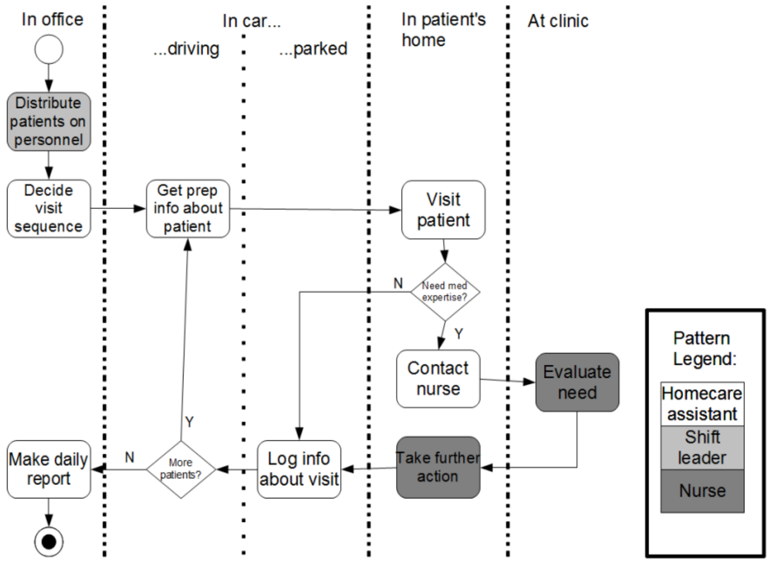


Fig. 2. Distinguishing actors by brightness (grey-tones), swim-lanes for location

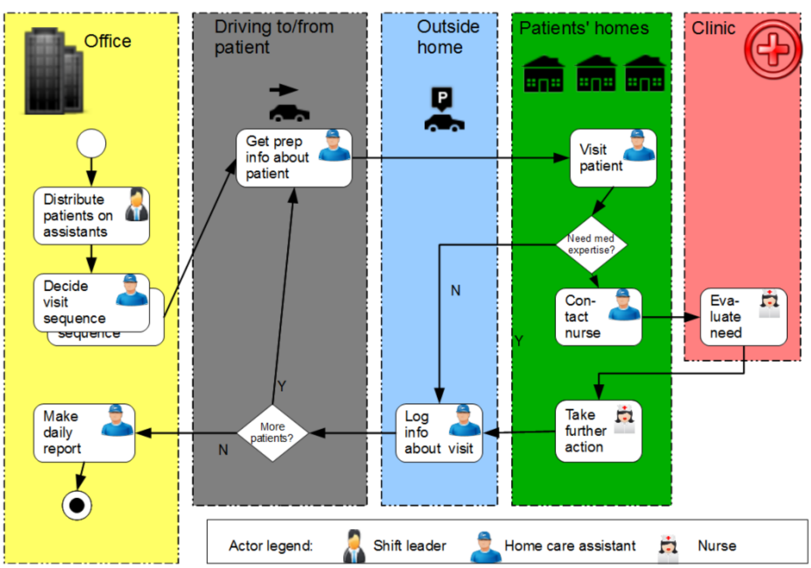


Fig. 3. Distinguishing actors by people icons, pools for location

3.2 Adapted Notations

Existing process modelling approaches like BPMN or UML Activity Diagrams tend to use the planar variables for activity order and possibly organizational placement in swim-lanes or pools. If the goal is to adapt an existing notation, it would make sense to keep using the planar variables for these purposes, so that location would instead have to be shown by retinal variables, cf. Figure 1. Many of these can be dismissed quite quickly using one or more of Moody's 9 principles. For instance, the four lower rows which use variations on the bordering line of the activity node will all be very subtle, with poor perceptual discriminability. Using size will be confusing, as this will intuitively be understood to indicate complexity or importance, not location, giving poor semantic transparency. The same would apply to orientation, intuitively indicating increase or decrease. Using totally different symbols ("shape, big variation") would be a radical change from standard notation and the user would probably be confused that such quite different symbols would all represent activities. On the other hand, the alternative "shape, small variation" suffers from poor perceptual discriminability, and worse the more locations must be distinguished. The alternatives with separate nodes have problems with regards to the principle of complexity management, since they cause a substantial increase in the number of nodes in the diagram, and thus also to the number of links between nodes, easily causing crossing lines and reduced readability. This was probably a key reason why a notation using colour performed significantly better than a baseline notation using UML notes in an experiment [6]. The advantage of using UML notes, on the other hand, is that it does not require any changes to standard UML notation.

In another experiment [7] the alternatives with colour-fills and pattern-fills performed equally well, although opinion-wise the experiment subjects had a strong preference for the colour version. Another possibility that might be tempting to try out is the "icon in icon" alternative. This shares the advantage of colour- and pattern-fills that the number of nodes and links in the diagram is not increased. If one is able to make intuitive icons, it would have an additional advantage in terms of higher semantic transparency. On the other hand, it would have a potential disadvantage for the principle of graphical economy (adding more different symbols to the notation), and in a large model with many small activity nodes, icons may become smallish and hard to read.

In Figure 4 the same home care example as in Figures 2 and 3 is shown again, now using planar variables for responsibility (swim-lanes) and activity order (top to bottom), and instead showing location by pattern fills. Colour got even better results than pattern-fills in the experimental evaluation, but pattern-fills are used as an example here for the purpose of black/white printing. Anyway, whether using patterns, colour or grey-tones, the diagrams would be structurally the same.

Figure 5 then shows the same example instead using icons to indicate the locations of actions. Four different icons are used, an office building for tasks performed in the office, a car with an arrow on top for tasks performed while driving, the same car with a P-sign above for tasks performed while parked, and a house for the patient's home. A legend might be useful for these also, but the much more intuitive relationship between these symbols and what they stand for, means that the user would likely learn and remember them more easily.

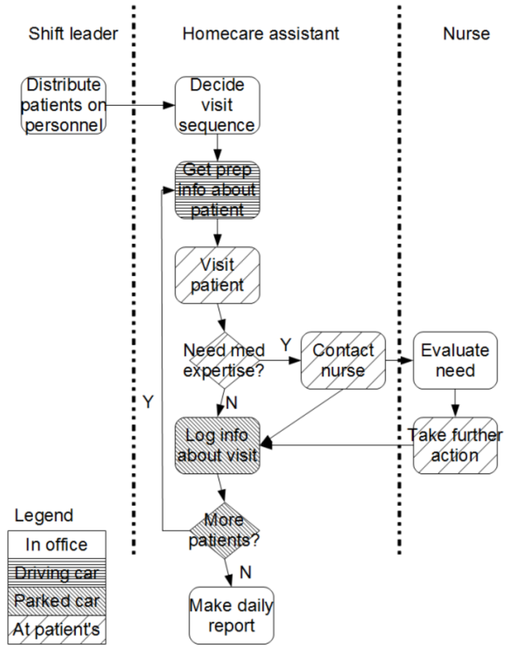


Fig. 4. Home Care Example, using pattern-fills for location

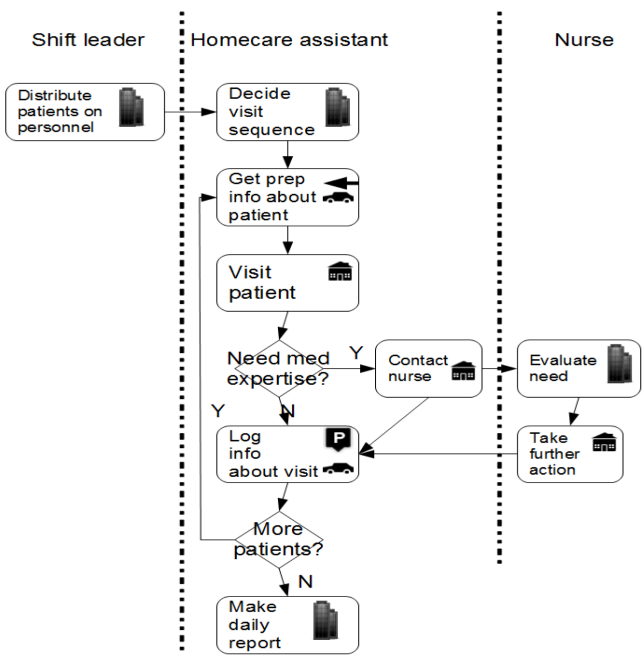


Fig. 5. Showing locations of actions by place icons inside activity nodes

4 Evaluation of Notations

Not all of Moody's 9 principles [12] are equally relevant to us. Two of the principles are excluded, namely *cognitive integration*, because it concerns relationships between several diagrams or diagram types while we are only considering activity diagrams at this point, *dual coding*, because this concerns the combination of text and visual means – which can be done fairly independent of what visual trick we use for location, and *cognitive fit,* which concerns the usage of different dialects for different stakeholder groups (where, if anything, we would need representatives from these various groups to perform the evaluations rather than doing it ourselves). This leaves us with 6 of Moody's principles, on the other hand we add one additional criterion which is not in Moody's list, namely support for *multiple locations*. In our examples in Figure 1, each activity is shows as taking place in one particular location. However, for a mobile information system some activities must be supported in many different locations, so that it is up to the user's preference where to perform it (e.g., either in the office or in the car). Some activities might even take place in several places at once, for instance if performed by two collaborating persons in different locations. The need to attach several locations to some activities might obviously cause additional challenges to our notation alternatives.

Table 1. Evaluation of notations, responsibility-centered alternatives above grey line, location-centered alternatives below

Show location by...	**SC**	**PD**	**ST**	**CM**	**VE**	**GE**	**ML**	**Sum**
Text in activity node	-	-			- -	++		-2
Text in note	-	-		- -	- -	++	+	-3
Dedicated location shape				- -	- -	-	++	-3
Iconic note	+	++	++	-	-	- -	++	+3
Icon in icon	+	++	++	+	-	- -		+3
Shape, small variation	+	-		+	- -		-	-2
Shape, big variation	+	+		+	- -	- -	- -	-3
Fill colour	+	++	-	+	+	+		+5
Fill brightness	+	+	-	+	+	+	-	+3
Fill texture	+	+		+	+	+	-	+4
Planar, resposibility by								
Iconic note	+	++	++	-	-	- -	- -	-1
Icon in icon	+	++	++	+	-	- -	- -	+1
Fill colour	+	++		+	+	+	- -	+4
Fill brightness	+	+		+	+	+	- -	+3
Fill texture	+	+	+	+	+	+	- -	+4

The rows in Table 1 are various notation alternatives for location, for space reasons we omitted some that were obviously poor already in the discussion of Figure 1. The columns are the criteria used (SC = semiotic clarity, PD = perceptual discriminability, ST = semantic transparency, CM = complexity management, VE = visual expressiveness, GE = graphical economy, and ML = multiple locations. The scores range from - - (very poor), through poor, neutral, good, to very good (++). The tows above the grey line are for notation with small adaptation, i.e., planar variables used for responsibility so location must use retinal variables. Below the grey lines are alternatives using planar variables for location, so that responsibility must be shown by retinal variables. Here we only included the 5 best alternatives from the upper half, since the relative merits of the various alternatives will be pretty much the same whether the retinal variables are supposed to depict location or responsibility.

For space reasons we cannot explain all the 105 marks in this table in detail, more information can be found in a technical report [23]. The analytical evaluation was performed by both authors independently, then going through a consensus process afterwards, to reduce the threat of scoring error and arbitrary interpretation of evaluation criteria. However, both evaluators – although having different cultural backgrounds – came from the same university, and having a four year scientific relationship (supervisor and phd student), so the results must be taken with caution.

The alternatives using fills (either colour, grey-tones or patterns) generally scored well in the evaluations, because they do not introduce new nodes or links (i.e., good for complexity management), put to use visual variables that are not used for any other purposes in the notation in question (i.e., good for visual expressiveness), and are fairly easy to discriminate visually (i.e., good for perceptual discriminability). This is especially true for colour (except for color-blind users), but brightness and texture will also be fine as long as each diagram has a limit of 4-5 different actors or locations that must be distinguished. Another good alternative is the use of meaningful icons inside the activity nodes ("icon in icon"). These have a big advantage where colour-fills and similar alternatives were not impressive, namely in being intuitively understandable, i.e. good for semantic transparency.

As for the choice between the radical, location centered notations using planar variables for location (lower part of Table 1), or the less radical responsibility centered notations using retinal variables for location (upper part), the planar alternatives might be slightly better for semantic transparency, but have problems with multiple locations, in which case activity nodes might have to be duplicated in several swim-lanes. For multiple locations, the dedicated location shape or iconic note will be the best, since an activity node can trivially be connected to a number of location nodes. Icon in icon or colour fills are not too bad either, since one can put several icons or colours inside the same node.

5 Discussion and Conclusion

Our first research question was about the most plausible visual variables to use for showing location in process models, and the analysis has indicated a two-level choice:

- showing location by planar variables, in which case responsibility must instead be shown by other means; the best seem to be fills in the activity nodes (e.g., colour, grey-tones, patterns) or meaningful icons in the activity nodes.
- Showing responsibility by planar variables, in which case location must instead be shown by retinal variables, the most interesting again seeming to be fills or meaningful icons.

The second research question was about the relative merits of radical location-centered notations versus smaller adaptations of mainstream responsibility-centered notations. The choice between these depends on several factors. If similarity to mainstream notations is important, it will be a good idea to keep showing responsibility by planar variables. On the other hand, for semantic transparency, it might be better to use planar variables for location. There is, however, a potential problem with models where some activities can be performed in multiple locations, since it would cause clumsy diagrams if activity nodes must be duplicated in several location pools. For activities supposed to be performable everywhere, there could be a workaround by designating a special "anywhere" pool, but this would not solve the problem for activities that should be supported in a couple of different places, but not all. Hence, while the usage of planar variables for location looks nice as long as the models only need to depict each activity as performed in one place, it might in the more general case not be as interesting as initially believed. From the evaluation tables, the aspect not shown by planar variables (whether this be location or organizational responsibility) can feasibly be indicated for instance by colours or similar fills inside the activity nodes, or by meaningful icons inside the activity nodes ("icon in icon") -- or, if multiple locations for each activity is a major issue, possibly putting the iconic nodes as separate nodes in their own right ("iconic node"). A major point made by Moody as reflected in the principle of cognitive fit [12], however, is that one should not have to make a choice of only one notation here. A reasonably sophisticated modelling tool should easily allow to switch between different visualizations of the same underlying conceptual model – depending on the preference of the user. So, a quick menu choice could determine whether to hide or show locations in a diagram, using a location-centered or responsibility-centered notation, whether to use colour or pattern-fills or icons - or other visual variables.

A major shortcoming of this work so far is that most of the example notations are only evaluated analytically, which does not allow for as certain conclusions as would be possible with empirical data. We have done experiments comparing the responsibility-centered “text in note”, “color-fill”, and “pattern-fill” notations, showing that color-fill performed better than notes and was more preferable to users than pattern-fills. However, the “icon in icon” alternative has not yet been tried in experiments, as have none of the location-centered alternatives. An obvious step for further work would therefore be to run more experiments, and also using more varied participant groups than have been used so far (CS students). Another interesting continuation would be case-studies with larger industrial examples.

Furthermore, it is not evident that swim-lanes or pools must be used at all. It could also be possible to experiment with notations where planar variables only indicate activity sequencing (e.g., top to bottom) or parallelism (e.g., left-right), showing both location, responsibility and other properties (e.g. time) with combinations of fills, icons, and text. After all, while swim-lanes or pools are visually appealing and have some evident advantages - for instance making it quick for each stakeholder to see what activities he is responsible for if swim-lanes are used for responsibility - they might also increase diagram complexity in many cases. For instance, activities which are subsequent and which would therefore intuitively be close in the diagram, might have to be drawn far apart because they belong to different swim-lanes - which creates longer links and potentially also more crossing links. Again, in a good modelling tool it should be possible to switch quickly between a display using swim-lanes or pools to one avoiding it - as well as to enhance or suppress any particular aspect of the model, e.g., being able to select whether the model is to show location of the activities or not. The study of tool support for the various diagram styles discussed in this paper must therefore also be a topic for further work.

References

1. Wohed, P., et al.: On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J., Sheth, A. (eds.) BPM 2006. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)
2. Dumas, M., ter Hofstede, A.: UML Activity Diagrams as a Workflow Specification Language. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 76–90. Springer, Heidelberg (2001)
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., Mikalsen, M.: Experiences from Model-Driven Development of Homecare Services: UML Profiles and Domain Models. In: Chaudron, M.R.V. (ed.) MODELS 2008. LNCS, vol. 5421, pp. 199–212. Springer, Heidelberg (2009)
5. Gopalakrishnan, S., Sindre, G.: Alternative Process Notations for Mobile Information Systems. In: Popplewell, K., et al. (eds.) Enterprise Interoperability IV. Making the Internet of the Future for the Future of Enterprise, pp. 13–23. Springer, London (2010)
6. Gopalakrishnan, S., Krogstie, J., Sindre, G.: Adapting UML Activity Diagrams for Mobile Work Process Modelling: Experimental Comparison of Two Notation Alternatives. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 145–161. Springer, Heidelberg (2010)
7. Gopalakrishnan, S., Krogstie, J., Sindre, G.: Adapted UML Activity Diagrams for Mobile Work Processes: Experimental comparison of colour and pattern fills. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 314–331. Springer, Heidelberg (2011)
8. Dourish, P.: Re-space-ing place: "place" and "space" ten years on. In: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, pp. 299–308. ACM, Banff (2006)
9. Harrison, S., Dourish, P.: Re-place-ing space: the roles of place and space in collaborative systems. In: Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, pp. 67–76. ACM, Boston (1996)
10. Nossum, A., Krogstie, J.: Integrated Quality of Models and Quality of Maps. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. LNBIP, vol. 29, pp. 264–276. Springer, Heidelberg (2009)

11. Krogstie, J.: Using a Semiotic Framework to Evaluate UML for the Development of Models of High Quality. In: Siau, K., Halpin, T. (eds.) Unified Modeling Language: Systems Analysis, Design, and Development Issues, pp. 89–106. IDEA Group Publishing, USA (2001)
12. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 35, 756–779 (2009)
13. Moody, D.L., van Hillegersberg, J.: Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In: Gašević, D., Lämmel, R., Van Wyk, E. (eds.) SLE 2008. LNCS, vol. 5452, pp. 16–34. Springer, Heidelberg (2009)
14. Moody, D.L.: Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax. In: Patrick, H., Raimundas, M. (eds.) IEEE International Conference on Requirements Engineering, pp. 171–180. IEEE, Atlanta (2009)
15. Mendling, J., Recker, J., Reijers, H.A.: On the Usage of Labels and Icons in Business Process Models. International Journal of Information System Modeling and Design 1(2), 40–58 (2010)
16. Baumeister, H., et al.: Extending Activity Diagrams to Model Mobile Systems. In: Aksit, M., Mezini, M., Unland, R. (eds.) NODe 2002. LNCS, vol. 2591, pp. 278–293. Springer, Heidelberg (2003)
17. Decker, M.: Modelling Location-Aware Access Control Constraints for Mobile Workflows with UML Activity Diagrams. In: 3rd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Sliema, Malta (2009)
18. Decker, M., et al.: Modeling Mobile Workflows with BPMN. In: 2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable, ICMB-GMR (2010)
19. Veijalainen, J.: Developing Mobile Ontologies; who, why, where, and how? In: International Conference on Mobile Data Management, pp. 398–401. IEEE, Manheim (2007)
20. Larsson, A.V.: Designing for use in a future Context – Five Case Studies in Retrospect. In: Institute of Technology. Linkoping University, Linkoping (2003)
21. Bertin, J.: Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press, Madison (1983)
22. Nossum, A.: Visual Representation of Integrated Information. Dept. of Computer and Information Science. Norwegian Technical University of Science and Technology Trondheim, Norway (2009)
23. Gopalakrishnan, S., Sindre, G.: Analytical Evaluation of Notational Adaptations to Capture Location of Activities in Process Models, Technical report M3W-1, Norwegian University of Science and Technology (NTNU): Trondheim. (2011), `http://www.idi.ntnu.no/~guttors/m3w1.pdf`

Modeling Business Strategy: A Consumer Value Perspective

Eric-Oluf Svee, Constantinos Giannoulis, and Jelena Zdravkovic

Department of Computer and Systems Sciences
Stockholm University
Forum 100, SE-164 40 Kista, Sweden
{eric-sve,constantinos,jelenaz}@dsv.su.se

Abstract. Business strategy lays out the plan of an enterprise to achieve its vision by providing value to its customers. Typically, business strategy focuses on economic value and its relevant exchanges with customers and does not directly address consumer values. However, consumer values drive customers' choices and decisions to use a product or service, and therefore should have a direct impact on business strategy. This paper explores whether and how consumer values influence business strategy, and how they might be linked to IS solutions that support the implementation of such strategies. To address these questions, the study maps consumer values to a business strategy approach via a meta-model commonly used for such purposes, based on strategy maps and balanced scorecards (SMBSC). Additionally, the applicability of the mappings is illustrated via a case scenario where the mappings are applied and the business strategy conceptualization captures them. Finally, based on these mappings, high level guidelines for linking consumer values to requirements for the development of IS solutions through business strategy conceptualization are proposed.

Keywords: Consumer value, strategy maps, balanced scorecards, requirements engineering.

1 Introduction

The ability of an enterprise to attain its vision is dependent primarily on its capability to efficiently marshal and align its resources, compete within a sustainable environment, and strive for innovation to aid in executing its business strategy. However, while the basis for much of this work is related to the interactions with the enterprise's customers, this vital and necessary connection between the two often lacks the rigor and structure necessary to elicit the desires of the customer and deliver them to the enterprise.

The importance of customer desires has long been acknowledged, as Morris [1] claims that all products provide services in their capacity to create the need of want-satisfying experiences. Drucker [2] claims that in addition to harmonizing the utilization of its resources, an enterprise should focus on the opportunities to create

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 67–81, 2011.

revenue by re-shaping the characteristics of existing services and products to satisfy its customers. Such customer desires have been typically addressed as economic values such as goods, services, money, and information. However, another perspective of customer desires, namely *consumer value*, has not been explored. A consumer value is a personal norm: a belief about what is needed, wanted, or ought to be, which serves as a guide to consumers in making decisions [3].

Therefore, any disconnection between consumer value and business strategy can prove problematic. For example, a bank might provide a robust e-banking system which is unappealing and unusable to the end-user. However, while the system satisfies the bank's needs, but not those of its customers, it will not be successfully adopted. Unfortunately, current approaches to business strategy do not explicitly capture the values regarding products and services that come directly from consumers.

The goal of this paper is to explore whether and how consumer values influence business strategy and consequently how this influence reflects on IS solutions that are operationalized to support business strategy implementation.

Given the numerous business strategy approaches available (e.g. *Strategy Maps and Balanced Scorecards* [4], the *Value Chain* [5], *Blue Ocean Strategy* [6], etc.) and the different approaches on consumer values (e.g. *Holbrook's Typology* [3], *Quantification of values* [7], *Rokeach's Value Survey* [8], *Schwartz's Value Survey (SVS)* [9], etc.), in this study we choose to explore prevalent approaches in both areas and, particularly for business strategy, approaches that include some level of conceptualization or formalism that can support traceability. Therefore, we use *Holbrook's Typology of Consumer Values* for capturing and structuring the desires and aspirations of consumers [3], and *Strategy Maps and Balanced Scorecards* as a business strategy approach [4], which is conceptualized in the means of a meta-model [10].

This paper is structured as follows: Section 2 provides an overview of Holbrook's Typology of Consumer Values as well as Strategy Maps and Balanced Scorecards (SMBSC). Section 3 presents our contribution, the mappings of Holbrook's consumer values to Strategy Maps and the SMBSC meta-model. Section 4 illustrates our contribution using a case scenario of a shopping mall. Section 5 outlines the links of our contribution to requirements engineering and Section 6 summarizes our conclusions and indicates the steps forward.

2 Related Work

2.1 Consumer Values

Holbrook refines the value concept, focusing on those held by individuals during a value exchange: consumer values. Consumer values are "an interactive, relativistic preference experience" [3]. Unpacking this definition, one finds that *interactive* entails an interaction between some subject and an object. Next, *relative* refers to consumer values being comparative, involving preferences among objects, varying across people, and they are situational, or context specific. *Preferential* refers to consumer values embodying the outcome of an evaluative judgment, where something is more or less desirable in comparison to something else, and finally *experience* refers to consumer values not residing in the product purchased, brand chosen, or object possessed, but in the consumption experience. This allows for a rather expansive view of value, because all products provide services in their capacity to create need- or want- satisfying experiences.

Dimensions of Consumer Values. Holbrook identifies three dimensions in consumer values [3].

Extrinsic/Intrinsic. Extrinsic is a means/end relationship wherein consumption is prized for its functional, utilitarian ability to serve as a means to accomplish some further purpose, aim, goal, or objective. For example, hammers are generally prized for their utility and not their beauty. *Intrinsic* occurs when some consumption experience is appreciated as an end in itself—for its own sake—such as listening to music.

Orientation. Self-oriented is when some aspect of consumption is cherished, either selfishly or prudently, for the individual's sake; a sweater has value partly because it keeps its owner warm. *Other-oriented* is where the consumption experience or the product on which it depends is valued by others, either beyond the subject, for its own sake, for how they react to it, or for the effect it has on them. For example, a Ferrari might be purchased to impress someone's co-workers.

Activity. Active entails a physical or mental manipulation of some tangible or intangible object, involving things done by a consumer to or with a product as part of some consumption experience: driving a Ferrari is part of what makes owning one desirable. *Reactive* results from apprehending, appreciating, admiring, or otherwise responding to an object, when the object acts upon the subject. Similar to the example given for intrinsic, listening to music can also be reactive, when the consumptive act is driven by the object and not the subject.

Typology of Consumer Values. Based on these three dimensions, Holbrook creates his Typology of Consumer Values where each archetype (author's term) represents a distinct type of value in the consumption experience [3].

Efficiency results from the active use of a product or consumption experience as a means to achieve some self-oriented purpose. This is a utilitarian value: recall that hammers are generally prized for their usefulness and not their beauty. Often, Efficiency is measured as a ratio of outputs to inputs. For example, the Efficiency of an automobile can be assessed as some ratio of distance traveled to volume of fuel used. Another view of Efficiency—convenience—is often a measure of utility derived versus time or energy expended. With *Excellence*, one admires some object or prizes some experience for its capacity to accomplish some goal or to perform some function: a Ferrari is a wonderful choice for a high-quality automobile, though a poor choice for a delivery truck. *Status* is sought by adjusting consumption in a manner that affects those whom one wishes to influence, consuming products or engaging in consumption experiences so as to project a particular type of image one wishes to portray. In other words, consumption is about communicating about ones' self to others in ways that contribute to success. Status and *Esteem* are intimately interrelated, with Esteem the reactive counterpart to Status. Esteem tends to result from a somewhat passive ownership of possessions appreciated as a means to building one's reputation with others. In other words, Esteem is about the reactive appreciation of consumption or lifestyle in a somewhat passive way as a potential extrinsic means to enhancing my other-oriented public image. *Play* is a self-oriented experience— actively sought and enjoyed for its own sake — and as such, typically involves having fun. *Aesthetics* refers to an appreciation of some consumption experience valued intrinsically as a self-oriented end in itself. The hallmark of this value is that it is enjoyed for its own sake, without a need for external justification. *Ethics* involves doing something for the sake of others— that is, with a concern for how it will affect them or how they will react to it— where such consumption experiences are valued for their own sake as ends in themselves. Donating blood is an activity that is not often done for selfish benefits, but rather is most often motivated by selfless reasons. *Spirituality* entails an intrinsically motivated acceptance, adoption, appreciation, admiration, or adoration of an Other where this "Other" may constitute some Divine Power, or some otherwise inaccessible Inner Being. Such an experience is sought not as a means to an ulterior end but rather as an end in itself prized for its own sake. People often donate money to churches for the sole purpose of feeling closer to just such an Other.

2.2 Strategy Maps and Balanced Scorecards

A strategy map is a general representation of the four organizational perspectives of the balanced scorecards in a cause-effect manner and facilitates the communication of direction and priorities across the enterprise according to [11].

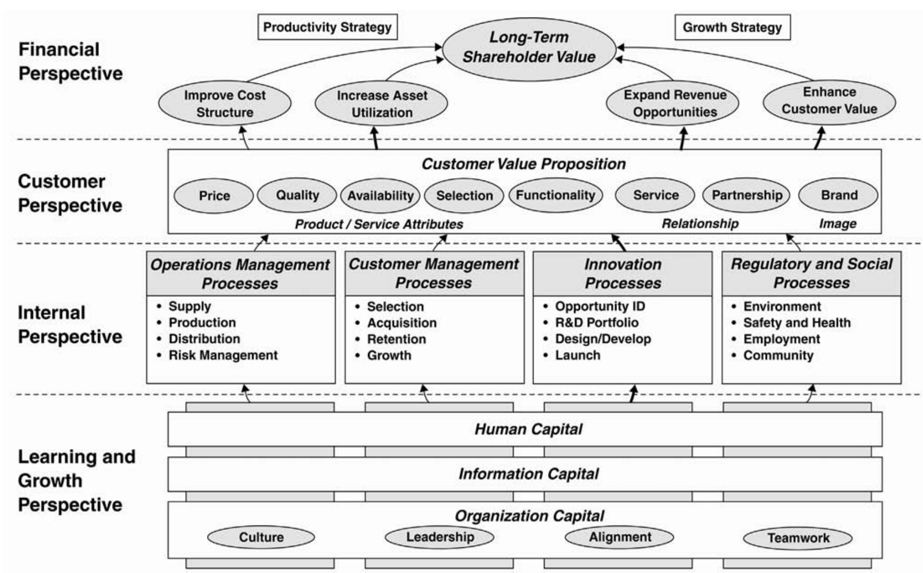


Fig. 1. The Strategy Map template [4]

Strategy balances long-term financial commitments aims at profitable revenue growth and short-term financial commitments aiming at cost reductions and productivity improvements (financial perspective). Strategy is based on a differentiated and clearly articulated customer value proposition (customer perspective). Value is created through focused, effective, and aligned internal business processes grouped into four clusters: operations management, customer management, innovation, and regulatory-social (internal perspective). 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, which includes human, information, and organization (learning and growth perspective).

A strategy map serves as a mediator between the mission, core values, the vision, and the strategy of an enterprise to the work performed. Kaplan and Norton have proposed a template for strategy maps representing how an organization can create value (Figure 1). Starting from a mission statement and core values, a strategic vision is defined, which should project the organization's overall goal. A set of goals are defined and initially grouped within the financial and customer perspectives. 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, which through cause-effect relationships, support goals at the customer perspective. Similarly, goals are set for all groups of capital referring to particular asset categories defining desired competencies, capabilities needed to support internal processes [12].

Scorecards consist of strategic objectives and related measures, which include concrete targets and initiatives towards their achievement [13] and are structured with cause-effect links/assumptions whose monitoring and assessment is essential for identifying interdependencies across an organization. According to [14], a balanced scorecard 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. When the four perspectives are addressed, providing complete coverage of business processes, a scorecard is considered balanced. While the time aspect is addressed indirectly via short-term targets set and also via the bottom up view of the four perspectives suggesting that what lies on the bottom is the outcome of planning at the top and is a prerequisite. Additionally a scorecard is also considered balanced because it covers both the internal as well as the external aspects of an enterprise.

3 Enriching Business Strategy with Consumer Values

In this section we present how strategy maps can be enriched by consumer values. In the first part we link strategy map goals to Holbrook's consumer value typology while in the second part we extend the SMBSC meta-model to incorporate goals reflecting consumer values.

3.1 Relating Consumer Values to Strategy Maps

Anticipating customers' needs is addressed in the customer perspective of strategy maps, where the customer value proposition is defined. Kaplan and Norton identify four generic strategies for defining the customer value proposition [4]. These are: *Low Total Cost*, where products and services are offered that are consistent, timely, and low-cost; *Product Leadership*, which offers products and services that expand existing performance boundaries into the highly desirable; *Complete Customer Solutions*, designed to provide the best total solution to our customers; and *System Lock-In*, where high switching costs maintain end-users as customers. Additionally, for each customer value proposition strategy a set of generic goals is defined.

In this study, we analyze Complete Customer Solutions as a possible example of a customer value proposition, which emphasizes building long-lasting relationships with customers [4].

The customer value proposition of Complete Customer Solution aims to provide the best total solution to customers, and it includes the following general goals (originally named objectives): Quality of Solutions Provided to Customers, Number of Products and Services per Customer, Customer Retention, and Lifetime Customer Profitability. Therefore, we analyze them one by one and identify which consumer values are relevant to the enhancement of strategy maps, and those which are not.

Table 1. Customer Perspective Goals for Complete Customer Solutions and their related Consumer Values

Consumer Value	**Customer Perspective Goals**			
	Number of Products & Services per Customer	Quality of Solutions Provided to Customers	Lifetime Customer Profitability	Customer Retention
Efficiency	√	√	√	√
Excellence	√	√	√	√
Play		√	√	√
Aesthetics		√	√	√
Status				√
Esteem		√	√	√
Ethics				√
Spirituality				√

Number of Products and Services per Customer. Superior product functionality, relating to matters such as speed, accuracy, and power, lies at the heart of high quality products. Two of Holbrook's consumer values are directly related to this: *Efficiency* and *Excellence*. From the consumer perspective, such a product is most likely to be both *Efficient*, where its active use allows for accomplishing some self-oriented purpose, as well as *Excellent*, where the product is admired for its intrinsic abilities to accomplish some end. An automobile allows someone to travel more quickly (Efficient) but the experience of driving one of high quality has a different level of appreciation for the consumer (Excellent) than driving one of lower quality.

Quality of Solutions Provided to Customers and Lifetime Customer Profitability. These objectives entail a deep understanding about what customers value, and combine it with the ability to bundle products into individually customized solutions. These objectives speak to the completeness of a solution, and are most closely related to the Holbrook's archetypes *Efficiency*, *Excellence*, *Play*, *Aesthetics*, and *Esteem*.

As an example, the utilitarian focus of Efficiency can be found in the time savings a customer would accrue by shopping via the new online portal of a business as opposed to the brick-and-mortar location of a competitor. For example, Amazon grew to prominence by being the prime mover in a retail space (book selling) that had not undergone a shift to online sales. Amazon now offers many products for sale, and has numerous purchasing and delivery options from physical goods to music downloads. It offers a complete, individually tailored solution to its customers, one that continues to grow and evolve over time.

Excellence is also utilitarian, but, according to Holbrook's dimensions, it is reactive rather than active. In comparing the actual experience with the expectations for that experience, Excellence relates closely to the concept of satisfaction and appears to constitute the essence of what is generally understood as quality [3]. In the case of a purchasing a book from Amazon, the customer could experience an innate sense of satisfaction based on the quality of the entire shopping experience, taking into account the efficiency-accrued time savings, among other benefits.

Play and its counterpart Aesthetics could also be related to high performance products. Product enjoyment, either based on direct use of the product or via an appreciation of its overall design, is seen as relating to these. Apple products command a premium in the marketplace, to a large extent because consumers derive pleasure both from using their well-designed products, as well as having a general appreciation for their appearance and Aesthetics.

Esteem is an internally directed idea, whereby the consumer has a self-congratulatory moment for having purchased a particular item, similar to the saying that "No one was ever fired for hiring IBM". Although the veracity of the statement might be questioned, the general premise is that, while IBM is not known for producing the most technologically advanced, the most powerful, or the fastest products, it is widely respected for providing a complete customer solution—from hardware, to software, to training—delivered in a customized solution.

Customer Retention. As Kaplan and Norton state, this objective is about the quality of relationships. Exceptional service is involved, making it critical that the organization has access to a diverse array of capabilities to develop the best means to serve their customers. Thus, all eight consumer values are needed to acquire and develop new customer segments, as well as to enrich and deepen those relationships. In addition to the five that have been previously discussed (*Efficiency*, *Excellence*, *Play*, *Aesthetics* and *Esteem*), the three remaining (*Status*, *Ethics*, *Spirituality*) can also play an important role in the process.

Status is important to this segment, as early adopters are one of the primary targets for customer engagement. By displaying the latest and greatest, consumers are often seeking to display various personal characteristics (intelligence, success) through the medium of the object.

Ethics and *Spirituality* can be very powerful consumer values, but they can be difficult to work with and must be handled with great care and delicacy, particularly Spirituality. And while it might be considered a great leap to include them in this objective, it is important to view them solely as possible means to achieve it. Whereas *Ethics* is driven by a selfless motivation, *Spirituality* is self-directed.

3.2 Consumer Value-Enhanced SMBSC Meta-model

In [10] the conceptualization of strategy maps and balanced scorecards in the means of a meta-model was presented, accompanied by a set of constraints. To support the mappings of consumer values to the four strategies of the customer value proposition proposed by Kaplan and Norton, the SMBSC meta-model must be updated.

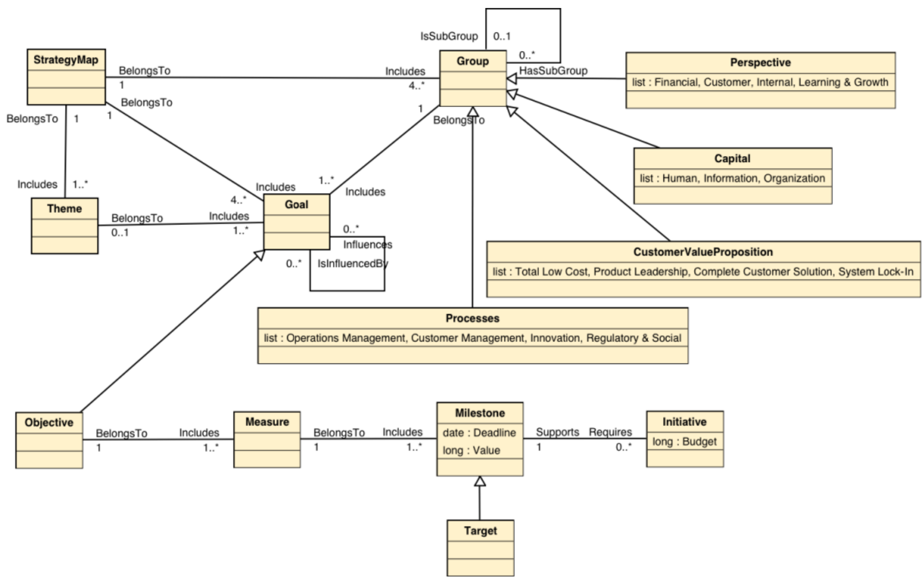


Fig. 2. The updated SMBSC meta-model

The updates refer to the substitution of the *GroupType* class into concrete groupings found in the template and Kaplan and Norton's original writings and refer to sub-groups of perspectives (e.g. for the customer perspective sub-groupings include the four customer value propositions). These sub-groupings are accompanied by additional constraints. For example an instance of *CustomerValueProposition* can only be a sub-group of *Perspective:Customer*, while an instance of the *Processes* can only be a sub-group of *Perspective:Internal*, etc. Figure 2 presents the updated SMBSC meta-model.

The four different strategies for defining Customer Value Proposition [4] — *Low Total Cost, Product Leadership, Complete Customer Solution* and *System Lock-in*] — are introduced in the SMBSC meta-model through a specialization of the class *Group*, the *Customer Value Proposition* class. The *Customer Value Proposition* class is constrained through the *IsSubGroup* association of the *Group* class to be sub-group of a group which is a *Perspective* and particularly of the type *Customer*.

In accordance to the strategy map template, the generic goals proposed by Kaplan and Norton reflect the customer value proposition chosen and constitute a grouping of goals within the customer perspective. In the meta-model, this is captured by each generic *Goal* belonging to the group *CustomerValueProposition*, which can be of type of one of the four strategies defined. Moreover, the group *CustomerValueProposition*, through the *IsSubGroup* association, is a sub-group of the group *Perspective* which is of type *Customer*. Therefore, similarly to the previous section, we examine product leadership, which includes three generic goals: First to Market, High Performance Products, and New Market Segments. Each of these goals are instances of the *Goal* class belonging to the instance of *Complete Customer Solution* of the *Customer Value Proposition* class which is a *Group*, and through the *IsSubGroup* association it is a sub-group of *Customer*, which is an instance of *Perspective* which is a *Group*.

Based on the mappings of consumer values to customer value proposition strategies provided in 3.1, the SMBSC meta-model provides traceability of consumer values from strategy maps (goals) to balanced scorecards. A *Goal* is expressed as an *Objective* when an appropriate *Measure* to demonstrate its achievement can be defined. An *Initiative* encompasses all actions identified as required towards the achievement of the objective, within the constraints of given *Milestones* and *Targets*. Additionally, mapping consumer values to goals in strategy maps makes them potentially measurable (not all goals are measurable), allowing the derivation of initiatives through balanced scorecards.

4 Case Scenario

In this section a case scenario is used to illustrate our contribution: a general strategy map of a shopping mall that has been enhanced by Holbrook's Typology of Consumer Values. To accomplish this, the customer objectives for Kaplan and Norton's strategy map template for Product Leadership are applied to a scenario that explores the development of a shopping map [15] along with and their related consumer values

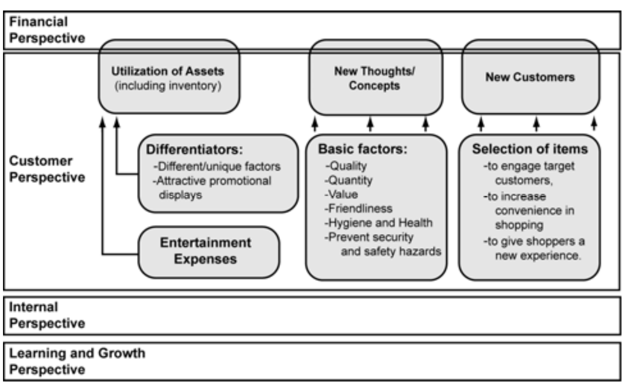


Fig. 3. Excerpt from the shopping mall strategy map found in [15]

and mappings as proposed in earlier. The strategy map from the shopping mall (Figure 3) is used to illustrate the proposed mappings, and more specifically to illustrate whether, and how, consumer values influence product leadership.

So what exactly would a strategy map look like for a shopping mall that wanted to become a product leader by adopting Kaplan and Norton's strategy map? According to the case scenario the mall's developers already plan to change their value proposition from an operational experience to one of customer convenience and intimacy. The updated customer value proposition for such a shopping mall would be that "the mall differentiates itself by offering a unique shopping experience, built on customer convenience and intimacy, occurring under one roof". However, the developers of the property did not made any provisions for incorporating what consumers actually want their experience to be in the shopping mall into the strategy map that they are creating.

4.1 Mapping Consumer Values to the Case Scenario

For the shopping mall case, we are mapping the consumer values from [16] to the Complete Customer Solution goals found in [15]. Table 2 first displays a consumer value from Holbrook, followed by a shopping value that exemplifies it. This in turn is supported by examples from marketing research, and these in turn are compared to the strategy map provided in the scenario. These were then judged on their level of implementation with three possible marks: a '-' was given if the shopping value was completely missing from the strategy map, a '*' was given if a shopping value was partially missing from the strategy map, and a '+' was given if there was a complete implementation within the strategy map.

Of the 31 shopping values presented, nine were found to have been completely implemented in the strategy map (29%), 14 were partially implemented (45%) and eight were not implemented at all (26%). This leaves 71% as an area for at least partial improvement, a significant number. And as shown previously, the close linkage between consumer values and consumer intent provides a compelling case for implementing a consumer value driven strategy; if their values are taken into account, consumers are more likely to make purchases. The ability to convert these values into concrete goals and objectives which the business can deliver on, often through improved alignment with IT, is enhanced through the use of the new meta-model.

Example of a value completely missing from the Strategy Map. Consumers were clear that, in terms of an efficient use of their resources (time, effort, money, etc) a key concern was access to the mall. Issues such as transportation to the mall, traffic encountered in the journey, and parking, were all important issues that the research highlighted. These are not addressed anywhere in the original strategy map.

Example of a value partially missing from the Strategy Map. The scenario speaks to high levels of staff training and has an explicit goal of Friendliness. While a laudable goal, details found in the consumer research reveal the 'right' way to implement such a strategy. Consumers stated that they did not want to spend energy dealing with pushy salespeople. However, unless this value is captured and populated further in the development process, this valuable detail could be lost. There is a fine line between friendly and obsequious, and care must be taken to ensure that the true intentions of the consumer are addressed.

Table 2. Mapping Consumer Values to the Shopping Mall Case Scenario

Consumer Value	Shopping Value	Mall Shopping Value	Shopping Mall Strategy Map	Complete Customer Solutions
Efficiency	*Convenience*	One stop shopping	+	# of Products
		Comparison shopping	*	# of Products
		Multi-purpose shopping	*	Quality
	Resources (Time/Effort Money)	Transportation	-	Quality
		Traffic	-	Quality
		Parking	-	Quality
		Time spent in mall	*	Quality
		Pushy Sales people	*	Quality
		Finding desired product	+	Quality
		Waiting in check-out lines	*	Quality
Excellence	*Customer Service*	Human contact	*	Retention
		Safe & secure shopping	+	Quality
	Product Performance	Quality	+	Quality
		Selection	+	# of Products
		Price	+	Quality
Play	*Sensory stimulation*	Appeal to five senses	+	Retention
		Instant gratification	*	# of Products
		Entertainment centers	*	Retention
		Cinema	-	Retention
		Games	-	Retention
		Eateries	+	Retention
		Special events/exhibits	*	Retention
		Walking for exercise	-	Retention
		Window shopping	*	Quality
	Social Interaction	People watching	*	Quality
		Socializing with friends	*	Quality
		Talking with other shoppers	*	Quality
		Escaping from routine	+	Quality
Aesthetics	*Ambience*	Architecture	-	Retention
		Interiors	-	Retention
		Visual display	*	Retention

Example of a value completely implemented by the Strategy Map. Ideas about Quality, Selection, and Price cut across many aspects of the shopping mall strategy map, and are also directly applicable to the generic Complete Customer Solutions map. In fact, these are some of the primary drivers behind each of the three customer objectives found in that map: First to Market, High Performance Products, and New Customer Segments.

Esteem is closely related to Status, differing only slightly in its perspective. In the scenario, the customer experience is designed to be one that the customer appreciates deeply on several levels: the shopping mall is family-friendly environment in which to easily and efficiently acquire high-quality items at competitive prices. This complete experience is something that the consumer could appreciate reflectively, taking pleasure from the superior physical environment as well as their own business acumen: how smart of them for getting such a good deal. The developer of the shopping mall could decide to build on this experience, perhaps adding components to the experience that could develop Status (e.g., higher end retailers).

4.2 Instantiating the Meta-model for the Shopping Mall

The shopping mall strategy map (Figure 3) is in accordance to Kaplan and Norton's template and its conceptualization, the SMBSC meta-model [10] adhering to the constraints defined. For example, the *New Thoughts/Concepts*, which is a group of goals (product/service attributes in the original SM template) within the customer perspective, then each goal on *Quality, Quantity, Value, Friendliness, Hygiene & Health, Prevent security & safety hazard*s belongs to the group *New Thoughts/Concepts* of GroupType *Service Attribute* (one of the groupings identified within the original SM template), which is a sub-group of the Perspective *Customer Perspective*. Each of these goals belongs to the shopping mall strategy map.

To illustrate how the meta-model can instantiate the mappings of section 4.1 we use a concrete example of a consumer value. Therefore, based on [16] and table 3, one stop shopping is identified as a mall shopping value, related to the shopping value of Convenience which is mapped to the consumer value of Efficiency and it is related to the generic goal of Number of Products per Customer. Therefore, one stop shopping is a goal:

- belonging to the shopping mall strategy map, through the *BelongsTo* association to *StrategyMap*,
- influencing the goal Number of Products per Customer, through the Influences association to Goal,
- belonging to the group Customer Value Proposition of type Complete Customer Solution, which through the *IsSubGroup* association to Group, is a subgroup of the group Perspective of type Customer.

5 Link to Requirements Engineering

Once a SMBSC model is created, it can be used to guide activities for an IS department via high-level requirements for new software solutions that will support the operationalization of the created strategy model. One possible approach here is to map the strategy concepts to those of a certain Requirements Engineering (RE) technique, such as Goal-Oriented Requirements Engineering, (GORE) [17], Scenario-Oriented RE (SCORE) [18], among others.

In the SMBSC framework the notion of goal is used to capture the aims of a company from the four different perspectives. In some of the requirements engineering techniques, such as in GORE, the notion of system goal stating the intent underpinning a system service, or a quality/constraint on the service provisioning, is used as a starting point when discovering system requirements. Thereby, even the goals are considered differently in business strategy and requirements engineering approaches, a possible starting point for propagating the results of a SMBSC model to a requirements engineering model is through goals. The effect of the alignment is two-fold: from the business strategy perspective, the outlined strategies are aligned with IS projects, and from the IS development perspective, requirements are associated with both system and business strategy goals, allowing either to establish and trace the rationales behind IS projects.

Within the scope of SMBSC, a *Goal* is expressed as an *Objective* when an appropriate *Measure* to demonstrate its achievement can be defined (see Figure 2). An *Initiative* then encompasses all actions identified as required towards the achievement of the objective, within the constraints of given *Milestones* and *Targets*. After having identified initiatives, they could be analyzed for the realization through ICT, that is, consider how an initiative can be facilitated completely or to a certain extent by an IS solution. These considerations are defined as the system goals in GORE approaches, expressing stakeholders' needs and intentions, particularly for the early phases of requirements engineering [19].

After having identified initiatives by means of SMBSC, there is a need to decompose them and elicit system-related goals. We suggest a set of the following guidelines to be used to carry out the decomposition:

1. Elicit system goals concerning the *planning* of an initiative
2. Elicit system goals concerning the *execution* of an initiative
3. Elicit system goals concerning the *control of the execution* of an initiative

Any of the guidelines above may result in the elicitation of zero or more system goals supporting a given initiative from a SMBSC model. As an illustration, for the shopping mall case described in the previous section, the initiative *Malls use modern technology in inventory management* stems from goals derived from mapping to the consumer value of efficiency, which itself is mapped to the shopping value is convenience and addresses the desire for *one-stop shopping* [16]. This initiative can be supported, by following Guideline 2, via the IS goal *Facilitate automated inventory management*, which may be further operationalized with the functional system requirement *The system shall store and retrieve the information on the available good items*. In addition, the given initiative can be supported, by following Guideline 3, by the IS goal *Facilitate an automated control of the minimal stock-level.*

The above outlined approach constitutes a basis to identify goals for IT projects aligned with the outcomes of a SMBSC analysis. In addition, two other aspects of

SMBSC can be considered when linking a consumer-centric strategy to a requirements engineering context: a) in SMBSC, objectives are related to measures, targets and milestones, which can be used to ensure and control the results of related IT projects, and b) the cause-and-effect links (i.e. dependencies) among the organizational perspectives in SMBSC propagate through *Goals* toward *Objectives* and further to *Initiatives*, and as such allow tracing the inter- and intra- dependencies of IT projects.

6 Conclusions and Future Work

To address the necessary link between consumers and the businesses that serve them as acknowledged in the introduction, we have proposed an integration of consumer values with business strategy. Particularly we have set an effort to relate Holbrook's Typology of Consumer Values [3] with Kaplan and Norton's Strategy Maps and Balanced Scorecards [4]. Holbrook's consumer values were mapped to the generic goals of a particular customer value proposition, Complete Customer Solutions, supported by the extended version of the SMBSC meta-model. Moreover, we have illustrated the applicability of the proposed mappings using the case scenario of a shopping mall. A strategy map, adhering to the complete customer solutions customer value proposition, was enriched with consumer values collected for shopping mall. Supported by the extended SMBSC meta-model and the traceability it provides, we have proposed guidelines for linking the initiatives derived from the refined strategy map requirements for the development of relevant IS solutions.

Steps forward in our research include exploring all four customer value proposition alternatives identified within the consumer perspective of strategy maps. We aim to explore other consumer value typologies and develop mappings to more business strategy approaches, while also evaluating their benefits within requirements engineering.

References

1. Morris, R.T.: The Theory of Consumer Demand. Yale University Press, New Haven (1941)
2. Drucker, F.P.: Management: Tasks, Responsibilities, Practices. Harper Paperbacks (1993)
3. Holbrook, M.B.: Consumer Value: A Framework for Analysis and Research. Routledge, London (1998)
4. Kaplan, R., Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston (2004)
5. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance. Free Press, New York (1985)
6. Mauborgne, R., Kim, W.C.: Blue Ocean Strategy: From Theory to Practice. California Management Review 47(3), 105–122 (2005)
7. Frondizi, R.: What is Value? An Introduction to Axiology, 2nd edn. Open Court Publishing Company, La Salle (1971)

8. Rokeach, M.: Beliefs, Attitudes and Values. A Theory of Organization and Change. Jossey-Bass Publishers, San Francisco (1968)
9. Schwartz, S.H.: Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries. In: Zanna, M. (ed.) Advances in Experimental Social Psychology, San Diego, vol. 25 (1992)
10. Giannoulis, C., Petit, M., Zdravkovic, J.: Modeling Business Strategy: A Metamodel of Strategy Maps and Balance Scorecards. In: 5th IEEE International Conference on Research Challenges in Information Science (RCIS 2011). IEEE, New York (2011)
11. Kaplan, R., Norton, D.P.: The strategy map: guide to aligning intangible assets. Strategy & Leadership 32, 454 (2004)
12. Kaplan, R., Norton, D.P.: Having trouble with your strategy? Then map it. Harvard Business Review 78, 167–176 (2000)
13. Kaplan, R., Norton, D.P.: The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
14. Olve, N.G., Petri, C.J., Roy, J., Roy, S.: Making Scorecards Actionable: Balancing Strategy and Control. John Wiley & Sons Ltd., West Sussex (2003)
15. Prasad, K.: Strategic Management: Text and Cases. PHI Learning Limited, New Delhi (2009)
16. Kim, Y.: Consumer Value: an application to mall and Internet shopping. Retail and Distribution Management 30(12), 595–602 (2002)
17. 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)
18. Anton, A.I., Potts, C.: A Representational Framework for Scenarios of System Use. Requirements Engineering 3(3), 219–241 (1998)
19. 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 D.C (1997)

Reasoning with Key Performance Indicators

Daniele Barone[1], Lei Jiang[1], Daniel Amyot[2], and John Mylopoulos[1]

[1] Department of Computer Science, University of Toronto, Toronto, ON, Canada
{barone,leijiang,jm}@cs.toronto.edu
[2] EECS, University of Ottawa, Ottawa, ON, Canada
damyot@eecs.uottawa.ca

Abstract. Business organizations continuously monitor their environments, looking out for opportunities and threats that may help/hinder the fulfilment of their objectives. We are interested in strategic business models that support such governance activities. In this paper, we focus on the concept of composite indicator and show how it can be used as basic building block for strategic business models that support evaluation and decision-making. The main results of this paper include techniques and algorithms for deriving values for composite indicators, when the relationship between a composite indicator and its component indicators cannot be fully described using well-defined mathematical functions. Evaluation of our proposal includes an implemented Eclipse-based prototype tool supporting these techniques and two ongoing case studies.

Keywords: Business intelligence, Business model, Conceptual modeling languages, Key performance indicators, Strategic planning.

1 Introduction

An indicator, or more precisely a Key Performance Indicator (KPI), is an industry term for a measure or metric that evaluates performance with respect to some objective. Indicators are used routinely by organizations to measure both success and quality in fulfilling strategic goals, enacting processes, or delivering products/services. For example, the indicator "Percentage increase of customer-base" may be appropriate for the goal "Increase market share", while "Average duration" might serve as indicator for the activity "Loan application".

Indicators constitute an important element of business modelling as they offer criteria for determining whether an organization is fulfilling its objectives, be they strategic goals, quality requirements, or production targets. Nowadays, they also see applications in other areas. In Requirements Engineering (RE), indicators have been used to evaluate the degree of fulfillment of goals [15], whereas in self-adaptive software systems they serve as monitored variables that determine whether a system is doing well relative to its mandate, or whether it should adapt its behaviour [14].

To choose the right indicators for a given object, be it a goal, process or product, one must have a good understanding of what is important to the organization. Moreover, this importance is generally contextual. For instance, indicators useful to a finance team may be inappropriate for a sales force. Because

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 82–96, 2011.

of the need to develop a good understanding of what is important, performance indicators are closely associated with techniques for assessing the present state of the business. A very common method for choosing indicators is to apply a management framework such as the Balanced Scorecard [8], whereby indicators measure a range of factors in a business, rather than a single one (e.g., profits).

The objects that indicators assess are generally composite, consisting of hierarchies of elements. For instance, goals are usually modelled as AND/OR tree hierarchies of sub-goals to reflect a reductionist view of problem solving. Likewise, processes are usually defined in terms of sub-processes ultimately reduced to atomic actions that an agent can perform, and products are modelled as aggregates of simpler parts that are themselves composite objects amenable to further decompositions. Alternatively – and orthogonally to the examples above – a process/product may be a root node of a taxonomy tree that defines specializations. Of course, the value of an indicator for an object should depend on the values of indicators for objects one level lower in the hierarchy. Unfortunately, there are no guidelines on what this dependency is and how to define it consistently for a given business model.

In this paper, we focus on *composite* indicators, which are indicators whose values are obtained from those of their *components*. These components themselves may also be composite, leading to a hierarchy of indicators. We are interested in the problem of propagating values of indicators from a lower level in a hierarchy to ones higher up, much like the label propagation in goal reasoning [1,5]. This type of analysis is essential for calculating / deriving values for composite indicators. This is a non-trivial problem, since in many cases there is no well-defined mathematical function that relates component indicators to a composite one. This might simply be due to lack of knowledge about the indicators, or the intrinsic nature of the indicators at hand. The main contributions of this paper include: i) different techniques and algorithms for deriving values of composite indicators, especially when the relationship between a composite indicator and its components cannot be fully described using well-defined mathematical functions, and ii) an Eclipse-based prototype tool supporting these techniques. In particular, this is an extended and improved version of [2], with additional material and examples on how to reason with composite indicators.

This research is conducted in the context of the Business Intelligence Network, a Canada-wide strategic research network. Our long-term objective within the network is to develop a conceptual modelling language, called Business Intelligence Model (BIM) [3], for modelling business objectives, processes and objects in order to support business intelligence activities.

The rest of the paper is structured as follows. Sections 2 presents key concepts for strategic business models. Section 3 introduces the concept of indicator and how it can be used to evaluate goals and situations. Section 4 presents three techniques to derive values of composite indicators using different estimation / approximation techniques. Section 5 briefly presents an Eclipse-based prototype tool that supports these techniques. Finally, Sections 6 and 7 discuss related work and conclusions, respectively.

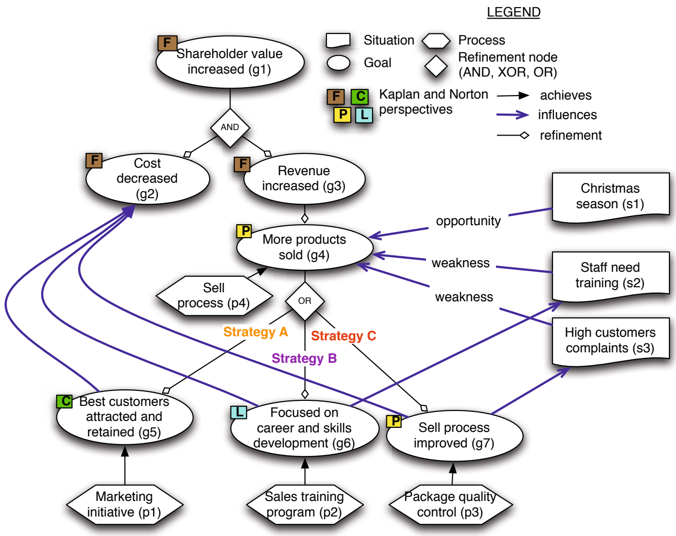


Fig. 1. Examples of goals, situations and influences

2 Strategic Business Models

In this section, we review some of the key concepts used in BIM [3] to support strategic business modelling and reasoning about strengths, weaknesses, opportunities and threats (popularly known as SWOT). Technical details about these concepts, including semantics, are presented in [7].

A *goal* (also intention, objective, vision, mission) represents a desired state-of-affairs, defined during strategic planning and pursued (hopefully successfully) during business operation. The most basic characteristics of goals include: i) a goal may be AND/OR-refined into subgoals so that its *satisfaction* level depends on that of its subgoals; ii) a goal may be satisfied in more than one way if it or any of its refinements are OR-refined, in which case a choice needs to be made among alternative subgoals in deciding how to fulfill the root-level goal; and, iii) a goal's satisfaction may be affected by that of goals other than its subgoals. Goal analysis produces a goal model consisting of an AND/OR refinement tree with additional positive/negative *contributions*. The satisfaction level of a goal can be inferred from that of others in the same goal model using a label propagation algorithm [5,1]. Examples of goals are shown in Figure 1. Notice how the "Shareholder value increased" goal is AND-decomposed into the sub-goals "Cost decreased" and "Revenue increased"; similarly, the "More Products sold" goal is OR-decomposed in three different alternative sub-goals (strategies), namely "Best customers attracted and retained", "Focused on career and skills development", and "Sell process improved". An example of influence among goals is represented by the one existing from the "Sell process improved" goal towards the "Cost decreased" goal.

In addition to goals, we model partial states of the world as *situations*. For strategic business models, we need the notion of organizational situation, such as "Christmas season", an opportunity for a sales organization, or "Competitor buys key technology", a potential threat. Analogously to satisfaction levels for goals, we have *occurrence* levels for situations, which define the degree to which a situation occurs in the current state-of-affairs. The situations "Christmas season", "Staff need training", and "High customers complaints" described in Figure 1, are some examples of partial states of the world that can occur within a business context.

To reason about goal satisfaction under the influence of situations, we extend the contribution relation so that it can be used to relate any combination of goals and situations. Hereafter, we refer to it with the term of *influence*. For example, the situation "Christmas season" positively influences the goal "Increase sales", while the situation "Booming economy" positively influences the situation "Growing inflation". Figure 1 shows some examples of such influences, e.g., the "Staff need training" situation, representing an internal weakness for the company, influences negatively the "More products sold" goal.

We characterize influences along two dimensions: i) *direction*: a positive (resp. negative) influence exists from a situation/goal to another if the occurrence/satisfaction of the source increases (resp. decreases) the occurrence/satisfaction of the target; and, ii) degree or *strength*: an influence is full if it is a causal relation (i.e., 100% effect on the target of the goal or situation influenced); otherwise, it is partial.

3 Indicators

An *indicator* is a measure, quantitative or qualitative, of the progress or degree of fulfillment of organization goals. The subject of an indicator is a particular *feature* or *quality* of an element in the business environment, e.g., the *workload* of an employee, or the *compliance* of an internal process with respect to external regulations.

To express *why* an indicator is needed and *what* it is measuring, we rely on two relations, *evaluates* and *measures*, as illustrated in Figure 2. In the example, the indicator "Number of products sold" is needed (why) to evaluate the goal "More products sold" by measuring the task "Sell products".

Each indicator, being a composite or component, has a *current value* which is is evaluated against a set of parameters: *target (value)*, *threshold (value)* and *worst (value)* [10]. The result of such evaluation is a normalized value (ranging in $[-1, 1] \subset \mathbb{R}$), which is often referred to as the *performance level*. Note that a current value can be assigned by either: i) extracting it at run-time from back-end data sources[1], ii) supplied by users to explore "what-if" scenarios, or iii) calculated by a *metric* expression (see Section 4.1).

An indicator can be *positive*, *negative*, or *bidirectional*, meaning that we want to maximize, minimize or balance its target. *Performance regions* are defined for

[1] *Dimensions* and *levels* [10] can be used to filter data from datawarehouses.

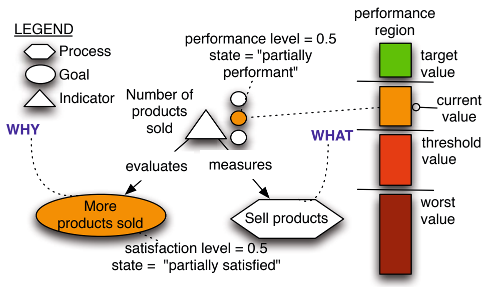


Fig. 2. An example of an indicator to evaluate a goal

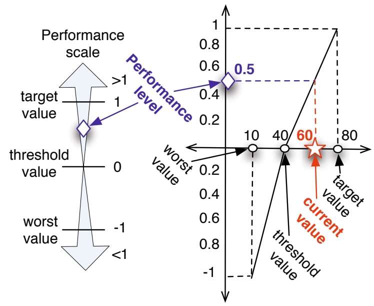


Fig. 3. Example of interpolation [12] to calculate performance levels

each type of indicator by properly combing the indicator's parameters. Figure 2 shows an example of performance region for a positive indicator, i.e., we want to maximize the number of products sold, in which $Target \geq Threshold \geq Worst\ value$.

The relative position of indicator current values within such regions leads to indicator performance levels, as shown by Figure 3. Notice how the worst and target values are mapped respectively to -1 and +1 levels, while the threshold value is mapped to 0. A linear interpolation[2] is used to approximate performance levels, as also described by System equation 1 [12]. For instance, the performance level (pl) for Figure 3 is $pl(60) = \frac{|60-40|}{|80-40|} = 0.5$.

$$pl(current\ v.) = \begin{cases} \frac{|current\ v. - threshold\ v.|}{|target\ v. - threshold\ v.|}, & \text{if } current\ v. \geq threshold\ v. \\ \\ -\frac{|current\ v. - threshold\ v.|}{|threshold\ v. - worst\ v.|}, & \text{if } current\ v. < threshold\ v. \end{cases} \quad (1)$$

Performance levels are, in turn, propagated to the corresponding goals to evaluate satisfaction levels. For example, in Figure 2, the performance level 0.5 is propagated to the satisfaction level of the corresponding goal which, in turn, is mapped to a "partial satisfied" state (orange colour[3]).

As we will show in the following section, indicators can be used to evaluate situations in a similar way we do for goals, by propagating a performance level to the occurrence level of the situation under evaluation. For example, the indicator "Number of products returned" can evaluate the situation "Low number of returned products".

4 Reasoning with Indicators

In a business model, indicators are associated with various business elements in the model. These elements in general are composite, consisting of hierarchies

[2] Other forms of interpolation can be used, e.g., polynomial, spline, etc.

[3] BIM provides mapping tables to map satisfaction, occurrence and performance levels to corresponding states of business elements.

of elements. Such structure implies hierarchies for indicators. For example in Figure 6, the goal hierarchy results in a hierarchy for associated indicators. More specifically, "Number of special package" is a component indicator of "Number of products sold", since it evaluates the goal "Sell process improved" which is a sub-goal of "More products sold".

We are interested in algorithms that propagate values of indicators from a lower level in a hierarchy to ones higher up. We classify such propagation into four categories, as described in Figure 4, based on what is known about the relation between a composite indicator and its components. In the simplest case, such a relation is fully described using a mathematical function, and there is no problem in computing values for the composite indicator. For example, profits can be calculated directly from revenues and costs. In other cases, when such a mathematical relation does not exist, indicator values have to be derived using estimation/approximation techniques.

Knowledge (More ↔ Less)		Knowledge	Outcome	Accurate (More ↔ Less)
More	*Quantitative (accurate)*	component current values, component strength influences, composite (well-defined) mathematical function.	composite current value, composite performance level.	More
	Quantitative (heuristic)	component current values, component strength influences, component conversion factors, composite (approx.) metric.	composite (approx.) current value, composite (approx.) performance level.	
Less	*Quantitative (normalized)*	component current values, component performance levels, normalizing function, composite (approx.) metric.	composite (approx.) performance level	Less
	Qualitative	component current values *and/or* label state values, propagation rules.	composite label state value, composite conflict.	

Fig. 4. Techniques classification

In what follows, we present three techniques to derive values of composite indicators using conversion factors, range normalization, and qualitative reasoning. An end-user may prefer one or the others depending on the quantity of information of the domain she/he posses, or on the available time she/he has for encoding such an information into the model. The qualitative approach may also be chosen when the user is interested in an early analysis of conflicts and inconsistencies within the same goals of a strategy.

4.1 Deriving Composite Indicators Using Conversion Factors

When a composite indicator does not share that same unit of measure with its components, a necessary condition for finding a metric that computes its values is that there is a suitable conversion factor for each component indicator that has a different unit of measure.

For example, consider the two indicators "Employee cost" and "Working time". In particular, Employee cost can be defined as a composite indicator whose value relies on the component indicator Working time. According to our requirement, we need to convert the current value of Working time values measured in hours into Employee cost units. One possible conversion factor is to take the average of the wage per hour for all employees[4]. Assuming that such an

[4] This value can be also defined as an "Average hourly wage" indicator and, in turn, as a related indicator for the Employee cost.

average is 20 and that the current value for Working time is 160 hours, we can calculate an approximated current value for Employee cost as:

1. 20 dollars = 1 hours $\rightarrow 20\frac{\text{dollars}}{\text{hours}} = 1$, where 20 is the conversion factor;
2. 160 hours $\cdot\, 20\frac{\text{dollars}}{\text{hours}} = 3,200$ dollars.

Notice that in many cases a conversion factor is an estimate based on previous experience / statistics. For example, the average wage per hour could be 30 instead of 20 for a different company. When conversions are impossible, e.g., it is not possible to convert gallons to square feet, we have to fall back to a "normalized" approach or to a "qualitative" one; these are presented, respectively, in Sections 4.2 and 4.3.

When conversion factors are available, we become able to define valid metric expressions that contain: i) current values for component indicators, ii) influence strengths, and iii) conversion factors. With this aim, we adopt and use off-the-shelf the grammar of the Jep Java Library (see Section 5), which allows us to express rich and flexible expressions to meet user requirements.

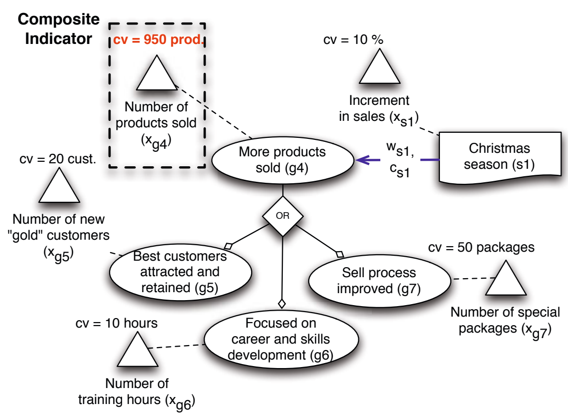


Fig. 5. Examples of reasoning with conversion factors: cv = current value, w = weight, c = conversion factor

An example of such an expression is $x_{g4} = x^e_{g4} + w_{s_1} \cdot c_{s_1} \cdot x_{s_1} + \sum_{j=5}^{j=7} w_{g_j} \cdot c_{g_j} \cdot x_{g_j}$, which is used in Figure 5 to calculate the current value of the "Number of products sold" indicator. In the expression, x^e_{g4} is the expected value of products sold while the different w_m and c_n are, respectively, influence strengths and conversion factors of the component indicators. When the designer defines an expression, she/he must consider two kinds of factors (where by "factors" we mean some quantity which can influence positively or negatively the final value of a composite indicator). First, she/he must take into account those factors derived from "influencers". In the previous example, we have the situation "Christmas season", which influences and contributes positively to the composite indicator with the quantity $w_{s_1} \cdot c_{s_1} \cdot x_{s_1}$. In particular, x_{s_1} is the current value (10 %) of the indicator "Increment in sales", w_{s_1} and c_{s_1} are, respectively, the strength of the influencer, and the conversion factor used to convert the 10 % increment value into a number of products sold. These last two parameters must be chosen accurately by the designer who must rely on her/his domain experience and/or estimates of historical data.

The second type of factors considered are those derived from sub-goals. For example, in Figure 5, the component indicators associated with the sub-goals "Best customers attracted and retained", "Focused on career and skills development" and "Sell process improved" all contribute to the composite indicator expression with a total quantity of $\sum_{j=5}^{j=7} w_{g_j} \cdot c_{g_j} \cdot x_{g_j}$. In this case, a sum operator was chosen to aggregate such quantity (i.e., products sold) derived from the achievement of one or more of these sub-goals. In fact, the achievement of the "Best customers attracted and retained" sub-goal increments the number of new "gold" customers, which in turn represents an increment of products sold. A gold customer buys an average of 5 products during intense sales seasons, and such information can be used by the designer to tune the conversion factor used in the above expression.

The two sets of factors, one from influencers, the other from sub-goals, are then aggregated together to compute a final value for the composite indicator. Again, a sum operator was used, but the designer can customize each expression depending on the semantic of influencers and sub-goals, and how their values may impact the final value of the composite indicator.

4.2 Deriving Composite Indicators Using Range Normalization

When conversion factors are not available, a way to derive composite indicators is to rely on range normalization, which takes values spanning a specific range and represents them in another range. Range normalization is applicable when we do not need to obtain the exact value for a composite indicator, but rather only its performance level. Indeed, when we calculate the performance level of an indicator (by using its current value and parameters as described in Section 3), we are producing a "normalized value" in a range within $[-1, +1] \subset \mathbb{R}$.

A performance level for a composite indicator is calculated by combining performance levels of component indicators. We propose a BNF grammar that allows to define rules for combining levels of component indicators into the one of a composite one. The grammar, presented in Table 1, is highly expressive and can be extended to accommodate new combination rules. Notice how the grammar builds a metric expression by combing (row $3 : combInfDec$) performance levels propagated from influencers (row $4 : plInf$) and from sub-nodes (row $5 : plDec$). The grammar also embeds the concept of "tolerance" [1], which allows to limit the *total* (negative or positive) influence of influencers to the node at hand.

This approach is applied in Figure 6 in which no conversion factors are available. By relying on the grammar, a metric expression was defined for the composite indicator "Number of products sold": $pl(x_{g4}) = sum^t\{0.75 \cdot pl(x_{s1}), max[pl(x_{g5}), pl(x_{g6}), pl(x_{g7})]\}$. The metric takes the maximum performance level among the sub-nodes, and sums such a level to the result obtained by multiplying the influence strength 0.75 by the performance level of the component indicator "Increment

Table 1. A BNF grammar for performance level metric expressions. We use the EBFN ISO notation for symbols: definition(=), concatenation (**,**), termination(**;**), alternative(|), optional([**...**]), repetition(**{...}**), grouping(**(...)**), terminal string (’**...**’).

(E)BNF Grammar for Performance Level Metric Expressions

```
         Start = plExp;

         plExp = 'pl(' , Identifier , ') = ' , (combInfDec | plInf | plDec) , ')';

    combInfDec = Function , '(' , plInf , ',' , plDec , ')' , {Operator , (Identifier | FN)};
         plInf = [Function] , InfluencersList , ['{'Tolerance'}'];
         plDec = [Function] , Sub-nodesList , ['{'Tolerance'}'];

  Sub-nodesList = '[' , Sub-node , {',' Sub-node} , ']';
InfluencersList = '[' , Influencer , {',' Influencer} , ']';
      Tolerance = '1' | '0' | ('0.' , DL);
       Sub-node = 'pl(' , Identifier , ')';
     Influencer = Weight ,'·' , 'pl(' , Identifier , ')';
         Weight = Identifier;
       Function = ('sum' | 'sum^t' | 'min' | 'max' | 'avg');
       Operator = ('+'| '-'| '*'| '/');
     Identifier = AC , {AC | D};
             AC = 'A' | 'a' | 'B' | 'b' | ... | 'Z' | 'z';
             FN = DL | ( DL , '.' , DL);
             DL = D | (D , DL);
              D = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
```

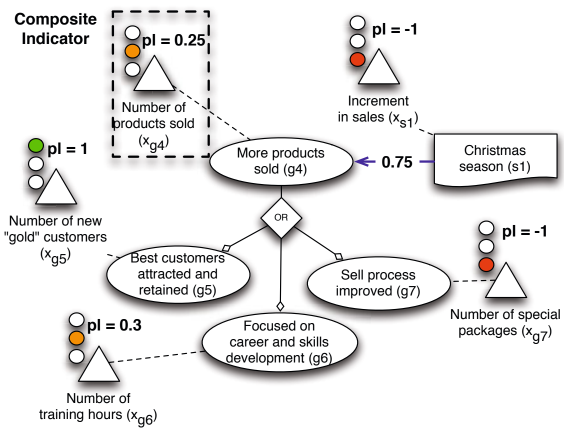


Fig. 6. Examples of reasoning with performance levels of indicators

in sales”. The special operator sum^t allows one to normalize the final result in the range $[-1, +1] \subset \mathbb{R}$, to be further used in other computations. In the example, the result is: $sum^t\{0.75 \cdot -1, max[1, 0.3, -1)]\} = 0.25$ (if such a value was greater than 1 or lower than -1, it would have been normalized to 1 or -1, respectively).

We have defined a BNF grammar for determining the exact syntax for our expression’s language and find all available options for any expression. Notice that, since the BNF here presented could encompass the expression of computations described in Section 4.1, we are considering to extend the Jep Java Library to satisfy such a goal.

A BNF grammar consists of a set of "non-terminals" and "terminals". Non-terminals are placeholders within a BNF definition, defined elsewhere in the BNF grammar. For example, the non-terminal "Function" that appears in the third row of Table 1 is defined some rows later by the set of terminals: { **'sum'** | **'sum^t'** | **'min'** | **'max'** | **'avg'** }. Terminals are endpoints in a BNF definition consisting, in our case, of keywords representing functions (as above) and operators (**+**, **-**, *****, **/**), lower (**a**–**z**) and upper case (**A**–**Z**) alphabet characters, and numbers (**0**–**9**) (in the grammar, all terminals appear in bold type).

Designers can define an expression starting from the non-terminal *Start* and recursively substituting each non-terminal (present on the right side of Table 1), with the appropriate definition, until all the non-terminals are (re)defined by terminals. For example, the non-terminal *Start* is defined by the non-terminal *plExp* which is (re)defined by the expression **'pl('** Identifier **')** = **'** (combInfDec | plInf | plDec) **')'**. The symbol "|" allows for alternatives, therefore the designer must take here her/his first design decision.

Suppose that she/he wants to define the performance level of an indicator *A1*, which is decomposed in two sub-indicators *A2* and *A3* (i.e., we have a graph with a goal G1 with two sub-goals G2 and G3).

She/he must choose the alternative *plDec*, which states for "decomposition", and:

1. (re)define the non-terminal *Identifier* in the non-terminals *AC(D)* and, in turn, in the terminals **A** and **1** to represent the indicator *A1*;
2. (re)define the non-terminal *plDec* in the non-terminals *Function Sub-nodesList* which, with further steps, result into a terminal expression **min[pl(A2), pl(A3)]**.
3. combine all together to obtain the final expression **pl(A1) = min[pl(A2), pl(A3)]** which states that the performance level of *A1* is the minimum among the levels of *A2* and *A3*. In goal reasoning [5], this semantic is often associated with an AND-decomposition where the satisfaction of the father goal is equal to the minimum satisfaction value of its sons.

4.3 Deriving Composite Indicators Using Qualitative Reasoning

Inspired by [5], we augment the techniques of sections 4.1 and 4.2, with a qualitative reasoning technique. In this case, instead of propagating indicator performance levels, we propagate the categorical label assigned to them. This technique has long been used for qualitative goal reasoning in RE.

A major difference between this technique and the ones presented in previous sections, is that conflicts are allowed, i.e., an indicator can be at the same time "fully performant" and "fully non-performant" (see Figure 7). This is analogously to [5], where goals have *satisfiability* values but also *deniability* ones: during label propagation, a goal can be both "partially satisfied" and "partially denied".

We associate to each indicator I_i two variables: positive performance (per^+) and negative performance (per^-). Ranging in {F,P,N} ("full", "partial", "none"),

Indicators colour (per^+, per^-)		Evidence	Mapping rule	Variables (per^+, per^-)
+	-	fully performant (green)	$cv \geq t$	per^+ = "full" per^- = "none"
+	-	partially performant (green-orange)	$M \leq cv < t$	per^+ = "partial" per^- = "none"
+	-	partially non-performant (red-orange)	$th \leq cv < M$	per^+ = "none" per^- = "partial"
+	-	fully non-performant (red)	$w \leq cv < th$	per^+ = "none" per^- = "full"

Fig. 7. Mapping rules, where: cv = current value, th = threshold value, w = worst value, and M = middle value among the target t and threshold th

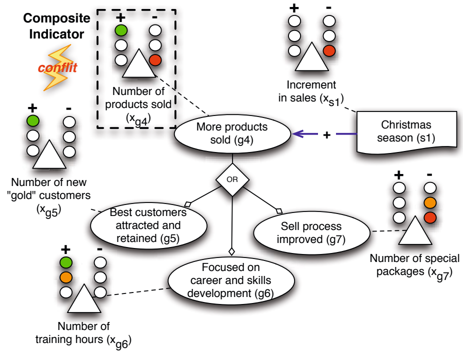


Fig. 8. Examples of qualitative reasoning

Table 2. Propagation rules in the qualitative framework. The (or), (+D), (-D), (++D), (- -D) cases are dual w.r.t. (and), (+S), (-S), (++S), (- -S) respectively. See [5], for details.

	$(I_2^r, I_3^r) \overset{and}{\longmapsto} I_1^c$	$I_2^r \overset{+S}{\longmapsto} I_1^c$	$I_2^r \overset{-S}{\longmapsto} I_1^c$	$I_2^r \overset{++S}{\longmapsto} I_1^c$	$I_2^r \overset{--S}{\longmapsto} I_1^c$
$per^+(I_1^c)$	$min \begin{cases} per^+(I_3^r) \\ per^+(I_2^r) \end{cases}$	$min \begin{cases} per^+(I_2^r) \\ P \end{cases}$	N	$per^+(I_2^r)$	N
$per^-(I_1^c)$	$max \begin{cases} per^-(I_3^r) \\ per^-(I_2^r) \end{cases}$	N	$min \begin{cases} per^+(I_2^r) \\ P \end{cases}$	N	$per^+(I_2^r)$

such that $F > P > N$, these variables represent the current evidence of performance or non-performance of an indicator I_i. For instance, $per^+(I_i) \geq P$ states that there is at least partial evidence that I_i is performant. To assign "evidence" and, therefore, values to the variables per^+ and per^-, we use the mapping rules described in Figure 7. For example, when the current value of an indicator I_i lies among its target value and the middle point M (a value which is equal distant from the target t and from the threshold th), we have that $per^+ =$ "$partial''$ and $per^- =$ "$none''$.

Propagation of the values from component indicators to a composite indicator relies on the axioms and (adapted) propagation rules from [5], which are summarized in Table 2. For example, in Table 2, the rule $(I_2^r, I_3^r) \overset{and}{\longmapsto} I_1^c$ states how labels are propagated when there is an AND-decomposition relation between goal G_1 and sub-goals G_2 and G_3 (here we refer to goal nodes, but they can also be situation nodes, or a mix of both), with associated indicators I_1^c, I_2^r and I_3^r. Analogously, $I_2^r \overset{-S}{\longmapsto} I_1^c$ states how labels are propagated when there exists an influence relation between goals G_2 and G_1, with associated indicators I_2^r and I_1^c. The strength of this influence is equal to $-S$, which means that if G_2 is satisfied, then there is some evidence that G_1 is denied, but if G_2 is denied, then nothing is said about the satisfaction of G_1, see [5] for further details.

Figure 8 provides an example of our qualitative approach. For each indicator, the per^+ variable is represented by a traffic light with a plus symbol on the top, a minus symbol is used for the per^- variable. The colours for the traffic lights are those described in Figure 7. A conflict is discovered for the composite indicator "Number of products sold" when values are propagated following the rules in Table 2. When such conflicts appear in a schema, although undesirable, they do help to highlight particular aspects of a business that need user attention because of possible inconsistencies. In the following, we explain in detail how such conflict is calculated.

For each indicator, the corresponding current value is extrapolated from the data sources. For example, for the indicator "Number of training hours" we have a current value of 16. The associated target and the threshold values are, respectively, 20 and 10; therefore, by following the second mapping rule of Figure 7 ($M \leq cv < t$), we obtain that the indicator "Number of training hours" is "partially performant (green-orange)" (as also reported in Figure 8).

By applying the same procedure for all the indicators (with the exception of the "Number of products sold"), we obtain: a fully performant "Number of new "gold" customers (green)"; a partially non-performant "Number of special packages (red-orange)"; and, a non-performant "Increment in sales". The next step is to rely on the propagation rules described in Table 2 to propagate and calculate the colour of the "Number of products sold" indicator.

We start to propagate the three indicators associated to the corresponding sub-goals by relying on the rules in the first column of Table 2. Indeed, the OR rules are dual with respect to the AND rules [5]. Therefore, for the per^+ and per^- variables, we need to choose respectively the *maximum* and the *minimum* among the values of the sub-indicators.

The result is:

$$\begin{cases} per^+(\text{from-sub-nodes}) = max[per^+(x_{g5}), per^+(x_{g6}), per^+(x_{g7})] = \text{“Full” – i.e., fully performant} \\ per^-(\text{from-sub-nodes}) = min[per^-(x_{g5}), per^-(x_{g6}), per^-(x_{g7})] = \text{“None”} \end{cases}$$

We use "from-sub-nodes" as a temporary node to store the result; in fact, we need to combine such values with the ones from the influencer "Christmas season". To propagate the associated indicator "Increment in sales" we must use the rules in second column of Table 2. Notice that, the influence from the "Christmas season" situation to the "More products sold" has a plus (+) symbol. As described in [5], this is a symmetric relation and it is a shorthand for the combination of the two corresponding asymmetric relationships $I_2^r \stackrel{+S}{\longmapsto} I_1^c$ and $I_2^r \stackrel{+D}{\longmapsto} I_1^c$ (the propagation rule for the latter is dual w.r.t. the former); this means that both satisfiability and deniability are propagated. Therefore, after propagation, we obtain:

$$per^+(\text{from-influencer}) = min \begin{cases} per^+(x_{s1}) = \text{“None”} \\ P \end{cases} = \text{“None”}$$

$$per^-(\text{from-influencer}) = max \begin{cases} per^-(x_{s1}) = \text{“Full”} \\ P \end{cases} = \text{“Full” – i.e., fully non-performant}$$

Again, we use a temporary node, namely "from-influencer", to temporary store the result which must be combined with the result in the node "from-sub-nodes".

Giorgini et al. [5] provide an algorithm to combine such results, from sub-nodes (our node "from-sub-nodes") and from influencers (our node "from-influencer"). In our case, the result is the one depicted in Figure 8 where a conflict is discovered. The main idea behind such an algorithm, is to take, for the per^+ variable of the composite indicator (i.e., the "Number of products sold"), the maximum value among all the temporary per^+ values (both sub-nodes and influencers). The same must be done for the per^- variable.

5 BIM's Tool Support

We have implemented a visual editor prototype to draw business schemas and reason about them. Our implementation uses *Graphiti*[5], an Eclipse-based graphics framework that enables easy development of state-of-the-art diagram editors for domain-specific modeling languages. The current version of the prototype implements the quantitative approach described in Section 4.1 by relying on Jep[6], a Java library for parsing and evaluating mathematical expressions. Jep supports strings, vectors, complex numbers and boolean expressions.

Figure 9 provides a snapshot of the tool. Marker "A" highlights the business schema and the toolbar containing business element constructs. Marker "B" highlights the property panel containing indicator parameters and current value. Marker "C" highlights the property panel containing the definition of the metric expression (notice available variables such as strengths, conversion factors, etc.).

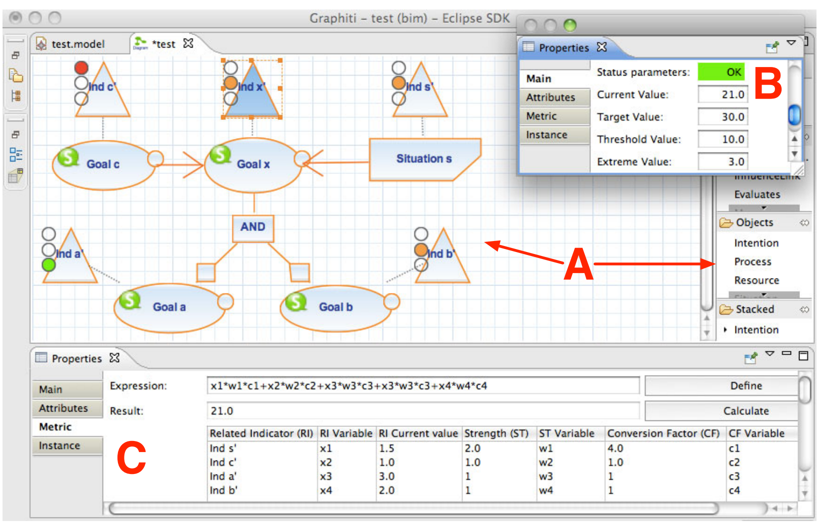


Fig. 9. Graphiti visual editor

[5] http://www.eclipse.org/graphiti/
[6] http://www.singularsys.com/jep/

6 Related Work

The use of business-level concepts—such as goals, processes and resources—has been researched widely for at least 15 years and is already practiced to some extent in both Data Engineering and Software Engineering. In the literature, different modeling proposals exist that are related to our work, such as i* [16], URN/GRL [6] and KAOS [4,15], all from the general area of Goal-Oriented Requirements Engineering. From these we have adopted intentional and social concepts. However, these models lack primitive constructs for influence relationships, (composite) indicators, and various types of situations integrated in the BIM modeling framework. Recent proposals have extended URN to include indicators [13]. We share ideas with this work; however: i) our indicators are more general and they can be used to measure any model object, including other indicators; ii) we provide more guidelines to distinguish "what" is measured and "why" it is measured; and iii) our indicators can be used to evaluate situations which, from our perspective, are fundamental for strategic reasoning.

In [11], the authors propose a formal framework for modelling goals (and for evaluating their satisfaction) based on performance indicators. Our work shares similar intentions but focuses more on the concept of composite indicator and a way to define metric expressions to calculate their values.

From a business perspective, our business schemas can capture what is commonly found in Strategic Maps [9] and Balanced Scorecards [8], but we also support reasoning and we include the concept of situation, a fundamental concept for supporting SWOT analysis. In fact, as we show in [7], we can map our approach to SWOT analysis and other languages that enable goal analysis techniques [5,6], including probabilistic ones.

7 Conclusions

In this paper, we presented a model-based approach to design and reason about an organization's business environment and strategies, with a focus on Key Performance Indicators and indicator composition in the context of the Business Intelligence Model language. We provided qualitative and quantitative techniques to analyze the impact of strategies on organization goals, by relying on different types of knowledge measured through indicators. An Eclipse-based prototype was used to support our findings and validate the feasibility of our approach. We argue that the indicators and composition mechanisms proposed here are more flexible and general than what is commonly found in related work.

As future work we are carrying out a real-world case study to empirically evaluate our composite indicator approach. We are currently involved in a Business Intelligence project at a Toronto-area hospital, where we are building a global picture of patient flow in order to identify sources of bottlenecks within and beyond the hospital (e.g., long-term care facilities, home care services, etc.).

References

1. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.S.K.: Evaluating goal models within the goal-oriented requirement language. Int. J. Intell. Syst. 25(8), 841–877 (2010)
2. Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Composite indicators for business intelligence. In: 30th Int. Conf. on Conceptual Modeling ER 2011 (short paper, to appear, 2011)
3. Barone, D., Mylopoulos, J., Jiang, L., Amyot, D.: Business Intelligence Model, ver. 1.0. Technical Report CSRG-607, University of Toronto (March 2010), ftp://ftp.cs.toronto.edu/csri-technical-reports/INDEX.html
4. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program. 20(1-2), 3–50 (1993)
5. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Reasoning with goal models. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 167–181. Springer, Heidelberg (2002)
6. Intern. Telecom. Union: Recommendation Z.151 (11/08). User Requirements Notation (URN) – Language definition, http://www.itu.int/rec/T-REC-Z.151/en
7. Jiang, L., Barone, D., Amyot, D., Mylopoulos, J.: Strategic models for business intelligence: Reasoning about opportunities and threats. In: 30th Int. Conf. on Conceptual Modeling ER 2011 (short paper, to appear, 2011)
8. Kaplan, R.S., Norton, D.P.: Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
9. Kaplan, R.S., Norton, D.P.: Strategy maps: Converting intangible assets into tangible outcomes. Harvard Business School Press, Boston (2004)
10. Parmenter, D.: Key Performance Indicators. John Wiley & Sons, Chichester (2007)
11. Popova, V., Sharpanskykh, A.: Formal modelling of organisational goals based on performance indicators. Data & Knowledge Engineering 70(4), 335–364 (2011)
12. 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)
13. Pourshahid, A., Richards, G., Amyot, D.: Toward a goal-oriented, business intelligence decision-making framework. In: Babin, G., Stanoevska-Slabeva, K., Kropf, P. (eds.) MCETECH 2011. LNBIP, vol. 78, pp. 100–115. Springer, Heidelberg (2011)
14. Souza, V., Lapouchnian, A., Mylopoulos, J.: System identification for adaptive software systems: a requirements-engineering perspective. In: 30th Int. Conf. on Conceptual Modeling ER 2011 (to appear, 2011)
15. van Lamsweerde, A.: Reasoning about alternative requirements options. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 380–397. Springer, Heidelberg (2009)
16. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: RE 1997, Washington, USA, pp. 226–235 (1997)

VAMEE: A Value Aware Method for Evaluating Inclusive E-Government Initiatives

Gustaf Juell-Skielse and Erik Perjons

Department of Computer and Systems Sciences, Stockholm University
{gjs,perjons}@dsv.su.se

Abstract. The growing use of ICT solutions for improving the public sector has created a need for valuating e-government initiatives. A number of methods for this purpose have been developed, but they are typically restricted to analyzing the benefits and costs of only one single actor. There is, therefore, a need for methods that take a broader view and take into account entire networks of actors. This paper proposes a novel method, called VAMEE, the purpose of which is to produce a well-grounded and easily understandable valuation of an e-government initiative that takes into consideration the benefits, costs, and interrelationships of all actors concerned. The basis of the proposed method is a combination of enterprise modeling techniques, in particular goal modeling and value modeling, with an established method for cost benefit analysis (i.e. Peng). VAMEE is designed to be inclusive, easily understandable, and visual. These properties of the method will support accurate and unbiased valuations as well as improved innovation in the development of e-government initiatives.

Keywords: e-government, benefit analysis, enterprise modeling, goal modeling, value modeling, requirements engineering.

1 Introduction

The growing interest in applying ICT to achieve better government [1] emphasizes the need for benefits valuation methods for e-government initiatives. E-government initiatives include investments in e-services to improve citizen service [2] as well as integrated processes to enhance government back office operations [3]. Through privatization and public private partnerships [4], e-government initiatives involve and affect entire networks of actors that depend on each other in terms of value creation, including actors such as citizens, employees at different levels of government, third party service providers, and companies [5]. Often, the benefits and costs of e-government initiatives are unevenly distributed among the actors, and one actor could carry a large portion of the costs while most of the benefits are reaped by other actors [6]. For example, local governments may carry costs for e-service platforms that primarily lead to administrative savings at third party service providers.

There are a number of models and methods available to measure and control e-government initiatives. The Value measuring methodology [7] by the CIO Council and the Guide for benefit realization [8] by the e-Government Delegation are two

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 97–111, 2011.

examples of methods developed for measuring e-government initiatives. These methods include definitions of key concepts and steps to follow in order to measure costs and benefits of investments in e-government. However, they do not take into account the multitude of actors involved and their interdependence in terms of value creation. Hence, the methods take an internal rather than an external perspective on e-government and focus on the benefits and costs of only one single actor. This restricted perspective limits the understanding of how investments in e-government enables value creation among networks of actors, which in turn may lead to erroneous investment decisions and slow e-government development and uptake.

The complexity of e-government initiatives has created a need for a practical approach that takes into account all actors involved in value creation, visualizes their relationships, and estimates the total value of such initiatives. In two previous papers [6, 9], we have presented and demonstrated such an approach for benefits valuation of e-government initiatives. The approach combines two established approaches, Peng [10] and value modeling [11], to visualize both the actors involved and the value creation that takes place within and between these actors.

In this paper, the purpose is to convert and extend the combined approach into a thorough and easily applicable method by clarifying and detailing its design objectives as well as its phases and activities. In addition, we expand the scope to include key aspects of change management, such as actor involvement and goal modeling. The method is named *A Value Aware Method for Evaluating Inclusive E-Government Initiatives* (VAMEE). The goal of VAMEE is to produce a well-grounded and easily understandable valuation of an e-government initiative that takes into consideration the benefits, costs, and interrelationships of all actors concerned. The basis of VAMEE is a combination of enterprise modeling techniques, in particular goal modeling and value modeling, with an established method for cost benefit analysis, i.e. the Peng method [10].

The paper is structured as follows. Section 2 gives the theoretical bases of the proposed method and describes the fundamentals of goal modeling, value modeling, and the Peng method for cost benefit analysis. Section 3 discusses the design science approach used for developing VAMEE, and introduces a running example based on a case study. Section 4 describes VAMEE in detail, and Section 5 provides a brief demonstration and evaluation of the method. Section 6 compares VAMEE to other valuation approaches in the literature, and Section 7 concludes the paper and suggests directions for future work.

2 Theoretical Bases for VAMEE

The VAMEE method is based on three types of modeling approaches: cost benefit analysis using Peng, value modeling, and goal modeling.

2.1 Cost Benefit Analysis Using Peng

Peng has become a popular method for cost and benefits evaluation of IT-investments and process changes, in both the private and the public sector [10, 12]. It is presented as a "structural method to evaluate, in dollars, all the different types of benefits that IT generates within an operation" [10, p.26]. It is similar to the Value Measurement Model [7] and includes both soft and hard measures [2]. The method consists of ten steps, and ideally it involves users and managers as well as functional and technical specialists from an organization. In Peng, cost and benefits are identified in workshops and organized in tree structures that depict the relationships between benefits, costs, process changes and IT functionality. All cost and benefits are expressed in monetary terms, although the intention is not to achieve accounting precision. In order to validate the results, the benefits are classified as direct, indirect and intangible benefits. Identification of IT costs is supported by a predefined list of types of costs. In the final step, the net value of the benefits and costs are calculated, and managers responsible for the realization of the benefits are appointed.

In addition, the many books on Peng provide a large number of cases from a variety of industries where the method has been applied. Through case based reasoning [13], participants can solve their own case by using or adapting existing cases. In a study of the Peng method [14], its strengths were found to be the business process oriented approach, the mixture of personnel categories participating in the evaluation, as well as the inclusion of assessing soft benefits. Weaknesses were found to be the difficulties of attracting and involving the right participants, the subjective nature of assessment, and difficulties in verifying the net values.

The main terms from the Peng method, benefit and cost, are in this paper defined as:

A *benefit* is an increase in a resource, expressed in monetary terms, that is caused by the implementation of an e-government initiative.

A *cost* is a decrease in a resource, expressed in monetary terms, that is caused by the implementation of an e-government initiative.

2.2 Value Modeling

A value model is a representation of a network of cooperating actors that together create value through resource transfers. A value model aims to provide a high level view of a value network using a limited set of modeling elements, such as actor, resource, and resource transfer. Value modeling can be a basis for profitability analysis, that is, reasoning about the economic viability of the actors participating in a value network; for process and service design, that is, as a foundation for designing business processes and services for involved actors in the network; and for relationship analysis, that is, establishing roles and responsibilities among different actors in the network.

There exists a number of value modeling techniques and ontologies, such as REA [15], BMO [16] and e3value [11]. The value modeling technique used in this paper is in line with the e3value technique. However, both the syntax and semantics differ in some respects in order to arrive at a technique that is as simple as possible given its purpose.

The main modeling elements in the value modeling technique used in this paper are: actor, resource, and resource transfer, which are defined as follows:

An *actor* is a human being, role, organization or organizational unit that can participate in the transfer of resources. Actor can refer to an individual actor (e.g. “Child Care Administration”), or a role (i.e. actor category, such as “Parent” and “Day Care Unit”) that can be taken by multiple individual actors. In the notation, the actor is represented as a rectangle with a stick figure and the actor’s name, see Figure 1.

An *economic resource* is an object that is viewed as being valuable by some actor and such that an actor can have legal control over it and transfers it to other actors. A resource can be categorized into one of the following categories: a good (e.g. “Food”); a service (e.g. “Day Care Service”); information (e.g. “Child Care Usage Info”); and money (e.g. “Child Care Fee”). In the notation, an economic resource is represented as a label on the transfer symbol (i.e. an arrow), and the category of the economic resource is represented as a label between square brackets after the economic resource label, see Figure 1.

A *resource transfer* is an action in which the right on an economic resource is handed over from one actor to another. For example, a day care service (an economic resource of the category “service”) is transferred from a day nurse unit to a parent, which means that the parent has the right to use the service. In the notation, the transfer is represented as an arrow between two actors, see Figure 1.

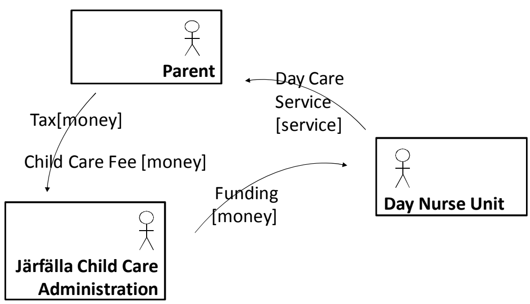


Fig. 1. A Value model of a child care network in a Swedish municipality. The Value model is on operational level.

Value models could be constructed on different levels. In this paper we use two different levels: operational and policy/planning, see Figure 2.

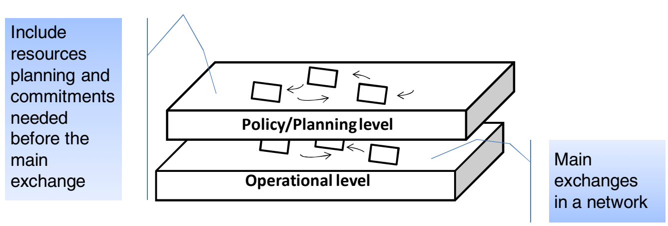


Fig. 2. Value models on different levels

A *value model on the operational level* represents the transfers of economic resources on the operational level, such as in Figure 1. Resource transfers on the operational level are the main transfers of economic resources between actors in a value network.

A *value model on the policy/planning level* includes resource transfers that support planning (e.g. scheduling) and commitments (e.g. orders), see Figure 2. These policy/planning level economic resource transfers are needed in some business settings before the transfers of economic resources on the operational level can be carried out. Another way to put it is that the results of economic resource transfers on the policy/planning level are used to regulate the economic resource transfers on the operational level.

2.3 Goal Modeling

A goal model is a representation of enterprise goals, usually structured as a hierarchy in which high levels goals are decomposed into sub-goals (which are supported by means). Goal modeling can be used to identify and structure desirable states of an actor in order to drive the actor towards these states.

There exists a number of goal modeling techniques and ontologies, such as i* [17] and the Business Motivation Model (BMM) [18]. The goal modeling technique used in this paper is in line with the BMM technique. However, both the syntax and semantics differ in some respects in order to arrive at a technique that is as simple as possible given its purpose.

The main modeling elements in the goal modeling technique presented in this paper are goals and relations between these.

A *goal* is a statement about a desirable state of an actor. A goal can be decomposed into several sub-goals.

2.4 Motivating the Theoretical Bases

The VAMEE method is based on three types of modeling approaches: cost benefit analysis using Peng, value modeling, and goal modeling. Peng is a participative, bottom-up approach for identifying and valuating of IT-enabled benefits and costs [10]. However, Peng focus on the benefits and costs of one single actor and does not take into account the multitude of actors involved in e-government initiative. Value modeling is a well established top-down approach for visualizing the complete network of actors involved in the value creation process. A value model provides a high level view of the resources that are exchanged among the actors of a network. Therefore, value modeling can extend the one actor's perspective of Peng by taking a network perspective. Goal modeling helps in identifying and formulating desirable future states of an actor, and supports the Peng method's identification of benefit and cost of e-government initiative. All three approaches aid in the understanding of how IT, process changes and economic value are interrelated and how an e-government initiative affects actors involved in a value network.

3 Methodological and Empirical Bases for VAMEE

This section describes the research strategy used, the objectives of VAMEE, and the case study in which the method was designed and applied. The case study is also the basis for the running example used in Section 4.

3.1 Design Science

For developing the method, we have used design science [19, 20] as a research strategy, in particular Peffers et al.'s model for design research [20], which consists of six steps:

1. Identify problem and motivate
2. Define objectives of a solution
3. Design and develop
4. Demonstration
5. Evaluation
6. Communication

The problem and its motivation have been discussed in Section 1. In the following sections, the remaining steps are addressed.

3.2 Objectives of the Method

The overall objective is to provide a method for measuring and controlling e-government initiatives, which is applicable to networks of actors and can be easily adopted in practice. This overall objective is broken down into three specific objectives for the method:

- Inclusive in terms of involving all actors of a network
- Comprehensible for actors in the public sector
- Complementary to commonly used project management and system development methods

3.3 Case Study

VAMEE has been designed during an e-government initiative study in the area of child care in Järfälla, a Swedish municipality [6, 9]. The method has been applied and the result has been used for refining the method.

The municipality Järfälla is one of Sweden's 290 municipalities, located 20 kilometers Northwest of Stockholm. With 65 000 inhabitants, it is, by Swedish standards, a relatively large municipality. More than 3 000 children in the age of 1-5 years are engaged in the child care provided by the municipality. Approximately three fourths of the day nurse units are owned by the municipality, and one fourth is privately owned.

The interaction between parents and child care administration is made up of four processes:

- Application and placement offer
- Schedule changes
- Salary changes
- Termination of contract

The interaction between parents and child care administration are based on paper forms that are sent using traditional mail services. A lot of phone contacts are made between parents and child care administrators, especially during the process of application and placement.

Schedule changes, which is the basis of the e-government initiative used as a running example in this paper, are primarily driven by changes in parents work schedules and are handled through forms that are handed in manually at the day nurse units. In 2007, Järfälla introduced e-services in order to replace the manual forms and the physical handling of these forms.

4 Description of VAMEE

This section describes VAMEE, which consists of two phases: *Motivate* and *Investigate,* see Figure 3. *Motivate* consist of three activities and *Investigate* of two activities.

Phases	Activities	Output
1. Motivate	Involve participants ↔ Set goals for actors ↔ Initial design of value network	- Formulated goals - Communicated commitment - Visualized value network
2. Investigate	Analyze benefits and cost ↔ Syntezise value network	-Calculated net value - Syntesized and visuallized value network

Fig. 3. The phases and activities of the VAMEE Method

4.1 Overall Design of the Method

VAMEE was constructed using Peng as a starting point. It has inherited much of the characteristics of Peng: user participation, bottom-up determination of costs and benefits, and a methodical approach. VAMEE has also inherited the characteristics of value modeling: a top-down approach for visualizing the entire network of actors involved in the value creation process as well as a value modeling ontology [11]. In addition, VAMEE emphasizes iteration, both within and between phases. For example, goals and value network descriptions are refined in several iterations, moving from ideas to detailed descriptions and commitments. Through iteration, the dynamic nature and learning aspects of technological change is recognized.

4.2 The Motivate Phase

The phase *Motivate* aims at establishing a preliminary understanding of the goal and scope of the e-government initiative under consideration and to mobilize participants for the work in the following phases. The phase includes the following activities: *Set Goals for Actors, Involve Participants,* and *Initial Design of Value Network.*

Set Goals for Actors

The activity *Set Goals for Actors* aims at formulating goals for the e-government initiative, e.g. to improve the quality of citizen service and to become more cost effective. There exists a multitude of independent actors in the public sector and it is important to clarify goals for each actor. Typically, this activity would start with discussions within one actor or in discussions between a small numbers of actors. In iterations between this activity and the activities *Initial Design of Value Network* and *Involve Participants*, actors affected by the e-government initiative are successively identified and subsequently involved in goal discussions. The activity is completed when all actors affected by the e-government initiative have formulated and agreed on individual goals and their relations. Conflicts between the goals need to be managed in this activity, or in the second phase and the *Synthesize* activity. The output of this activity is goal statements for each actor.

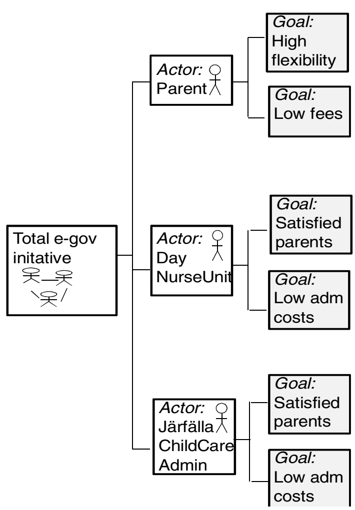


Fig. 4. The goals for the actors involved in the e-government initiative

Running example: In the case at Järfälla municipality, three actors were identified: the Child Care Administration at the municipality, a private Day Nurse Unit, and the parents. The main goals, in the context of the e-government initiative, for the Child Care Administration and the Day Nurse Unit were Low costs for administrative work and Satisfied parents, while the main goals for the parents were High flexibility and Low fees, see Figure 4. These goals were elicited in discussions with participants representing the three actors.

Involve Participants

The activity *Involve Participants* aims at involving participants representing all actors in the value network. Ideally, both managers and functional experts from each actor are involved, thereby ensuring that decision powers, as well as operational understanding, are available. In this activity, the participants approve of the preliminary goals and they develop a shared understanding of how VAMEE will be applied. As the value network is scoped, the number of actors may increase, and more participants need to be involved in the work. The activity is completed when participants from all actors have taken an active part in the initial cost benefit analysis; have committed to preliminary goals; and have arrived at an adequate understanding of how VAMEE will be applied.

Running example: In the Järfälla case, relevant participants for the Child Care Administration included both managers and public servants, while the Day Nurse Unit included managers as well as child-care staff, see Figure 5.

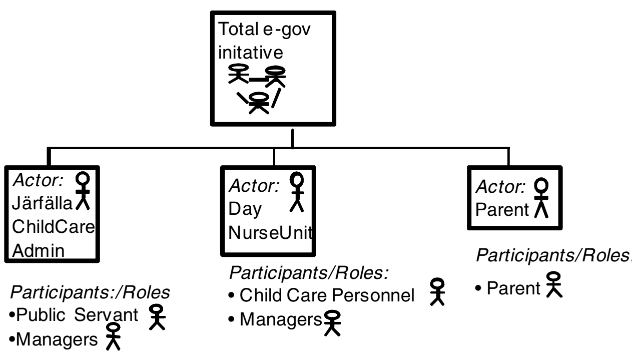


Fig. 5. The actors and participants/roles in the e-governmental initiative

Initial Design of Value Network

The activity *Initial Design of Value Network* aims at specifying actors as well as resource transfers in the value network for the e-government initiative. This can be a highly complicated task when analyzing the public sector. First, there are many different governmental organizations at local, regional and national levels that regularly collaborate to provide governmental services to the citizens. Second, in most governmental areas, there are public organizations that source or purchase the services, using tax revenue from citizens. Third, there are organizations that will provide the services. In many countries, the latter type of organizations can be publicly or privately owned, but funded by public organizations.

The result of this activity will be a visualized draft of a value network for the e-government initiative, which will be refined in the next phase.

Running example: The value network for the Järfälla case when introducing a schedule change service is shown in Figure 6. The diagram identifies the three actors of the case as well as the main resource transfers between them. As can be seen, these resources are various kinds of information in digital form as well as an e-service. For example, the parents will provide the Day Nurse Unit with information on their planned child care usage in digital form, and the Child Care Administration will provide parents with the e-service for changing the schedule.

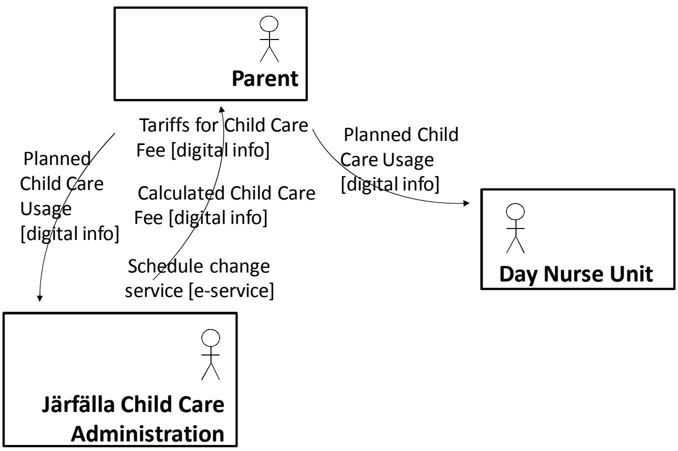


Fig. 6. A Value model on policy level of a child care network

4.3 The Investigate Phase

The phase *Investigate* aims at identifying and quantifying costs and benefits for each of the actors involved in the e-government initiative. *Investigate* is made up of the activities *Analyze Benefits and Costs* and *Synthesize Value Network.*

Analyze Benefits and Costs

The activity *Analyze Benefits and Costs* aims at identifying and estimating costs and benefits for each of the actors in the value network. Benefits and costs are identified in workshops and organized in a tree structure that depicts the relationships between benefits, process changes, and IT functionality. Identification of IT costs is supported by a predefined list of cost types. All benefits and costs are expressed in monetary terms although the intention is not to achieve accounting precision. The net value of the benefits and costs are calculated.

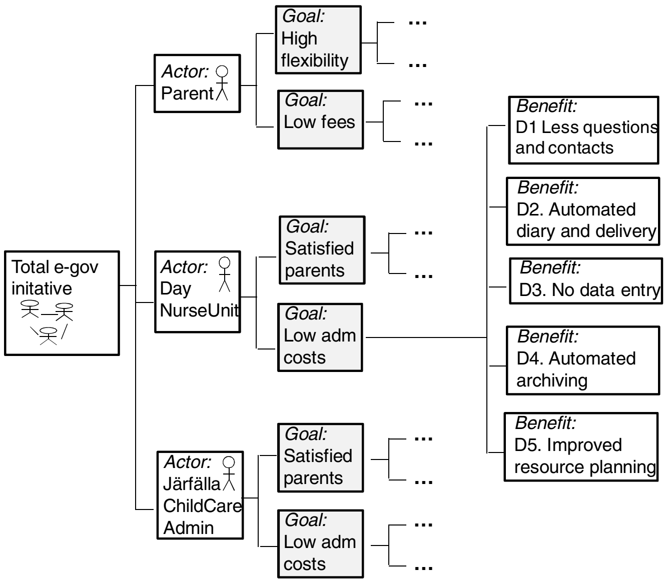


Fig. 7. The benefits related to the goal "Low adm costs" of the Day Nurse Unit

Running example: The benefits describe and quantify in monetary terms what improvements can be achieved if the e-government initiative is implemented, including both e-services and administrative back-office systems. For reasons of space, we only show the benefits of the Day Nurse Unit, see Figure 7. The costs as well as the relations between the costs, benefits, business processes and IT are showned in [9]. The benefits for the goal "Low Adminstrative Costs" for Day Nurse Unit are:

D1. Less questions and contacts (saving 39 000 SEK/year). The form used by the e-service includes information and controls that reduce the need for direct contact with the day nurseries.

D2. Automated diary and delivery (saving 39 000 SEK/year). Manual diary entry is replaced by automated handling by a workflow system.

D3. No data entry (saving 104 000 SEK/year). There is no longer need for manual data entry at the day nurse units since data is already entered by parents using the e-service.

D4. Automated archiving (saving 26 000 SEK/year). Archiving is automated in the workflow system and replaces physical archives.

D5. Improved resource planning (N/A). Schedules provided by parents are used for resource planning at day nurse units. The day nursery staffing plans become more accurate when the quality of the information provided by the parents improves. However, the value of this item was not assessed in the cost benefits analysis.

Synthesize Value Network

The activity *Synthesize Value Network* aims at mapping the results from the activity *Analyze Benefits and Costs* to the value model. In this activity, the benefits and costs from the Peng analysis will be mapped to the transfers of economic resources visualized in the value model, created in the previous phase. The benefits and costs are shown within parentheses after the resource labels. If a benefit or cost cannot be mapped to a resource transfer, another resource transfer needs to be added to the value model. The final step is to analyze the distribution of values visualized in the value model (i.e. actors and their transfers of economic resources) together with the results from the benefit and cost analysis in the previous activity. This will support government officials to prioritize e-government initiatives.

Running example: The initial value model from Figure 6 is in Figure 8 extended by mapping its resource transfers to benefits (for example benefits D1 to D5 presented in the previous step). That is, the model in Figure 8 shows which resource transfers that contribute to the various benefits. (Costs are not shown in Figure 8.)

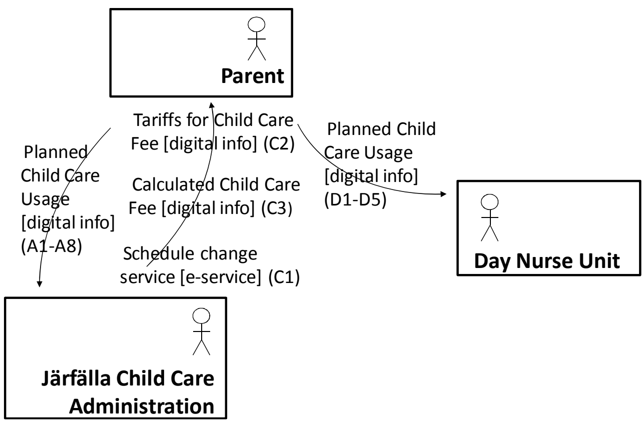


Fig. 8. A Value model on policy level of a child care network in the e-government initiative, including the benefits

5 Demonstration and Evaluation

Most parts of the method have been used in a case at Järfälla municipality, as described in Section 3. The experiences from this use provide an initial demonstration of the method and indicate that the method supports users in identifying actors and benefits that were not included at the beginning of the Järfälla municipality e-government initiative, [9]. The method has not yet been empirically evaluated in a thorough way, but we here offer an evaluation in the form of informed arguments:

- *Inclusive in terms of involving all actors of a network:* The visualization of the value network supports the identification of all relevant actors in an e-government initiative. Furthermore, as all actors are visualized in the network, it becomes easier for them to understand their role and contribute to the development and evaluation of the initiative. There is also a specific activity in VAMEE, *Involve participants*, that supports this objective.

- *Comprehensible for actors in the public sector:* VAMEE is based on well-known concepts such as actor, goal, resource, benefit and cost, which are all widely used in the public sector. Furthermore, the method's results are offered in a visual form.
- *Complementary to commonly used project management and systems development methods:* The method includes a value network perspective which is usually not considered in project management and systems development methods, thereby providing a complementary perspective to such methods. Furthermore, the method can easily be included, as a plug-in, in any project management or systems development method, as the goals in the VAMEE method can work as an interface to the goals specified in other kinds of methods.

6 Related Research

The increased investments in e-government in recent years have led to an increased interest in evaluating the results of e-government investments. Therefore, many researchers and practitioners have presented and discussed methods for evaluation, but also which types of benefits and costs shall be used for measuring investments in the public sector.

In this paper, value modeling, goal modeling and benefit analysis based on Peng are combined in order to produce an evaluation of an e-government initiative.

Value models have been used for profitability analysis [11, 21]. Such an analysis addresses the volumes and monetary values of resource exchanges in a value network. The aim is to analyze the economic viability of the network. However, existing profitability analyses using value models have no support for how to estimate the revenues and costs. The VAMEE method includes such support.

Using process analysis [22, 23] as the basis for evaluating the result of e-government investments is a common approach in research, but these are mainly focusing on one actor and not the entire network of actors, which is the focus of the VAMEE method.

There are many research papers specifying which benefits, costs, and dimensions of benefits and costs that need to be measured in cost benefit analysis [2, 7, 24]. In general, both researchers and practitioners have stated that such benefits, costs, or dimensions need to consist of both monetary values and softer values, such as social and public values. Several multidimensional frameworks for the evaluation of e-governmental initiatives have, therefore, been designed [25]. However, several researchers have claimed that the types of benefits and cost depend on the situation for the e-government initiative and cannot be specified in advance [26]. The VAMEE method supports actors to specify benefits and cost for the e-government initiative based on the situation for the e-government initiative.

7 Conclusions and Future Work

In this paper, we have proposed a new method for the valuation of e-government initiatives. The characteristics of the method include a network perspective, a high degree of visualization, and a firm grounding in enterprise modeling. Utilizing enterprise modeling for the purpose of valuation enables organizations to manage knowledge about themselves and the environment in which they are embedded. The method has been successfully used in a case at a Swedish municipality.

A main advantage of the method is that it widens the perspective of the actors involved in an e-government initiative, as the method requires the benefits, costs and interrelationships of all actors to be made explicit and taken into account. This widened perspective contributes to the correctness of the evaluation, as the interests of all concerned actors are taken into consideration. The broader perspective also supports innovation, as involving more actors opens up new opportunities for value creation and distribution. To involve all concerned actors is particularly important in the public sector, where the needs and interests of all citizens have to be respected.

Future work will focus on further widening the perspective on the benefits of an e-government initiative. The current use of the Peng method requires the benefits to be expressed in monetary terms, i.e. benefits are viewed in one dimension only. However, a multi-dimensional view on benefits can be preferable in many circumstances and should be investigated. Furthermore, the method needs more validation through comprehensive case studies and the involvement of other relevant actors such as software vendors.

References

1. Grönlund, Å.: Ten Years of E-Government: The 'End of History' and New Beginning. In: Wimmer, M.A., Chappelet, J.-L., Janssen, M., Scholl, H.J. (eds.) EGOV 2010. LNCS, vol. 6228, pp. 13–24. Springer, Heidelberg (2010)
2. Gupta, M.P., Jana, D.: E-government evaluation: A framework and case study. Government Information Quarterly 20, 365–387 (2003)
3. European Commission, A Digital Agenda for Europe, http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0245:REV1:EN (viewed May 20, 2011)
4. Savas, E.S.: Privatization and Public-Private Partnerships. CQ Press, Washington (2000)
5. Swedish Association of Local Authorities and Regions, Local self-government, http://www.skl.se/artikel.asp?A=48688&C=6393 (viewed December 7, 2009)
6. Juell-Skielse, G., Perjons, E.: New Challenges for e-Government: Value Realization among Independent Actors. In: 4th International Conference on Research Challenges in Information Science (RCIS), pp. 437–445. IEEE Computer Society, Los Alamitos (2010)
7. CIO Council. The value measuring methodology: Highlights. Washington, DC7 CIO Council, Best Practices Committee (2001)

8. E-Government Delegation. Vägledning i nyttorealisering, http://www.edelegationen.se/sites/default/files/imce/filer/publikationer/Vagledning_i_Nyttorealisering.pdf (viewed June 30, 2011)
9. Juell-Skielse, G., Perjons, E.: Improving E-Government through Benefit Analysis and Value Modeling. In: Ahamed, S.I., Bertino, E., Chang, C.K., Getov, V., Liu, L., Ming, H., Subramanyan, R. (eds.) Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC 2009, Seattle, Washington, USA, July 20-24, pp. 332–339. IEEE Computer Society, Los Alamitos (2009)
10. Dahlgren, L.E., Lundgren, G., Stigberg, L.: Make IT profitable: PENG - A practical tool for financial evaluation of IT benefits, Ekerlids Förlag (1997)
11. Gordijn, J., Akkermans, H.: Value based requirements engineering: Exploring innovative e-commerce idea. Req. Engineering Journal 8(2), 114–134 (2003)
12. Dahlgren, L.E., Lundgren, G., Stigberg, L.: Öka nyttan av IT i vården!: att skapa och värdera nytta i verksamheten med hjälp av PENG-modellen, Ekerlids (2003) (In Swedish)
13. Riesbeck, C.K., Schank, R.: Inside Case-based Reasoning. Erlbaum, Mahwah (1989)
14. Fredriksson, O.: Beyond Assessing Economic Net Value from Electronic Commerce-Supported Business Process Redesign – a Discussion of the PENG Method. In: Bubenko jr, J., Fischer-Hübner, S., Lindskog, S., Nilsson, A.G. (eds.) Proceedings of the Promote IT 2004 Conference, May 5-7, pp. 382–393. Karlstad University (2004)
15. McCarthy, W.E.: The REA Accounting Model: A Generalized Fra,ework for Accounting Systems in Shared Data Environment, The Accounting Review (1982)
16. Osterwalder, A.: The Business Model Ontology, PhD thesis, HEC Lausanne (2004), http://www.hec.unil.ch/aosterwa/ (accessed January 5, 2008)
17. Yu, E.: Modeling Strategic relationships for process engineering. Dep. Of Computer Science, University of Toronto (1995)
18. OMG Business Motivation Model, Version 1.1, Object Management Group (2010b), http://www.omg.org/spec/BMM/1.1/,maj2010 (April 19, 2011)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–106 (2004)
20. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2008)
21. Johannesson, P., Andersson, B., Weigand, H.: Resourse Analysis and Classification of Purpose Driven Value Model Design. Int. Journal of Information System Modeling and Design 1(1), 56–78
22. Kohli, R., Hoadley, E.: Towards Developing a Framework for Measuring Organizational Impact of IT-Enabled BPR: Case Studies of Three Firms. The Data Base for Advances in Information Systems 37(1) (2006)
23. Becker, J., Bergener, P., Räckers, M.: Process-Based Governance in Public Administration Using Activity-Based Costing. In: Wimmer, M.A., Scholl, H.J., Janssen, M., Traunmüller, R. (eds.) EGOV 2009. LNCS, vol. 5693, pp. 176–187. Springer, Heidelberg (2009)
24. Grimsley, M., Meehan, A.: e-Government information systems: Evaluation-led design for public value and client trust. European Journal of Information Systems 16, 134–148 (2007)
25. Chircu, A.M.: E-government evaluation: towards a multidimensional framework. Electronic Government 5(4), 345–363 (2008)
26. Irani, Z., Love, P., Elliman, T., Jones, S., Themistocleous, M.: Evaluating e-Government: learning from the experiences of two UK local authorities. Information Systems Journal 15, 61–82 (2005)

Grounded System Dynamics: A Procedure for Underpinning System Dynamics with a Domain Modeling Method

F.P (Fiona) Tulinayo[1], P. (Patrick) van Bommel[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
[2] Public Research Centre Henri Tudor, 29 avenue John F. Kennedy
Luxembourg-Kirchberg, Luxembourg
F.Tulinayo@science.ru.nl, pvb@cs.ru.nl, e.proper@acm.org

Abstract. In this paper, we present a procedure Grounded System Dynamics (GSD) which we use as a guide to underpin a System Dynamics (SD) model with a domain modeling method called Object-Role Modeling (ORM). GSD is a combination of two existing methods (SD with ORM). By combining these two methods we generate synergy effects by using already existing modeling methods and by so doing we overcome some of the weaknesses of SD model building. Secondly, transformation of information from an ORM model of dynamic domains into an SD model is achieved. To apply the GSD procedure to a real-life case (Mukono Health Center (MHC), we use SD-ORM mapped constructs. As a result from the GSD procedure application, is an SD model to which we define quantitative foundations that result into simulations. Our approach(GSD) has been validated using case studies one of which is described in this paper. From this conclusions are drawn.

Keywords: System Dynamics, Object-Role Modeling.

1 Introduction

System Dynamics as a method was developed by Jay Forrester and dates as far back as 1961. A review and history can be found in [8]. It combines both quantitative and qualitative aspects [19] to explore complex models. In SD qualitative models, the structural features of the process are made explicit through casual loop diagrams [15]. In this paper however, we draw our attention to the quantitative aspect where stock and flow diagrams [7] are created and sets of equations input into the stock and flow diagram resulting into simulations. These simulations allow a modeler to provide quantitative estimates of system effects [15]. There have been earlier comparative studies between SD and other methods [21,6,2,4,3]. In these studies, a number of views on how SD relates or can work with these methods are given. In the context of enterprise modeling SD is typically used in process analysis design and optimization.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 112–125, 2011.

Consideration of many variables and complication in untangling the question of causation makes complex [28] system dynamics models hard to conceptualize and describe. Richardson [20] particularly states that; "*.....of the three tasks-model conceptualization, model formation and model understanding. Unquestionably the two most difficult are the two that are least formal- conceptualization and understanding. The future of the field needs software support for understanding the links between stock and flow/ feedback structure and dynamics behavior."* From this statement we note that, model conceptualization is one of the main issues in SD modeling. The issue of SD model conceptualization is further emphasized in [16,20,13,18]. Sharif [22] further states that; *"....there is a strong case for starting to apply systems dynamics methods more openly in the BPM and MIS research fields, as I feel the tools and techniques available are vastly under-rated in terms of their applicability and capability to provide novel representations of real-world situations....."*. In this statement we see Sharif justifying why SD should be combined with other methods. In this research therefore, we contribute to addressing the issue of SD model conceptualization by seeking support from domain modeling in the creation of SD models.This is because domain modeling identifies relationships among all entities within the scope of the problem domain and provides a structural view of the domain hence, improving model conceptualization. As an example of a domain modeling language, we use ORM in particular because of its conceptual focus and roots in verbalization.

In figure 1 we present a logical framework where we show the strengths and gap in both ORM and SD. We also show why grounding SD with ORM is necessary and of what impact it is to the domain of enterprise modeling and strategic decision making.

The underpinning of SD with ORM will not only improve SD model conceptualization but also make input data reusable and transferable (from one system to another or from one organization to another).

ORM is a fact-oriented approach for modeling information and querying the information content of a business domain at a conceptual level [11]. ORM is comparable to Entity Relationship (ER) Diagrams in use [5]. It has a graphical constraint notation that is claimed to be far more expressive for data modeling purposes than, for example, Unified Modeling Language (UML) class diagrams or industrial Entity Relationships (ER) diagrams [11]. ORM takes a static perspective on the domain in the sense that it aims to capture the fact types and entity types that play a role in the (dynamic) domain, while SD takes a dynamic perspective in which the dynamic behavior of the domain is captured. When modeling ORM only, the information for the dynamic perspective (the richness of an SD model in this regard) is missing.

To underpin SD with a domain modeling method, we use the design science approach because it focuses on first clarifying the goals of the artifacts (which in this case is GSD) and then on building and carefully evaluating the utility of the artifacts, and to a lesser degree, their reliability and validity [12]. Design science approach further places additional emphasis on the iterative construction and evaluation of artifacts which in this case are; the GSD procedure and its resulting

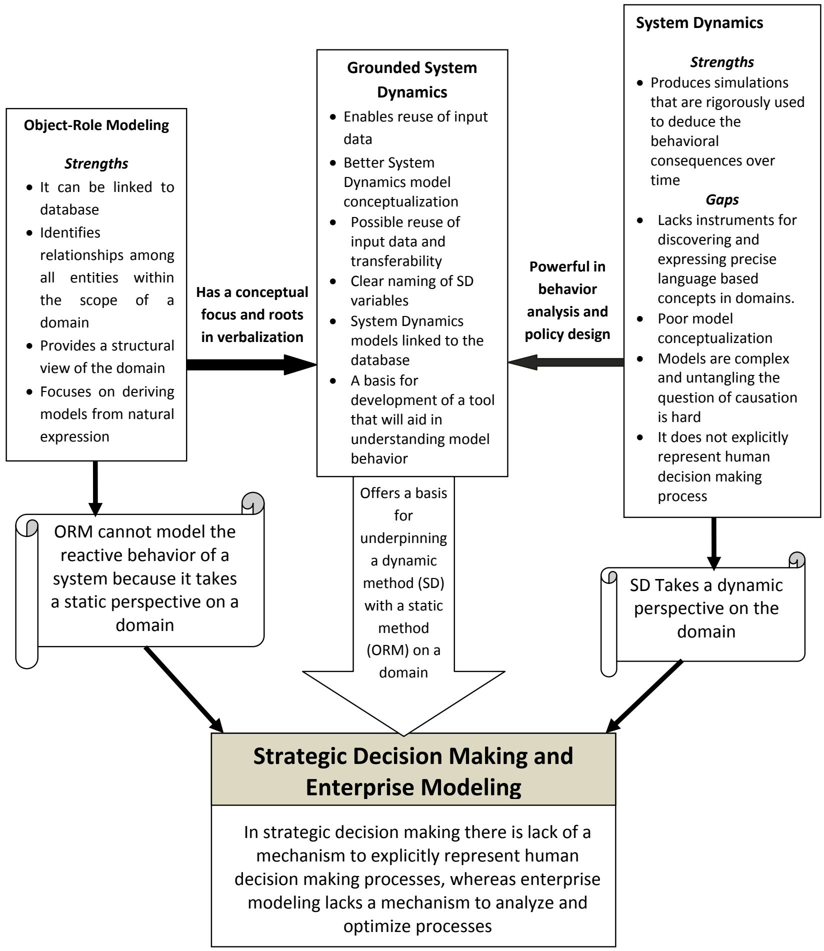


Fig. 1. Research problem and underlying principles

model(s). Design science approach also aims to ensure that the artifact (GSD procedure) is well grounded in both theory and empirical evidence to establish its validity, reliability, and practical utility. In this study we have so far identified the extent to which features of ORM static models can be transformed into SD models [26]; mapped ORM and SD concepts [27] and presented an investigation on the update behavior of the two methods [25].

The aim of this paper therefore, is to use the mapped SD-ORM concepts presented in [27] as a basis for this procedure. This procedure is what we follow while underpinning SD model(s) with a domain modeling method. We also apply this procedure (GSD) to a case where we input quantifications that lead to simulation results. These simulation results provide information about the problem domain, allow continuous testing of assumptions and sensitivity analysis of parameters [17].

The rest of the paper proceeds as follows; In sections 2 and 3 respectively brief introductions to SD and ORM are given. In section 4, a GSD outline is given and applied with the help of a case. In subsection 4.3, snapshots of some of the simulation results are presented and discussed. Finally, in section 5 conclusions plus hints for further work are given.

2 Brief Introduction to SD Stock and Flow Diagrams

The system dynamics stock and flow diagram[1] elements include: stocks, flows, feedback loops (connectors or information links) and converters. In fig 2 we have stock '*Admitted patients*' depicted as a box and defined as a container (reservoir) containing quantities describing the state of the system. The value of a stock changes overtime [3] through flows. Flows can be imagined as pipelines with a valve that controls the rate of accumulation to and from the stocks. They are represented as double solid lines with a direction arrow. The arrow indicates the direction of flow into or from a stock see '*patient*' in fig 2. Flows are influenced by stocks and converters. Converters either represent fixed quantities (constants) or variable quantities (Auxiliaries). Auxiliary variables are informational concepts having an independent meaning that contain information inform of equations or values that can be applied to stocks, flows, and other converters in the model [14]. An example of an auxiliary in fig 2 is '*Patient arrival*'. Constants are state variables which do not change [3] for example *PatientName*. On the STELLA SD software which we are using, both auxiliary variables and

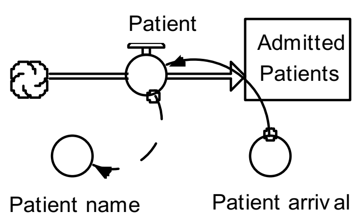


Fig. 2. SD basic building blocks

constants are depicted as small circles. Information from auxiliaries, constants and flows, is shared through connectors (information links). Two types of connectors exist, the action connectors (depicted as solid wires) and information connectors (depicted as broken wires) [24] *see* connectors from '*Patient arrival* to *patient*' and from '*Patient* to *patient Name*' respectively. These connectors are immaterial and link inputs to decision function of a rate. The underpinned meaning to these connectors is that information about the value at the start of

[1] For SD terminologies used in this paper we use [23] and all SD models are drawn using an SD STELLA 9.0.2 software. This is because STELLA is easy to use and offers a practical way to dynamically visualize and communicate how complex systems and ideas really work.

the connector influences information at the arrow tip of that connector. Connectors can feed information into or out of flows and converters but only extract information out of stocks [14]. Lastly, we have the sectors which are subsystems or subcomponents within a system. They hold/handle all decisions, stocks, information about a particular element or area and contain different information that is used in an information system. Sectors have not been represented in fig 2, but can be seen in fig 9.

3 Brief Introduction to ORM

ORM basic building blocks are entity types, value types and roles [10][2]. An object type is a collection of objects with similar properties, in the set-theoretical sense. Objects are things of interest, they are either entity or value types. Object types are designated by solid-line named ellipses see; *Patient* and *Labor Ward* in fig 3. Object types have reference modes, *see fig* 3, where object type *Patient* has reference mode *id*. This reference mode indicates how a single value relates to that object type. Instances of value types are constants with a universally understood denotation, and hence require no reference scheme. They are identified solely by their values, their state never changes and are designated by dotted ellipse see *Patient Name*. The semantic connections between object types are depicted as combinations of boxes and are called fact types. Each box represents a role and is connected to an object type or a value type. The roles denote the way entity types participate in that fact type. The number of roles in a fact type

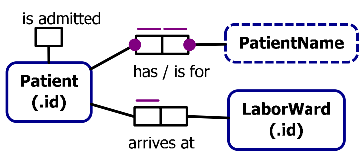


Fig. 3. ORM basic building blocks

is referred to as fact type arity and the semantics of the fact type are put in a fact predicate. A predicate is basically a sentence with object holes in it, one for each role as depicted in fig 3 (see; *arrives at*, *is for*, *is admitted*) etc. These predicate names are written beside each role and are read from left to right, or top to bottom. It is through predicates that entity types relate to each other. Note that the constraints in fig 3 are not discussed because they are outside the scope of this paper. But these and more can be found in [10].

4 Grounded System Dynamics (GSD)

After mapping the SD-ORM constructs [27], we now present a procedure which we are to follow while underpinning SD models with ORM. To achieve this, we

[2] For ORM terminologies in this paper, we use [10] and to model ORM models we use Microsoft Visual Modeler 2005.

conducted multiple discussions, read various scholarly works and in a step by step manner, we were able to come up with this procedure which we refer to as Grounded System Dynamics (GSD). GSD is a combination of two existing methods (SD with ORM). By combining these two methods we generate synergy effects by using already existing modeling methods and by so doing we overcome some of the weaknesses of SD model building. Secondly, transformation of information (data) from an ORM model of dynamic domains into an SD model is achieved. Underpinning of SD with ORM is done by following five steps which we outline is subsection 4.1. In these steps we map different SD constructs to ORM constructs and explain how they relate to one another. For a clear guide and explanation, we apply this outline to a case study of Mukono Health Center (MCH) in subsection 4.2.

4.1 GSD Procedure Outline

In this subsection we describe the steps followed in transforming ORM model concepts into SD model concepts.

1. Identify all possible stocks.
2. Identify all relevant flows.
3. Identify possible converters.
4. Identify possible connectors (information links).
5. Create sectors.

4.2 Case Study

At MHC we looked at the process pregnant women go through on their due dates. MHC receives an average of 250 deliveries per month. It has a total of *sixteen* maternal employees that is; *two* doctors who are available on phone in case of any emergency, *nine* nurses and *five* volunteers. There are *eleven* beds in total, available for admissions. Most of these beds are given to patients who have caesarian birth because they require a lot of attention and tend to stay longer at the health center.

The process: A patient comes to the labor ward with her antenatal card from the antenatal clinic. She queues up. Her waiting time depends on a number of factors including; her arrival time, the number of patients around and number of nurses on duty. When her turn comes, the nurse on duty takes her history and then examines her. This examination takes approximately 30 minutes. The nurse also establishes the patient's labor stage. If the patient is 4cm dilated, she is admitted to the general ward. She only returns for examination if there is any complication or after 4 hours. During this time, after every 30 minutes monitoring of the labor progress, status of the mother and cervical dilation is done. When the patient is 8cm dilate, she is taken to the delivery room which has only two beds. While there, the nurse monitors descending of the head 2 hourly and the sticker. For normal progress the liquor is clear. When the patient has 10 cm dilate, she is ready to give birth. After delivery, she is taken back to the general labor ward. Normal delivery patients stay at the labor ward for a maximum period of 24 hours and patients who have had caesarian birth stay for a period of 4-7 days. On discharge, the baby is taken for immunization. From this given information, we now construct an ORM model shown in fig 4.

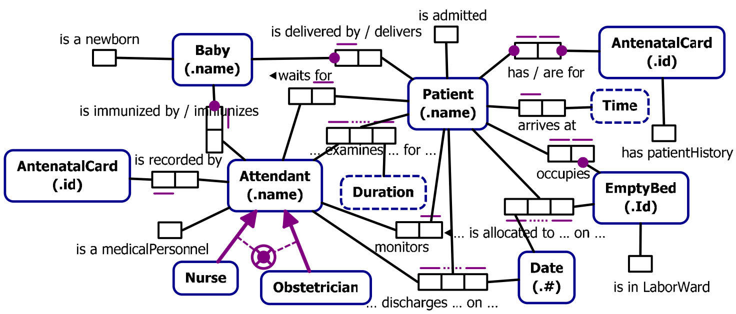


Fig. 4. MHC Labor suite developed ORM model

We use the developed ORM model in fig 4 to apply the GSD outlined procedure in subsection 4.1.

[***Step 1:***] *Identify all possible stocks*
In order for us to generate stocks, we identify an ORM element that has similar characteristics to a system dynamics stock; that is it holds items, accumulates and can be measured. The element found to have these characteristics is a *role*. But not all roles are mapped to a system dynamics stock. It's only the *unary fact types* that are mapped to a SD *Stock*. This is because they relate to one object type and contain objects from that particular object type. The total number of unary fact types therefore is equal to the total number of stocks in an SD model.

The identified unary fact types from fig 4 are; *newborn baby*, *medical personnel*, *admitted patient*, *Patient History* and *Labor ward* and are depicted as stocks in fig 5. Note that all unary fact types have words like '*is a*', '*has*' and '*is in*' at the beginning of each unary fact type predicate but when represented as stocks these prepositions are removed. This enables us make the stock name clear. Secondly, for some stocks for example; *newborn* and *admitted* we concatenate (join two character strings end-to-end) the unary fact type name with the object type name they relate with to get the stock name *newborn baby* and *admitted patient* respectively. This makes identification of the object type connecting to the unary fact type (stock) plus its quantification easy. Note that in fig 4, each object type has a unary fact type. This may not be the case with all ORM models but we

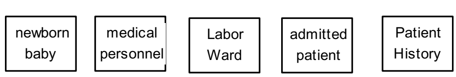


Fig. 5. Identified stocks from MHC Labor suite ORM model

advise the modeler to have a unary fact type attached to each object type so that objects in each object type have a store.

[***Step 2:***] *Identify all relevant flows*
The element identified to be similar to an SD flow in ORM is an *object type*. This is because it connects different roles. That is, for each role connection, objects held by that object type play a unique role. Flows in SD connect to different stocks and converters through connectors. The identified flows from fig 4 are '*Empty bed*', '*Patient*', '*Attendant*', '*Antenatal card*' and '*Baby*' and are represented in fig 6.

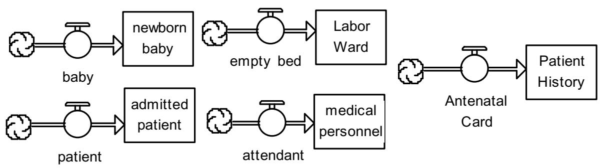


Fig. 6. Flows from MHC labor suite ORM model are connected to stocks

[***Step 3:***] *Identify possible converters*
Converters include constants and auxiliary variables. Auxiliary variables from a conceptual point of view are informational concepts having an independent meaning. They are similar to fact types that have more than one role.[3] This is because they combine two or more variables consistently that cause change to the recipient, have an independent meaning and relate to more than one element. The roles contained by these fact types have predicate names. We use these predicate names to name the auxiliary variables. This is done by concatenating the object type name with the fact type name. For example; roles '*examines*' and '*is examined by*' make a fact type which we refer to as '*examination*', this fact type name is concatenated to object type name *patient* giving us a flow name, '*patient examination*' see fig 7. We also include value types which we map to constants in SD. This is because a value type is identified solely by its value

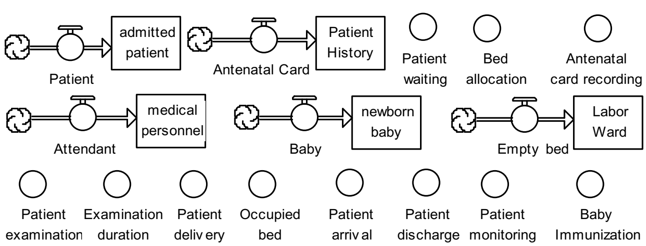


Fig. 7. Auxiliary variables identified from binary and ternary fact types

[3] In this paper we only consider *binary* and *ternary* fact type to be similar to auxiliary variables but all fact types with more than one role are referred to as auxiliary variables

and it never changes its state (i.e. it is a constant). On the other hand, constants in SD are state variables which do not or change slowly [3] that they could be assumed constant for the time scope of the model. Note that the roles played by these value types are not included in the model as auxiliary because they are assumed constant.

[***Step 4:***] *Identify possible connectors (information links).*
After Identifying the converters, we now introduce the connectors. connectors are immaterial and connect inputs of a decision function to a rate. They are similar to predicators in ORM since they both act as connectors of two elements which are; object types to roles in ORM and, converters to flows and stocks in SD. With the help of ORM verbalization, the direction of the connectors is determined, that is from an object type to a role. Using some of the verbalization in fig 4 as examples, we show how these verbalizations help in identifying the direction of some connectors in fig 8. For example:

- We have verbalization *"Patient delivers baby".* In this verbalization, we have two object types which are mapped to SD flows. Here our focus is on the connector connecting auxiliary '*Patient delivery*' to flow '*baby*'. The direction of the arrow is determined by the way a modeler reads from role 'delivers' to object type (flow) 'baby'. Not forgetting that an auxiliary is made up of more than one role. Therefore, this implies that an auxiliary may have more than one connector either connecting to or from it.
- As a second example let us look at verbalization "*Antenatal card is recorded by Attendant".* In this verbalization we have object types (flows) '*Antenatal card*' and '*Attendant*'. In fig 8 we see that flow '*Antenatal card*' has a connector directional arrow facing auxiliary "*Antenatal card recording*". This is because the verbalization is read from object type '*Antenatal card*' to role 'is recorded by'. We also have another connector coming from flow '*Attendant*' to "*Antenatal card recording*" which comes from verbalization '*Attendant*' records '*Antenatal card.*

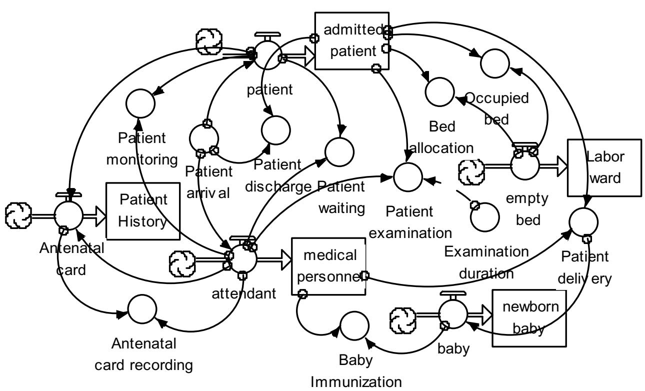


Fig. 8. Information Links introduced in SD MHC Labor suite SD model

We note here that, the verbalization alone cannot give us all the necessary connectors (it can only give us links from one direction but not feedback). We therefore advise the modeler(s) to further identify the feedback.

[***Step 5:***] *Create sectors*
In [9,1] it is stated that ORM conceptual object types act as semantic 'glue'. This means that an ORM model is a network of allied object types and relationship types [9]. Sectors on the other hand are SD elements and are subcomponents within a system that handle all information about a particular element. If they are subcomponents that means they contain different elements and when put ('glued') together they make a complete system. Therefore, object types plus their 'glued' roles are similar to SD sectors because when both are 'glued' or put together they make up a complete model (either ORM or SD) and contain more than one element. The total number of sectors therefore, is equal to the total number of object types in fig 4. In each sector we have an object type plus roles connected to it. The sectors identified from fig 4 are as follows:

Sector *Empty bed* which comprises of object type *Empty bed* plus roles; '*Is occupied by*', '*is allocated to*' and '*is in labor ward*'.

Sector *Attendant* which comprises of object type *Attendant* plus roles; *'Is a medical personnel'*, *'monitors'*, *'records'*, *'examines'*, *'discharges'*, *'is waited for by'* and *'immunizes'*.

Sector *Patient* which comprises of object type *Patient* plus roles; '*is examined by*', '*is discharged by*', '*is admitted*', '*is monitored by*', '*occupies*', '*is allocated*', '*is arrives*', '*has*', '*delivers*', and '*waits for*'.

Sector *Baby* which comprises of object type *Baby* plus roles; '*is a newborn*', '*is immunized by*', and '*is delivered by*'.

Sector *Antenatal Card* which comprises of object type *Antenatal Card* plus roles; '*has patient History*', '*is recorded by*', and '*is for*'.

Here we note that, fact types that make an auxiliary variable have more than one role. Therefore, auxiliary variables relating to more than one flow are put in one of the sectors with the flow it connects too.

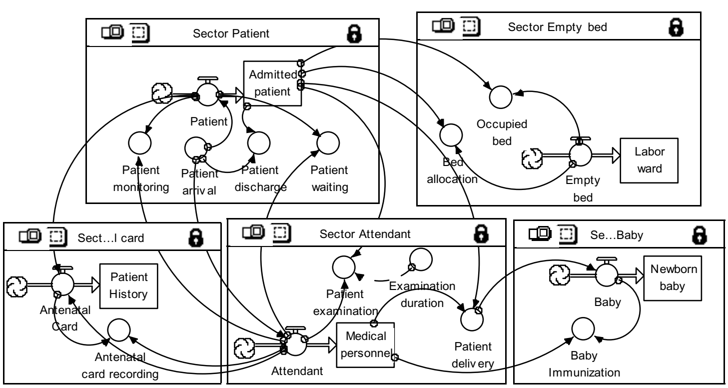


Fig. 9. Sectors identified from MHC Labor suite ORM model

In conclusion, having applied the GSD procedure to a case, the modelers achieved two main things: One, better conceptualization of the underlying SD model concepts. And two, an SD model that is underpinned with an ontology domain modeling method.

4.3 Simulation Results

In this section, we use figs 10 a) and 10 b) to show some of the resulting SD model simulations. To obtain these simulation results, we started by stating the initial values for all stocks. Then defined input quantities and formulas for all auxiliary variables starting with *patient arrival* which is an input to flow *patient*. Some auxiliary inputs for example *Patient waiting duration* are captured with a time delay. This delay function returns a delayed value input using a fixed time lag. For example in fig 9, we have variable *Patient waiting duration* where its input parameter is equal to *DELAY [Attendant, 30 minutes]*. This input parameter causes variable *patient waiting* lag behind variable *attendants* by 30 minutes. This means that for the first 30 minutes of the simulation, the delay returns the initial value of *Patient* since no initial value is specified.

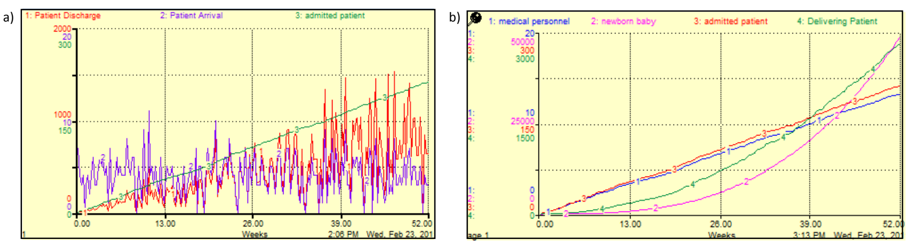


Fig. 10. Some of the simulation results of fig 9

For variables *patient arrival* and *patient discharge* in figure 10 a), we have oscillations. This is so because the quantitative measure to these variables has no defined limit. In fig 10 b), we see that over a period of time the gap between variables *medical personnel* and *admitted patients* keep on increasing. Secondly, in fig 10 b), we see that *delivering patients* are consistently increasing over a period of time and at the 51st week they rise above the *newborn babies*. The increase of *delivering patients* above *new born babies* at week 51 is an indicator that these simulation results are not completely reliable. This may be due to lack of outflows which drains the stocks.

From the input SD simulation algorithms whose choice depended on variables involved in the MHC labor suite SD/ORM model. In conclusion, the final simulation results in fig 9 do not give conclusive results. This is because some of the system dynamics construct e.g. outflows are not represented in the model. We believe that with more studies and application of the GSD procedure to different case domains, better simulation results will be achieved.

5 Conclusions and Further Work

This study is a work in progress and as such it is not possible to draw any firm conclusions on the end results. At the current stage it is possible to say that the GSD procedure is a good guide in transforming ORM models into SD models. This procedure gives the modeler better conceptualization of different variables plus reasons as to why a particular SD element is mapped to that ORM element.

Secondly, much as we have followed a GSD procedure to transform ORM model concepts into SD model concepts, we note here that, not all ORM and SD elements have been represented. For example; in SD we have elements like Conveyors, queues, decision process diamonds and Biflows missing and in ORM we have all constraints missing. Therefore, for further research we recommend that an investigation on how these missing elements can be used (mapped) to improve the GSD transformed SD model.

In step 2 of the GSD procedure, we realize that outflows are not represented in fig 6, yet in a system dynamics stock and flow model, a stock has both an inflow and outflow to drain the stock. Stocks accumulation is dependent on their flows and are mathematically calculated as the integration of net inflows:

$$Stock(t) \int_0^t [Inflow(s) - Outflow(s)]ds + Stock(t_0) \qquad (1)$$

with $Inflows$ and $Outflows$ denoting the values of the outflow at anytime s between the initial time t_0 and the present time t [23]. The net flow determines the rate of change of any stock:

$$d(Stock)/dt = Inflow(t) - Outflow(t) \qquad (2)$$

Therefore, we suggest that some auxiliary variables be referred to as out flows for example; *Patient discharge* can be used as an outflow from *admitted patient* and a new variable like *death* could be introduced in the model to act as an outflow from *newborn baby*. That way we have both inflows and outflows represented in the model.

So far, the GSD procedure has been applied to one case already and we will continue to apply it to other case studies. We will further refine the procedure and also devote more attention to coming up with a generic meta-model. Finally, a comparison of the SD model development process with the GSD procedure using various case studies will be done.

References

1. Bloesch, A., Halpin, T.A.: Conceptual queries using ConQuer-II. In: Embley, D.W., Goldstein, R. (eds.) ER 1997. LNCS, vol. 1331, pp. 113–126. Springer, Heidelberg (1997)
2. Borshchev, A., Filippov, A.: From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In: 22nd International Conference of the System Dynamics Society (2004)

3. Burmester, L., Matthias, G.: Combining System Dynamics and Multidimensional Modelling - A Metamodel Based Approach. In: Proceedings of the 14th Americas Conference on Information Systems, Toronto, ON, Canada (August 2008)
4. Chang, L.C., Tu, Y.M.: Attempt to Integrate System Dynamics and UML in Business Process Modeling. In: Sterman, J.D., Repenning, N.P., Langer, R.S., Rowe, J.I., Yanni, J.M. (eds.) Proceedings of the 23rd International Conference of the System Dynamics Society. System Dynamics Society, Boston (2005)
5. Chen, P.P.: The Entity-Relationship Model-Towards a Unified View Data. ACM Transactions of database systems 1(1), 9–36 (1976)
6. Duggan, J.: A Comparison of Petri Net and System Dynamics Approaches for Modelling Dynamic Feedback Systems. In: 24th International Conference of the Systems Dynamics Society, Nijmegen, The Netherlands (July 2006)
7. Elf, M., Putilova, M., von Koch, L., Ohrn, K.: Using System Dynamics for Collaborative Design: a Case Study. BMC Health Services Research 7, 123 (2007)
8. Forrester, J.W.: Industrial Dynamics. MIT Press, Cambridge (1961)
9. Halpin, T.A., Bloesch, A.: Data modeling in UML and ORM: A comparison. Journal of Database Management 10(4), 4–13 (1999)
10. Halpin, T.A., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2008)
11. Halpin, T.A., Wagner, G.: Modeling Reactive Behavior in ORM. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 567–569. Springer, Heidelberg (2003)
12. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
13. Lane, D.C.: The Emergence and use of diagramming in system dynamics: a critical account. System Research and Behavioral Science 43, 1135–1150 (2008)
14. Leaver, J.D., Unsworth, C.P.: System Dynamics Modeling of Spring Behavior in the Orakeikorako Geothermal Field. Elsevier Ltd 36(2), 101–114 (2007)
15. Melao, N., Pidd, M.: A Conceptual Framework for Understanding Business Processes and Business Process Modelling. Information Systems Journal 10, 105–129 (2000)
16. Morecroft, J.D.W.: A Critical Review of Diagramming Tools for Conceptualizing Feedback System Models. Dynamica 8(1), 20–29 (1982)
17. Morecroft, J.D.W.: System Dynamics and Microworlds for Policymakers. European Journal of Operations Research 35(3), 301–320 (1988)
18. Papageorgiou, G.N., Hadjis, A.: New Planning Methodologies in Strategic Management; An Inter-Paradigm System Dynamics Approach. In: Dangerfield, B.C. (ed.) Proceedings of the 26th International Conference of the System Dynamics Society. System Dynamics Society, Athens (2008)
19. Pidd, M.: Tools for Thinking: Modelling in Management Science, 2nd edn. John Wiley, The Atrium (2003)
20. Richardson, G.: Problems for the future of system dynamics. System Dynamics Review 12, 141–157 (1996)
21. Scholl, J.: Agent-based and System Dynamics Modeling: A call for cross study and joint research. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICS), Maui, Hawaii, vol. 3, pp. 1–10 (January 2001)
22. Sharif, A.M.: Industrial Viewpoint can systems dynamics be effective in modeling dynamic business systems? Business Process Management Journal 11(3), 612–615 (2005)
23. Sterman, J.D.: Business Dynamics- Systems Thinking and Modeling for a Complex World. McGraw Hill Higher Education, New York (2000)

24. Tan, K.S., Ahmed, M.D., Sundaram, D.: Sustainable Enterprise Modelling and Simulation in a Warehouse Context. Business Process Management Journal 16, 871–886 (2010)
25. Tulinayo, F.P., Groessler, A., Hoppenbrouwers, S.J.B.A., van Bommel, P.: Complementing System Dynamics with Object-Role Modeling. In: Ford, A., Ford, D.N., Anderson, E.G. (eds.) Proceedings of the 27th International Conference of the System Dynamics Society. System Dynamics Society, Albuquerque (2009)
26. Tulinayo, F.P., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Integrating System Dynamics with Object-Role Modeling. In: Stirna, J., Persson, A. (eds.) The Practice of Enterprise Modeling, vol. 15, pp. 77–85. Springer, Heidelberg (2008)
27. Tulinayo, F.P., Hoppenbrouwers, S.J.B.A., van Bommel, P., Proper, H.A.: Integrating System Dynamics with Object-Role Modeling and Petri Nets. In: Mendling, W.E.J., Rinderle-Ma, S. (eds.) Enterprise Modelling and information systems Architectures, GI-edition, Ulm, Germany. IFIP, pp. 41–54 (September 2009)
28. Zhang, Z., Jai, L., Chai, Y.: A New Kind Methodology for Controlling Complex Systems. International Journal of Engineering Applied Sciences 6(2) (2010)

Enterprise Modeling in an Agile World

Harald Wesenberg

Statoil ASA, Forusbeen 50, N-4035 Stavanger, Norway
hwes@statoil.com

Abstract. As the pace of business increases, the speed at which enterprise models must be delivered increases accordingly. Enterprise modelers cannot spend years in an attic developing perfect models, but must deliver models that are useful in time to be used. In this paper I will take a look at the experiences we have with enterprise modeling in Statoil, a global oil company headquartered in Norway.

Keywords: Enterprise modeling, Agile, PoEM2011.

1 Introduction

This paper accompanies my keynote from the Practitioners of Enterprise Modeling 2011 conference (PoEM 2011) held in Oslo, November 2011. In the keynote, I take a look at the experiences we have had with enterprise modeling in Statoil, and how our practices has changed with the emergence of agile software development practices and business models. As befitting a keynote, this paper is not based on years of scientific research. It is a summary of observations, experiences and reflections I have made over the years, forged into an opinionated keynote at a conference. In no way does this paper reflect official Statoil policies on software development, enterprise modeling or anything else.

The paper is structured as follows:

- **Section 2** defines what an enterprise model is in the context of this paper and compares this to other models typically found in an enterprise
- **Section 3** looks at some of the lessons we have learned from the past years of enterprise modeling in Statoil
- **Section 4** offers a conclusion and some recommendations

2 Defining Enterprise Models

Not all models are enterprise models. In a large enterprise, dozens, if not hundreds or thousands of models are created every day on whiteboards, paper and presentations to illustrate some aspect of the real world.

An enterprise model, on the other hand, has certain characteristic that differentiates it from other models:

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 126–130, 2011.

- **Enterprise Models are for Communication through Time or Space.** An enterprise model exists over a longer period of time, and is distributed widely throughout the enterprise. It is often found in a corporate repository of sorts, and is accessed through web portals or similar. Although many software developers create models as part of their development work, few of these models are used over a longer period of time, or outside that team of developers. Hence, such models are not enterprise models.
- **Enterprise Models are Abstractions.** Although all models are abstractions of the real world [1], the concept of abstractions is particularly important for enterprise models. They often cover an enormous amount of complex enterprise knowledge that cannot be easily transferred to a model. Therefore, abstractions are necessary when developing enterprise models, else they will be too complex to be of use. If the wrong abstraction is chosen a model will never become an enterprise model, but instead gather dust in a drawer somewhere.
- **Enterprise Models are Managed.** Enterprise models are (or at least should be) managed properly. They are often subject to strict versioning routines, configuration management practices and release plans. In many ways enterprise models are similar to source code, and should be subject to the same professional practices. If the models are not managed properly they will not be trusted and they will subsequently fail to achieve their full potential as enterprise models.
- **Enterprise Models Must have the Right Quality.** Model quality can be measured in many ways. I have found that it makes sense to talk about three dimensions of model quality: Syntactic quality (how well the model uses the modeling language), semantic quality (how well the model reflects the real world) and pragmatic quality (how well the model is understood by the target audience) [4,5]. In enterprise models the balance between these dimensions becomes very important; else the model will not be used by its intended target audience.

Although not all enterprise models exhibit all of these characteristics equally, they are useful as a means of distinguishing enterprise models from other models.

2.1 Introducing the Agile World

Agile means a lot of things to different people. In the context of this paper (and keynote) I define Agile to reflect the way the pace of change increases in many areas of life, forcing individuals and organizations to change with it. Software that were released in a yearly release cycle 10 years ago is now released twice a day [2], business models change frequently, and companies are on a path of continuous change as they acquire, merge and divest at an increasing speed.

For enterprise modelers, this increasing pace of change means that enterprise models must be developed faster and faster to be useful. They must be managed to stay current, and they need to be in the right place at the right time to have an impact.

3 Enterprise Modeling in Statoil

Statoil is a global oil company headquartered in Norway. It has more than 20.000 employees in more than 30 countries worldwide, and has spent significant resources in various forms of enterprise modeling over the years. We have achieved a fair success with enterprise modeling in its corporate management system [3] where workflow models are used extensively to communicate requirements and best practices throughout the enterprise. The current management system contains some 1500+ business process and workflow models with associated requirements and best practices, all available through a corporate web portal anywhere in the company. The models are used daily in many parts of the organization, and are a significant contributor in reducing operational, environmental and safety risks etc.

In this section I will review some of the experiences and observations we have made over the years of developing enterprise models.

Every Model is a Journey. All enterprise models develop over time. As the world it reflects changes, the models changes with it. This journey lasts for the lifespan of the model, and throughout the journey effort must be made to ensure that the model is current and useful.

When setting out to develop an enterprise model, be sure to have someone with thorough knowledge of the real world (domain) it reflects as part of the development team. This ensures that the right details are kept as the real world is abstracted into an enterprise model. Another important part on the team is played by the youngster that is current on best practices for modeling and is capable of challenging the domain experts on their assumptions and beliefs. Only when they work together will the journey be a successful one.

Every Model Tells a Story. A model tells a story to its target audience. This story will change over time as the model is developed, but the core of the story must be known before modeling starts. I have seen many models fail to meet their purpose because the core story has not been clearly understood by the development team.

Every Model has an Audience. The story of the model and the abstractions used in developing the model is determined by the needs of the stakeholders for the model, in particular the audience. Many times, I have seen models usefulness virtually destroyed by the need of the model development team to include too many details. The development team lacks the ability to see which aspects of the model that makes it useful to the target audience, and instead makes a model that is useful for them.

Everything Changes with Scale and Time. When the number of enterprise models grows into the thousands and models must be kept for many years, a lot more effort must be put into model administration and maintenance. Models must be managed through versions, branches and applicability so that the model library is current and useful for the end users.

All Models are not Enterprise Models. As a development effort (software, business or organizational development) is running, a lot of models are developed during the course of the effort. All of these models are not enterprise models, and very few of them should be brought back into the library of enterprise models. Only the models that will be widely used throughout the enterprise for a period of several years should be transferred to the enterprise model library, while the rest should be discarded or archived. This reduced the effort that needs to be spent on maintenance and administration of the enterprise models.

4 Recommendations

Based on my experiences with enterprise modeling in Statoil and elsewhere, I have a few recommendations to model development teams, as they set out to develop a model.

Know Your Audience and Tell Them the Right Story. A model is developed to tell a story to a set of end users. This story governs the entire life cycle of the model, and a model developed to tell one particular story will rarely tell another story just as well. Thus, the end user audience must be known and studied before modeling begins, to ensure that the model tell the right story.

Be Pragmatic. When determining what story to tell and which abstractions to make during the development of a model, it is important to keep the most important aspect of a model in mind: A model is useless unless it is understood by its target audience. Therefore, the development must be pragmatic in its modeling approach both when it comes to abstractions and content. Only include model elements that are useful to the target audience, and hide details that only serve to clutter the model. Even though these details may seem important to the development team, they may confuse the target audience and thus make the model useless.

Don’t Get Lost in the Details. Tied to the previous point, it is more important to get the level of details right. With too many details, the development will take too long, and the model will be outdated before it is finished. Better to have a conceptual model that is understood by most target audiences and delivered on time for it to be useful, than have a detailed logical model developed too late and outdated from the start.

Retire Models When They Are no Longer Useful. An enterprise model must be kept current to be useful. This means that all models have an inherent debt, i.e. resources that must be spent at some point in the future. A model that is not current reduces the confidence in the model library, and all models in the library should be reviewed periodically to ensure that they are current and still in use.

References

1. Jackson, M.A.: Problem Frames: Analysing & Structuring Software Development Problems. Addison-Wesley Professional, Reading (2000)
2. Bosch, J.: Architecture in the age of composability. Keynote SATURN 2011, San Francisco, California (May 2011), http://www.sei.cmu.edu/library/abstracts/presentations/bosch-saturn2011.cfm

3. Wesenberg, H., Landre, E.: Using the corporate Management System to communicate architecture. Presentation SATURN 2011, San Francisco, California (May 2011), `http://www.sei.cmu.edu/saturn/2011/abstracts-presentations.cfm#25`
4. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modelling. IEEE Software 11(2), 42–49 (1994)
5. Carlsen, S., Krogstie, J., Sølvberg, A., Lindland, O.I.: Evaluating Flexible Workflow Systems. In: Hawaii International Conference on System Sciences (HICSS-30), Maui, Hawaii (1997)

Aligning TOGAF and NAF – Experiences from the Norwegian Armed Forces

Håvard D. Jørgensen[1], Tore Liland[2], and Stein Skogvold[3]

[1] Commitment AS, P.O. Box 534, N-1327 Lysaker, Norway
Havard.Jorgensen@Commitment.no
[2] Norwegian Defence Logistics Organization, Rødskiferveien 20, N-1352 Bærum, Norway
TLiland@mil.no
[3] Acando AS, Lille Grensen 5, N-0159 Oslo, Norway
Stein.Skogvold@Acando.no

Abstract. This paper reports on experiences from establishing a reference architecture framework for the Norwegian Armed Forces. Like a number of other nations and NATO agencies, the armed forces chose TOGAF as their architecture development methodology (ADM), and the NATO Architecture Framework (NAF) for metamodel and content organization. In order to make TOGAF and NAF work together and address the particular requirements of the armed forces, significant adaptation was required. Previous work has analyzed the combination of TOGAF and military frameworks on the high level, but no detailed mapping between TOGAF 9 and the NAF, DoDAF, or MODAF architecture content frameworks were available. Such a mapping is presented here. The resulting framework has been implemented as a set of UML profiles, and as the content structure for the national military architecture repository. It has been applied by a number of initiatives, ranging from enterprise capability maps to technical interoperability between systems and acquisition projects.

Keywords: Architecture frameworks, TOGAF, NAF.

1 Introduction

This paper presents experiences from ongoing work at the Norwegian Armed Forces. The goal is to deliver a reference architecture for the networking and information infrastructure (NII), with the methodology, guidelines and competence needed to sustain it. The paper focuses on the adaptation of the architecture framework.

This is a case study, which focuses on the aspects of NAF and TOGAF that needed to be adapted to fit the needs of a particular organization. Previous analyses that has linked TOGAF ADM to military architecture frameworks [2,12,13] have taken a broader and more high level perspective, without drawing on experience from actual implementation of a combined framework. They have not dealt with implementation details such as metamodels and repository structures. In addition to practical relevance for other nations and agencies that seek to apply TOGAF with a military

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 131–146, 2011.

architecture framework, our work sheds light on the different perspectives, strengths and weaknesses of TOGAF and NAF.

The next section describes the background of the work, the use of enterprise architecture descriptions by the armed forces, and introduces NAF and TOGAF. Section 3 outlines the integration and adaptation that went into designing a customized architecture framework. Section 4 reports on implementation and usage experiences, while section 5 proposes directions for further development of practical architecture frameworks, as well as implications for future research.

2 Background

The key objectives for the architecture efforts of the Norwegian armed forces cover four levels: **Capability** planning and strategy development; **Project portfolio** management, migration planning, and investment decisions; **Project** management and inter-project coordination; **Solutions development** of secure, interoperable and flexible systems.

Up until now, the main focus has been on solutions development, and a major aim of the project reported here was to extend the use of architecture to cover also the higher levels. The armed forces have more than 10 years of experience with enterprise architecture (EA). It commissioned the development of a customized architecture framework called MACCIS [3, 8], and later participated in the development of NATO architecture standards, including NAF.

Top level management is committed to architecture. The strategic IT plan of the Chief of Defense describes architecture as a key enabler, and the architecture plan describes how to use “architecture as a methodology for describing complex relationships in a network-based defense and apply these descriptions as a foundation for management and decision making”. The Department of Defense has developed a NII reference model, and identified core areas for improvement like service orientation, modularization, interoperability, standardization, and reduction of the number of system variants for different user communities and platforms.

In the armed forces, the architecture responsibility is distributed between the IT department (INI) and the logistics organization (FLO). INI is responsible for the functional architecture, while FLO is responsible for the technical architecture. A governance structure is in place. The Architecture Advisory Board has the whole Networking and Information Infrastructure (NII) as its area of responsibility, and holds regular meetings to assess the architectural implications of new projects. At FLO, the Architecture Forum plays a similar role for projects in the acquisition/development phase, and evaluates standards before ratification.

In addition to the formal architecture governance organization, there are local architecture initiatives. Most notably, the Norwegian Defense Research Establishment (FFI) uses architecture descriptions in the development and validation of new concepts, and the common administrative systems project (LOS) uses ARIS to design their SAP adaptations.

2.1 NAF

In NAF [8], NATO defines four kinds of architectures. The *overarching* architecture should look several years into the future and answer the questions of *what* the enterprise is doing, and *why*. A *reference* architecture typically covers a span of a few years, describing *how* the enterprise functions, leading to a set of different *target architectures* for solutions development, which covers the technical aspects (*with what?*). A *baseline* architecture describes the technical aspects of the current enterprise. The core of NAF is a set of views that describe different aspects of an architecture [8]:

- **All view** (NAV) sets the scope and context of the architecture, including the subject area and timeframe, doctrines, tactics, techniques, procedures, relevant goals and vision statements, concepts of operations, scenarios, and environmental conditions.
- **Capability view** (NCV) supports the process of analyzing and optimizing the delivery of military capabilities in line with strategic intent. It contains a capability taxonomy and dependencies between capabilities, augmented with schedule data and measures of effectiveness to enable the analysis of gaps, overlaps, and trade-offs.
- **Operational view** (NOV) is a description of the tasks and activities, operational elements, and information exchanges required to accomplish missions and realize the capabilities expressed in NCV.
- **Service-Oriented view** (NSOV) supports the development of a Service-Oriented Architecture (SOA). NSOV describes the services needed to support the operations described in NOV. A service is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer.
- **Systems view** (NSV) describes the technical systems and system interconnections, their structure, functionality, behavior, and quality. Organizational, material, hardware and software resources are covered in order to define the physical architecture that implements the logical views.
- **Technical view** (NTV) provides the technical systems implementation guidelines upon which engineering specifications are based. NTV includes a collection of standards, implementation conventions, rules, and criteria.
- **Programme view** (NPV) describes the relationships between capability requirements and the ongoing development projects. This information can be leveraged to show the impact of acquisition decisions on the architecture.

Each of these seven views is further decomposed into subviews, which are diagram types for the enterprise architecture models. NAF derives this core structure of views and subviews from the US Department of Defense Architecture Framework (DoDAF) [1]. It also includes additional views from the UK Ministry of Defence Architecture Framework (MODAF) [5], and NAF's metamodel is aligned with that of MODAF. NAF does not prescribe a detailed methodology, though users are advised to follow the guidelines of DoDAF.

2.2 TOGAF ADM

TOGAF ADM has matured over more than a decade of industrial experience. Until version 9, it was agnostic of architecture framework and metamodels. It has been widely used with frameworks from Zachman and various modeling tool vendors, and with customized frameworks developed by different industries and organizations.

TOGAF ADM consists of nine phases. The *preliminary* phase outlines vision, objectives and scope, and mobilizes resources for the main architecture development cycle, which covers the phases A to H. Though the phases are represented as sequential, activities within different phases are often performed concurrently. The ADM is iterative, over the whole process, between phases, and within phases.

The central activity of requirements management collects, organizes and feeds architecture requirements into the phases of the cycle. Phase A continues the preliminary work of defining the vision, objectives, principles, and scope of the architecture. Phases B, C, and D collect information and populate the architecture model with business, information systems and technology descriptions respectively, while phases E and F utilize the architecture to select and govern development projects. Phases G and H deal with the long term governance and change management of the architecture, respectively.

2.3 Related Work

Previous analyses have explored the use of TOGAF 8 ADM with DoDAF [12] and MoDAF [2]. These analyses form the foundation of our work in integrating the two frameworks. However, in order to define a fully functioning methodology, we also explored the new architecture content framework (ACF) developed for TOGAF 9:

- How the architecture products of this framework maps to NAF subviews. TOGAF connects its architecture products to the phases of the ADM.
- How the metamodel of ACF maps to that of NAF. This provides insights into e.g. how a service oriented approach is best realized.

We also looked at the revisions that NAF v.3 and 3.1 makes to previous DoDAF, MODAF and NAF versions. In total, this provides a more up to date and detailed reference than previous work [2,12,13].

3 Adapting and Integrating TOGAF ADM with NAF

At the start of our project, NAF had been selected as the standard architecture content framework, in order to interoperate with coalition partners. TOGAF was the chosen architecture development methodology. These approaches had however not been customized to the needs of the armed forces. Enterprise Architect from Sparxsystems, a UML tool, was selected as the standard modeling tool for the whole enterprise, and a NATO Architecture Repository (NAR) had been set up, storing XMI files in a version control system.

3.1 Approach

Standard frameworks like TOGAF and NAF can be used in a wide variety of organizations. However, before they can be effectively used together within an architecture project, tailoring at three levels is necessary.

1. **Framework:** Align the TOGAF ADM phases and activities with the content framework of NAF.
2. **Enterprise:** Tailor the frameworks for integration into the enterprise of the armed forces. This includes integration with project and process management frameworks, customization of terminology, development of presentational styles, selection, configuration, and deployment of architecture tools, etc.
3. **Project:** Adapt the framework for the stakeholders of each particular architecture project. Tailoring at this level will select appropriate deliverables and model views to meet stakeholders' concerns.

The scope of our project is the overall reference architectures for the armed forces, so we did not customize to any specific project. We followed this approach:

Framework Adaptation

- Resolve the differences in approach between the two frameworks.
- Adapt the detailed steps in each TOGAF phase to the content structure reflected in the NAF subviews.
- Establish a minimal set of principled mappings from TOGAF elements to NAF elements, one-to-many and many-to-many where necessary. This should remove any ambiguities uncovered above.

Adaptation to the Enterprise

- Define the purpose, scope and role of the reference architecture in the landscape of other architectures in the military sector.
- Define clearly the stakeholders and user roles for the reference architecture, their concerns and objectives.

Implementing the architecture framework

- Define metamodels for the modeling languages, in our case as UML profiles in Enterprise Architect,
- Establish template architecture content and navigation structures, in our case as package structures in an Enterprise Architect model,
- Establish the architecture repository, and structure it according to the content framework,
- Establish customized frames of reference for different diagrams, like the NNEC Services Framework [8] for service taxonomies,
- Provide example models of each diagram type, for training and support,
- Define a template project plan with the work breakdown structure of TOGAF ADM, in our case in Microsoft Project.

This section describes the framework adaptation results, while the next section deals with adaptation to the enterprise and implementation experiences.

3.2 NAF and TOGAF Approaches

As a starting point for adapting the ADM to NAF, Figure 1 helps us to understand the use of ADM in the landscape of different architecture descriptions in the armed forces. The same figure is found in NAF, which substitutes Architecture Vision with Overarching Architecture, Architecture Definition with Reference Architecture, and Transition Architectures with Target Architectures. This means that what TOGAF sees as an integrated architecture description constructed by a single ADM cycle, NAF envisions as a set of interrelated descriptions, each developed by different people for different purposes.

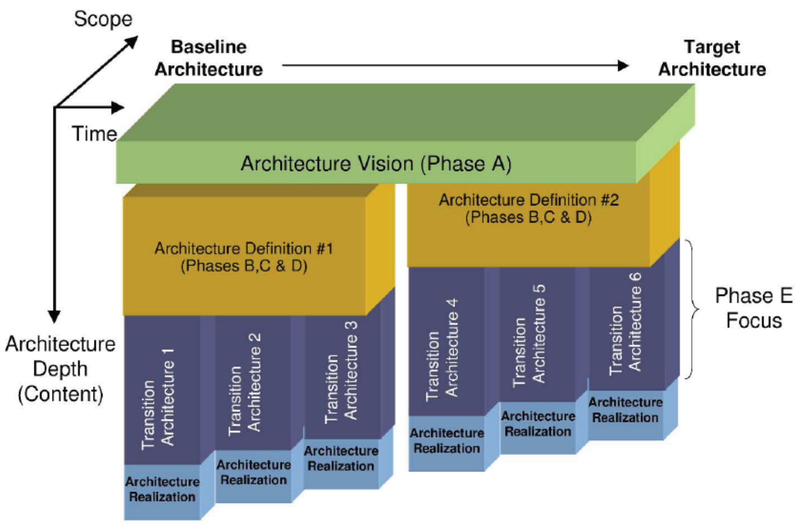


Fig. 1. TOGAF ADM phases and architecture content [11]

3.3 NAF and TOGAF Content Frameworks

Figure 2 below shown the views of NAF organized in the content framework of TOGAF. At this level the frameworks are well aligned. The only minor deviation is the conceptual information model, which TOGAF places in the data architecture, and NAF regards as an operational view. DoDAF v.2 [1] is better aligned with TOGAF in this area, through its Data and Information viewpoint. The motivation part of TOGAF's business architecture corresponds to NAF capability views, while operational views cover the organizational and functional aspects. NAF system views define most of TOGAF's IS and technology architectures, though NSOV should probably be used for high level services. At the bottom, technical views define implementation governance, while program views may be used for migration planning.

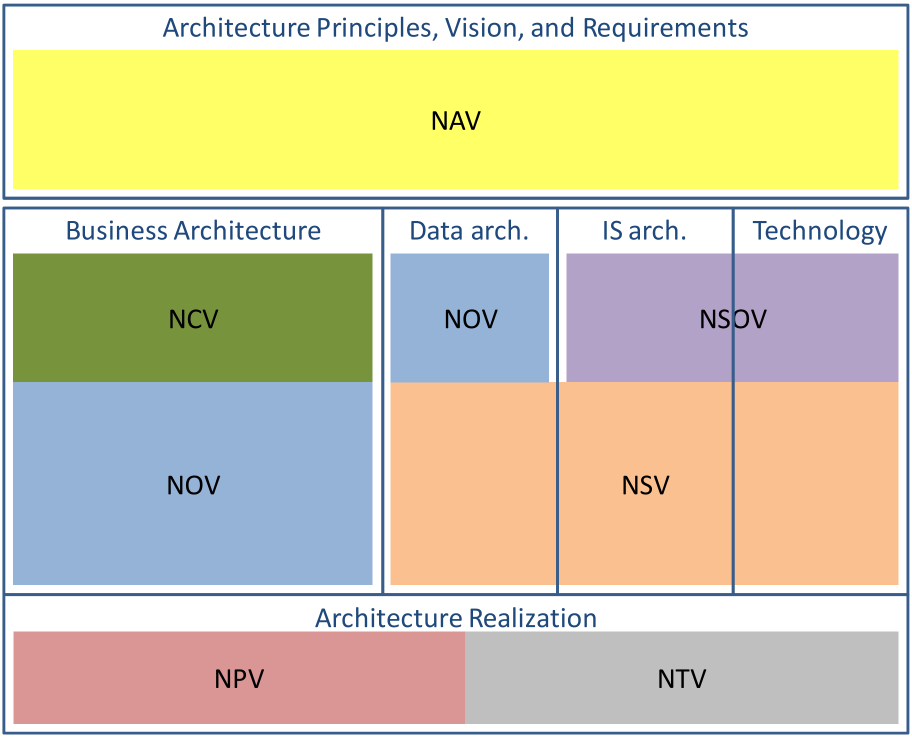


Fig. 2. Alignment of NAF and TOGAF Content Frameworks

3.4 NAF Subviews and TOGAF Architecture Products

In order to implement an architecture framework, however, we need to define precisely which architecture products to use. The devil is in the details, and when we approach the level of NAF subviews and TOGAF architecture products, the alignment of the two standards is no longer so straight forward.

As mentioned above, previous analysis [2,12,13] have mapped MODAF and DoDAF subviews to the phases of TOGAF ADM. With TOGAF 9, we have an additional resource for this mapping that these analysis did not, the TOGAF content framework (ACF) and metamodel. We therefore explored every TOGAF architecture product and identified suitable NAF subviews for each, using the metamodel types listed as corollary. The table below summarizes our mappings (v), and compares it to previous proposals (x) [2,12,13].

The differences between these mappings illustrate that TOGAF and NAF stem from different traditions, information systems and systems engineering respectively, and take different perspectives. Until you look into the detailed metamodels, these differences may not be so evident. Another important issue is that the high level mapping is mainly based on DoDAF, which compared to MODAF and NAF offers better support for an information systems perspective. The most important differences between the two mappings are:

			A. Architecture Vision	B. Business Architecture	C. Information Systems Arch.	D. Technology Architecture	E. Opportunities and solutions	F. Migration Planning	G. Implementation Governance	H. Architecture Change Mgmt.
NAV	1	Overview and summary	xv						x	
NAV	2	Integrated Dictionary		x	x					
NAV	3	Architecture metadata	x							
NCV	1	Capability vision	xv	xv					x	x
NCV	2	Capability taxonomy	x	xv					x	
NCV	3	Capability phasing				xv	x		x	
NCV	4	Capability dependencies		v		x	x		x	
NCV	5	Capability to organisational deployment	v	xv	x	x		x		
NCV	6	Operational activity to capability mapping		x						
NOV	1	High level operational concept description	xv	x					x	
NOV	2	Operational node relationship description		xv	v	v				
NOV	3	Operational information Exchange matrix		xv	v					
NOV	4	Organisational relationships chart	v	xv						
NOV	5	Operational activity model		xv	x			x	x	
NOV	6	Operational behaviour	v	xv	v					
NOV	7	Information model			xv					
NSOV	1	Service taxonomy		xv	xv					
NSOV	2	Service definitions		x	x	x			x	
NSOV	3	Capability to service mapping		xv	x					
NSOV	4	Service behavior		x	x	x				
NSOV	5	Service functionality		x	x					
NSOV	6	Service composition		v	v	x	x	x	x	
NSV	1	System interface description		v	v	xv				
NSV	2	System communications description			v	xv				
NSV	3	Resource Interaction Matrix				x				
NSV	4	Systems functionality description			xv					
NSV	5	System function to operational activity			xv	x				
NSV	6	Systems data exchange matrix			xv	v				
NSV	7	System quality requirements description				xv				
NSV	8	Systems configuration management			v	v	x	x		
NSV	9	Technology and skills forecast					x			x
NSV	10	Resource behaviour		v	xv	x				
NSV	11	System data model			xv	x				
NSV	12	Service provision		x	x		x			
NTV	1	Standards profile				xv				
NTV	2	Standards forecast				xv	x			x
NTV	3	Standard configurations				v		x	x	
NPV	1	Programme Portfolio relationships					x	x	x	
NPV	2	Programme to capability mapping				v	x	x	x	

Fig. 3. Assignment of NAF subviews to TOGAF phases

- Our detailed mapping does not find direct usage of All views in TOGAF, except NAV-1. This is thus an extension that NAF introduces.
- The TOGAF content framework does not provide much detail for the later phases (E-H), so here the earlier mapping is valuable.
- Our mapping finds more use for the operational views in the information systems architecture phase. This has to do with an emphasis on the logical, implementation-independent models of the applications and data, which are important for portfolio planning.
- Our mapping also finds more use for system views in the information systems architecture, in order to represent physical application components as well as the logical ones.
- Detailed service oriented views, defining functions, composition and behavior, do not have a clear counterpart in TOGAFs content framework, where operational and system diagrams seem sufficient for representing these aspects on the logical and physical layer, respectively.

The detailed mapping also illustrates some important features of NAF. First and foremost, some NAF subviews fill several different purposes, according to TOGAF ADM. The most important case is NSV-1, which can be used for at least 12 different architecture products: Application Portfolio Catalog, Interface Catalog, System/Organization Matrix, Role/System Matrix, Application and User Location Diagram, Software Engineering Diagram, Software Distribution Diagram, Technology Standards Catalog, Technology Portfolio Catalog, System/Technology Matrix, Environments and Locations Diagram, and Platform Decomposition Diagram. These products should be distinguished in the architecture models as different diagrams.

Vice versa, there are several TOGAF architecture products that require modeling of constructs from multiple NAF subviews. These products are candidates for customized views as extensions to the NAF content framework. The typical examples are diagrams that show connections from the logical architecture down to the physical, or from overarching capability views down to operational models. NAF supplies some of these, but not everything that TOGAF requires.

3.5 NAF and TOGAF Metamodels

An underlying issue in the architecture products mapping presented above, is the mapping between language constructs in NAF and TOGAF. Their metamodels are quite different. TOGAF presents a simple conceptual definition of a modeling language, with a core set of elements and five extensions. NAF defines a much larger metamodel as a UML profile. It is a technical implementation, fragmented into separate metamodel diagrams for each subview, and lacks a conceptual core that connects similar constructs across the views into a unified type hierarchy.

One critical issue that we had to resolve, was the situation where a single TOGAF construct could be mapped to a number of NAF constructs within different views. An example is given below, for Organization Unit. When defining scope and objectives, NAF models organizations as Enterprises, and links them to the phases or time spans that the architecture descriptions address. On the operational level, Nodes represent organizational actors, and you can also model actual organization units. In the physical systems architecture, organization types interact with other kinds of resources, and capability configurations represent the set of human and physical resources that together realize a capability, implement a node, or provide a service.

In addition to these direct representations of Organization Unit, NAF Capability can also be used for defining the functional composition and dependencies of the organization, at the logical level without relating it to concrete organizations.

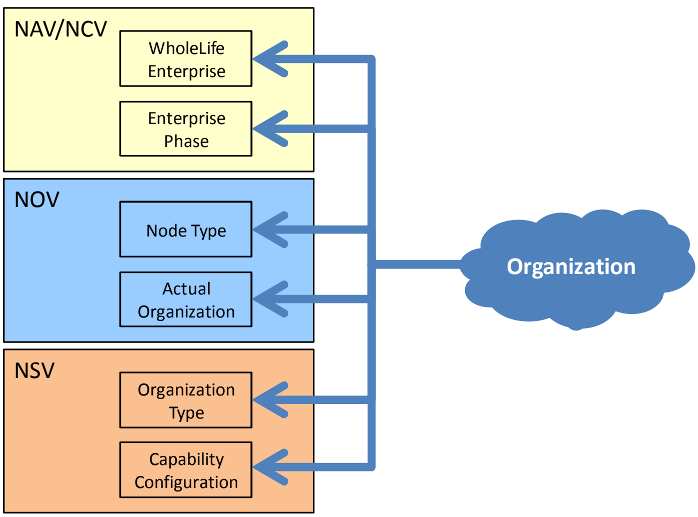


Fig. 4. Example of metamodel mapping problem

Similarly, a TOGAF Process can be mapped to a Mission, Enduring Task, Standard Operational Activity, Operational Activity, Service Function, and Function in NAF.

In addition to making it difficult to apply TOGAF ADM guidelines directly to a NAF architecture model, these metamodel mismatches causes a fragmentation of the architecture. Where TOGAF connects both scoping and objectives, logical operations and physical realization to a single Organization Unit element, NAF forces you to represent the organization as different elements in different views, and the framework does not even include all the links needed for linking these different representation together. These issues had to be resolved in our architecture framework implementation.

The most important metamodel mapping challenge that we encountered, however, was due to the more detailed and physical perspective that NAF takes, compared to TOGAF. Where TOGAF provides direct links for simple mapping between the core elements of the business, application, data, and technology architectures, it often takes several steps of more detailed indirect links to connect the same elements in NAF. For instance, rather than simply stating that a Project contributes to a Goal, in NAF you always have to state *when* (CapabilityIncrement, ProjectMilestone) the contribution happens, and *what* (CapabilityConfiguration) it consists of. If you model goals as EnterpriseGoals rather than Capabilities, you have two additional steps to go, via EnterpriseVision. Another example is the link between a Service and the data it uses. In NAF you must go from Service via ServiceInterface, ServiceInterfaceDefinition, ServiceOperation, and ServiceParameter before you arrive at the Entity in the information model. There are several examples like these, where we often have decided to add the direct relationships from TOGAF to our metamodel, in order to create a more high level, cost-effective, and sustainable model.

4 Implementation and Usage Experiences

The initial analysis of NAF and TOGAF, as reported above, identified a difference in their top level frameworks, focusing on

- Architecture layers (business, application, data, technology) in TOGAF,
- Depth of detail (overarching, reference, baseline, target) in NAF.

In order to integrate the frameworks, the Norwegian Armed Forces Architecture Framework in Figure 5 proposes a combination of these two dimensions, with NAF going down and TOGAF across. We have decomposed the business architecture of TOGAF into the NAF categories of capabilities, processes and organizations, which are similar to the three layers of business architecture in TOGAF. Similar frameworks are found in NATO documents [6], and in the Danish government's OIO "shelf system" [9], where the NAF layers are called conceptual, logical, and physical.

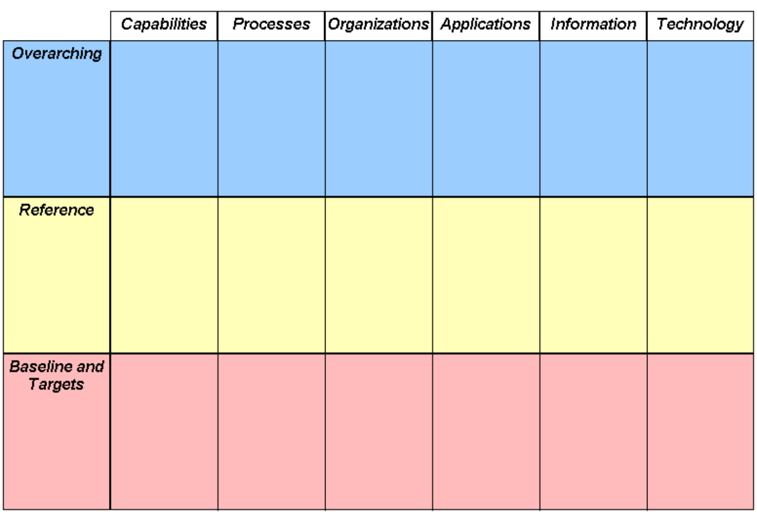


Fig. 5. The architecture content framework of the Norwegian Armed Forces

The organization principles of this two-dimensional framework are:

- *Vertical traceability* from one level to the next through specialization, decomposition, instantiation,
- *Horizontal traceability* on each level through various associations and dependencies, e.g. "uses", "supports", "is provided by",
- *Dependencies* between similar elements within each cell, e.g. communication relationships, information flow, and some decomposition and specialization hierarchies.

In our implementation, this matrix guides the organization of content for asset management in the governance framework. The upper level primarily contains NAF all and capability views, while the middle level deals with operational, service oriented, and program views. The lower level consists mainly of system and technical views.

The figure below places core concepts from the NAF metamodel in the architecture content framework. It provides high level guidance about how to model on the different levels. Some concepts are applied on more than one level, e.g. Capability, Service, and Entity. A common set of types are used for organization, application and technology resources on each level. In the overarching architecture, we focus on the services that these resources offer without bothering about their structure, while the reference architecture captures more detailed services and the nodes that perform them. Finally, on the physical level, ResourceType specializations are defined for organizations (organization, role, post), applications (software), and technology (artefact, physical architecture).

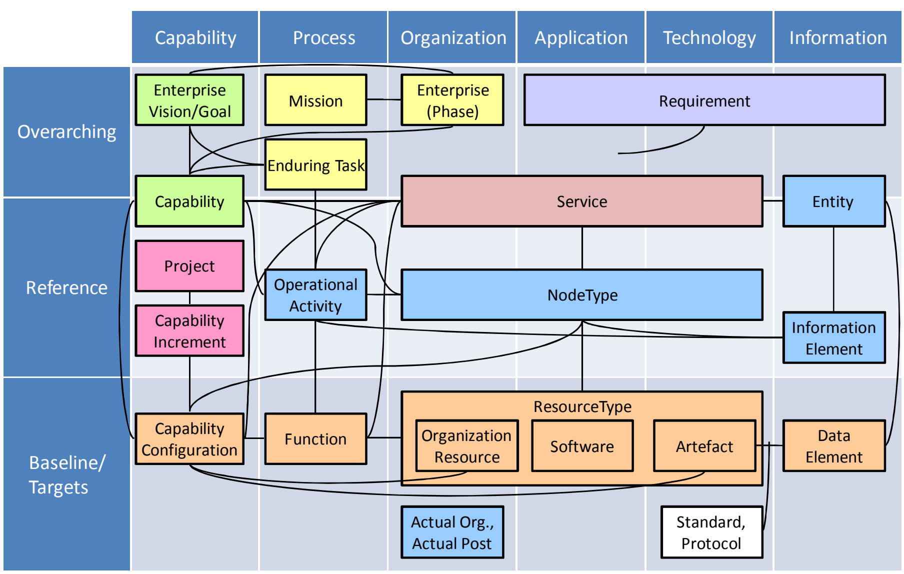


Fig. 6. Core NAF concepts in the architecture content framework

The architecture framework further defines which NAF constructs to use at the overarching, reference, target and baseline levels respectively, as well as which views and subviews should be emphasized. This solution combines framework adaptation with enterprise level adaptation, because we found the different layers of NAF to correspond well with the concerns of the core stakeholder groups in the armed forces. The overarching architecture is needed by top level management for planning long term capability development, while the reference architecture is used by the IT departments for business and IT alignment, project portfolio management, and managing system ownership. Target architectures are mainly applied by the IT departments and their suppliers in acquisition and development projects.

4.1 Implementation Experiences

The amount of work required for creating a fully functional implementation of a metamodel that is not supported out of the box by your tool vendor, should not be underestimated. In our case, we've had to make close to 150 changes to the NAF metamodel provided as a Sparxsystems model by the UK MOD [5], and several of the changes had to be replicated in a number of diagrams. Just a few of these changes were of a conceptual nature, many simply had to do with fixing omissions in the metamodel diagrams of each subview, so that the produced profile would be complete.

As it turned out, some of the ways in which MODAF and NAF uses UML constructs are theoretically valid, but practically not very useful. In particular, we changed the way several relationships are modeled. In the specification, NAF represents relationships in many different ways. Almost none of them use UML Associations, though some use Dependency. The constructs that created problems for us, however, modeled relationships between elements as Property (22 cases), TaggedValue (33 cases), Attribute (5 cases) or Slot (2 cases). In addition to the added complexity of representing relationships in so many different ways, these solutions were difficult to visualize and track in the tool, cumbersome to model because they could not be dragged directly from a profile toolbox, and not as interoperable with non-UML tools that support NAF. Rather than representing relationships the way you would if you were programming, we thus decided to switch to a more high-level modeling approach, using two way links rather than one way attribute type references.

The representation of concrete instances was another area where we chose to extend the standard NAF metamodel. NAF out of the box supports the modeling of actual organization elements and projects as instances. We however saw the need to model other actual elements in the same way, e.g. locations, IT resources, military vessels. We thus allow the modeling of instances of all types, but see no need for defining separate stereotypes for each of these instance types, when the type is already given by the stereotype of the classifier of the instance. In order to ensure a uniform and simple framework, we also decided against a common practice in the past of using instances that stand for their classifier in a given diagram, as pure symbolic instances.

A final important simplification was to remove stereotypes for behavior modeling. These constructs are already built into the standard UML diagrams, and adding several stereotypes just to say which kind of NAF elements the behavior diagram elements stand for, was unnecessary when the models already contain these links. There was also a problem in many cases that these elements could not easily be dragged from the toolboxes, and the standard UML elements catered for more convenient ways of modeling. Finally, stereotyping some elements, like Ports, cluttered their visual presentation in the diagrams, e.g. by making their minimum size much larger than we wanted.

In addition to adapting the metamodel and UML profile, our implementation involved customizing analysis reports, XSLT transformations and scripts for exchange of model data with other tools. A particularly challenging issue was the need for extracting portions of the overall architecture database into smaller models for parallel development, e.g. by the suppliers of a given project. Again UML made things difficult. The roles of a relationship are modeled as properties of the elements that participate in the relationship. This means that adding a relationship involves altering the packages where both endpoints reside. When everything in the entire architecture is connected indirectly to everything else, this makes it difficult to modularize the architecture into sub-models that can be worked on independently. We finally arrived at a solution to this problem that involves strictly controlled use of the compare and merge functionality of the tool. This is coupled with a custom script that separates out a package from the rest of the architecture, putting all of the elements that the package refers to but does not own, in a separate Context package that the user has to handle the right way during the merging process.

4.2 Usage Experiences

The established architecture framework and development methodology has been in place a few months at the time of this writing. Major aspects of 5 development projects have been modeled, including project definitions with milestones and objectives, requirements, operational nodes and their information exchange, and systems with components, interfaces and standards. At the time of this writing, the architecture repository contains roughly 17K elements and 29K relationships.

It soon became evident that a generic architecture development methodology was insufficient to motivate usage, so a number of customized methodologies have been developed, for requirements management at the project and portfolio level, and for integrated solutions delivery across projects. These methodologies apply a small subset of TOGAF ADM tasks and NAF architecture views, extended with custom views and tasks.

Finally, the implementation has proven capable of insourcing a number of previously developed architecture descriptions, from national as well as international activities. This also includes models developed in other tools, like ARIS and ERWin, as well as business architecture models developed in a locally defined metamodel by the Norwegian Defense Research Establishment.

5 Conclusions and Further Work

As the experiences reported in this paper illustrate, there are fundamental differences between NAF and TOGAF that you should take into account when bringing the two together. The physical systems engineering perspective of NAF conflicts with the information systems approach of TOGAF, and the enterprise wide portfolio management scope of TOGAF does not always fit the acquisition project focus of NAF. These differences are natural consequences of the differences between military and business environments. In most businesses, hardware is a commodity, and most of the IT complexity lies in the application software. Consequently, this is the primary focus of TOGAF. Military hardware, on the other hand, is more custom made, with diverse and dynamic communication backbones. This implies that the cost, uncertainty, and complexity of the IT architecture to a much larger extent reside on the physical level. NAF consequently pays more attention to these aspects than TOGAF does.

Where TOGAF proposes an elaborate methodology and a simple content framework, NAF contains a simple methodology and an elaborate content framework. The two approaches are thus complementary. Ideas for simplifying the rather complex methodology of TOGAF or the content framework of NAF, can be derived from the simpler solutions chosen by the other standard. So far, this has mainly resulted in a simpler metamodel in our work, while the architecture development methodology to a greater extent has been adapted in order to fit with the local organizational practices and procedures. The work has also resulted in a number of change requests for NAF, put forward to NATO. As many stakeholders still see the metamodel as too complex, further simplification is ongoing.

Compared to previous work [2,12,13], our experiences shows that going all the way to implementation uncovers a lot more challenges than a high level conceptual analysis. Future practice-oriented enterprise modeling research should similarly focus on real world implementations to understand which differences actually make a difference when you compare modeling frameworks.

References

1. Department of Defense, Department of Defense Architecture Framework version 2.0 (2009)
2. Hi-Q Systems: TOGAF to MODAF mapping (2008)
3. Jørgensen, H.D., Ohren, O.P.: Achieving Enterprise Interoperability through the Model-based Architecture Framework for Enterprises. In: Enterprise modelling and ontologies for Interoperability Workshop (EMOI), CAiSE, Riga, Latvia (2004)
4. Jørgensen, H.D.: Enterprise modeling – What We Have Learned, and What We Have Not. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 3–7. Springer, Heidelberg (2009)
5. Ministry of Defence, Ministry of Defence Architecture Framework version 1.2.004 (2010)
6. NATO NC3 Board: Compendium of NNEC-Related Architectures (2006)
7. NATO NC3 Board: RFCP Regarding NAF v.3.1, Chapter 5, (March 1, 2010)

8. NATO NC3 Board: NATO Architecture Framework (NAF) v.3, appendix 1 to annex 1 to AC/322-D (2007) 0048 (2009)
9. OIO Architecture Guide, `http://ea.oio.dk/`
10. Object Management Group: Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF), v. 2.0 (2010)
11. The Open Group: TOGAF Version 9, standard (2009)
12. The Open Group: The Open Group Architecture Framework (TOGAF) and the US Department of Defense Architecture Framework (DoDAF), white paper (2006)
13. The Open Group: The Open Group Architecture Framework (TOGAF 9) and the US Department of Defense Architecture Framework (DoDAF), white paper, rev (2010)

Towards an Approach for Stakeholder-Oriented Elicitation and Identification of Concerns in EA*

Carsten Lucke and Ulrike Lechner

Professur für Wirtschaftsinformatik
Universität der Bundeswehr München
85577 Neubiberg, Germany
{carsten.lucke,ulrike.lechner}@unibw.de

Abstract. The concept of concern is used in Enterprise Architecture (EA) to express a stakeholder's area of interest in a system whose architecture is to be described. Many EA-related problems are rooted in weak stakeholder orientation. We propose an approach to explicitly model stakeholders' concerns as part of an architecture description. Our contribution is a modeling notation for concern elicitation and a method for concern identification. Our approach is based on goal-oriented requirements engineering and is compatible to the conceptual framework of the ISO 42010 international standard. We claim that our approach allows for a more thorough understanding of stakeholders' concerns and facilitates a stronger stakeholder orientation in EA.

Keywords: Goal-oriented requirements engineering, GORE, enterprise architecture, concern elicitation, concern identification, stakeholder orientation.

1 Introduction

Several predominant challenges in the field of enterprise architecture (EA) are related to stakeholders and caused due to a weak stakeholder orientation [1-4]. Stakeholder orientation means the careful consideration of EA stakeholders and their concerns, which is an important success factor for any enterprise architecting effort [1, 2, 5, 6]. In terms of an enterprise architecture description (EAD) a proper stakeholder orientation comprises identifying stakeholders and their architecture-related concerns (cf. [7]). Despite the availability of EA frameworks, notations, models or tools, alignment of EADs to stakeholders' requirements remains a problem in EA practice [1-3]. Determining stakeholders' architecture-related concerns is critical to identify suitable architecture viewpoints (cf. [7]).

We develop an approach (i.e., ASTEAM – Approach for STakeholder-oriented EA Modeling), which facilitates the determination of architecture viewpoints tailored to

* We thank Dr. Tobias Lehmann from Inopus GmbH and our colleague Sascha Krell for helpful comments and the opportunity of collaboration on this project. We gratefully acknowledge the fruitful collaboration of our workshop participants as well as our project partners.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 147–161, 2011.

stakeholders' requirements. ASTEAM comprises a modeling notation and a method to guide development. In this article we present these parts of ASTEAM that are targeted at eliciting stakeholder requirements and identifying their concerns. Our contribution is (1) a modeling notation to develop concern models and (2) a method to identify concerns based on these models. Particular to our approach is that we model requirements in a GORE-based fashion and derive stakeholders' concerns from these requirements. Structural patterns aid this derivation process. The design of ASTEAM is compatible to the ISO 42010 standard [7] and based on goal-oriented requirements modeling (GORM) and goal-oriented requirements engineering (GORE) [8-10].

We claim three advantages. First, our approach allows for a more thorough understanding of stakeholders' concerns. Second, requirements that form the basis of concerns can be directly elicited in interviews and meetings. This allows for a more tailored stakeholder orientation than featured by EA frameworks or concern list based approaches (cf. section 2). Third, we claim that more tailored concerns eventually lead to more tailored viewpoints taken in an architecture description and therefore to a better stakeholder orientation.

Section 2 covers theoretical foundations. Section 3 presents our research approach. In section 4 we propose a conceptual model integrating GORE and EA concepts as described by the ISO 42010 standard. We present sample models from an EA project, introducing our modeling notation for concern elicitation and the associated method guiding concern identification. Section 5 concludes.

2 Definitions and State of the Art

In this paper we conform to the definitions and concepts provided by the ISO 42010 international standard [7]. It defines architecture as the "fundamental conception of a system in its environment embodied in elements, their relationships to each other and to the environment, and principles guiding system design and evolution" [7]. The architecture of a system is captured in an architecture description (EAD) defined as a "collection of work products used to describe an architecture" [7]. The ISO 42010 standard provides a conceptual model of architecture description (cf. Fig. 1).

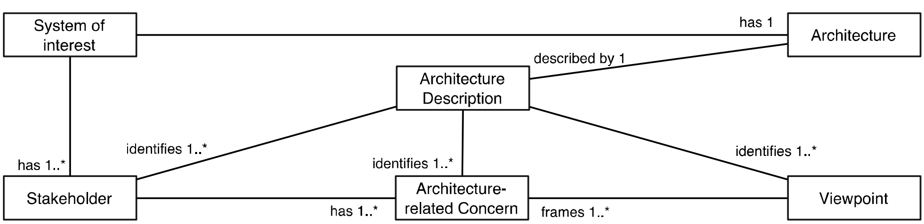


Fig. 1. Conceptual model of architecture description: content model (cf. [7])

This conceptual model captures that an architecture description shall identify stakeholders. We conform to the ISO 42010 standard's definition of a stakeholder as "individual, team, organization, or classes thereof, having concerns with respect to a system" [7]. According to ISO 42010, an *architecture-related concern* (cf. Fig. 1) is an "area of interest in a system pertaining to developmental, technological, business, operational, organizational, political, regulatory, social, or other influences important to one or more of its stakeholders" [7]. Within an architecture description the stakeholders' concerns are framed by one or more viewpoints. A *viewpoint* is a "work product establishing the conventions for the construction, interpretation and use of architecture views and associated architecture models" [7].

Identification of stakeholders and concerns is considered both in EA and requirements engineering (RE). Although the importance of requirements engineering for EA is acknowledged in a number of publications [2, 6, 11-15], we are aware of only a few approaches, which offer methodological guidance for a stakeholder-specific identification of concerns in connection with requirements engineering. EA frameworks (EAF) for instance are typically rather abstract (cf. [2, 16]). Since they are designed in regard to certain generic stakeholders and concerns they prescribe viewpoints applicable to this generic stakeholder/concern combination only.

We see the need for an approach that aids the enterprise architect in capturing stakeholders' requirements and information demands and in identifying architecture-related concerns based on these. There are a few approaches guiding stakeholder-oriented concern identification most of which use predefined concern lists to identify stakeholders' actual concerns.

The Enterprise Architecture Management Pattern Catalog (EAMPC) [17, 18] utilizes a best practice list of EAM concerns and amongst others identifies dependencies between these concerns, methodologies required to address these concerns (M-Patterns) and viewpoints, addressing these concerns (V-Patterns).

Another approach for *stakeholder-oriented modeling and analysis of EA* is developed at the HSG St. Gallen [1, 2]. It comprises a modeling and analysis framework, a viewpoint system and a development method. The method defines steps to identify stakeholders' concerns, elicit requirements, select viewpoints and define an information model. For concern identification this approach advocates the use of a list of potentially useful concerns as for instance provided by the EAMPC [2]. The concern identification step is followed by a requirements elicitation step to gather detailed requirements in regard to concerns identified relevant.

The Pedigreed Attribute eLicitation Method (PALM) [19, 20] is "a lightweight method based on goal oriented requirements engineering that begins with a canonical list of business goals and elicits specific business goals from the perspective of various stakeholders" [19]. "Outcome of the business goal elicitation method is a set of quality attribute requirements with a pedigree rooted in business goals" [20]. Business goals are interpreted in terms of quality attribute requirements in order to inform the definition of a system's software architecture. Lists of business goals as well as architecture-related quality attributes (e.g., ISO 9126) are used and discussed to identify and agree on relevant architecture quality attributes.

All aforementioned approaches are able to identify a large range of stakeholder concerns; their flexibility relies on the quality and extent of the concern- or quality attribute lists. The approaches offer a good opportunity to quickly identify common concerns (i.e., in the sense of best practices). Conversely, this means they have weaknesses to address specific or uncommon concerns. Concerns that are not captured in the concern or quality attribute lists will not be found as a result of RE activities.

3 Research Design

Our research design follows an iterative approach. A first concept of ASTEAM was developed based on theory and the proposition that typical GORE concepts hold for a reasonable representation of the notion of *concern*, making this concept easier to hypostatize [4]. To consider practitioner needs a workshop was held, resulting in a prototype version of ASTEAM. This prototype was then applied in a project to gain practical experience and reach a mature version of ASTEAM; the paper at hand presents the state of our ASTEAM approach after this project.

The practitioner workshop was held on the topic of stakeholder-orientation in EA modeling to discuss the early concept of ASTEAM. This half-day workshop took place in September 2010. Three EA researchers and five EA practitioners of companies and organizations operating in public business, government and the defense sector attended this workshop. The workshop had three topical parts, each of which was preluded with a short presentation followed by a round of discussions. These three parts were: (1) stakeholder-related issues and means to a stronger stakeholder orientation in EA; (2) the ASTEAM methodology and its integration into the typical enterprise architecting process; (3) the ASTEAM modeling notation, its model types and model elements. We discussed, which model elements the attendees expected, based on their experiences, to be part of EA models that aim at understanding stakeholders' requirements related to an EA effort. Discussions were tracked in a workshop protocol and criticism and suggestions were used to craft a revised version of the ASTEAM prototype.

The ASTEAM prototype as defined after the workshop was applied in an EA project for an industry partner (IP) operating in the aviation industry. The project went from September 2010 to January 2011. About 150 people are employed in our IP's department, which is part of a leading manufacturing and support company in the aviation industry. The main area of activity of our IP is software maintenance (SWM) of avionic software used in two different types of aircrafts with a lifetime period of thirty years "plus". The goal of the project was to develop a department-wide baseline EAD on the subject matter of review support for aircraft software (SW) to be maintained. The first author of this publication participated in the project as an architect.

We used the following sources of evidence to inform our concern modeling activities:

- *Stakeholder interviews*. Interviews were open-ended; question asking was semi-structured (tending to unstructured) to gain substantial insight [21]. Initial question asking was informed by contract and strategy documents.
- *Contract documents*. These documents give important information about project goals.
- *Strategy documents* are used as source to an understanding of IP's organizational goals, helping to understand the organizational context of the project.

Interviews were not recorded; contract and strategy documents are confidential and only accessible in the IP's intranet. Important information was therefore written down if possible and directly captured in an ASTEAM concern model.

4 Concern Elicitation and Identification

This section presents our concern elicitation modeling notation and concern identification method. The presentation of our modeling notation and method follows the order of their application in practice. The conceptual model underlying our concern model is presented in section 4.1 and the concern modeling notation in section 4.2. Our method comprises of three steps:

1. Stakeholder identification;
2. Concern elicitation (cf. section 4.3); and
3. Concern identification (cf. section 4.5).

A prerequisite to concern elicitation is identification of (key) stakeholders. ASTEAM makes no explicit specification for this step. We refer to stakeholder theory (e.g., [22]) or enterprise architecting methods (e.g., [6, 23]) for guidance regarding this task. How to conduct concern elicitation is described in section 4.3 and here we exemplify our approach with sample models from our practical project. Section 4.4 presents structural patterns of concern models; section 4.5 presents our method that guides the identification of concerns utilizing these structural patterns.

4.1 Conceptual Model

Our ASTEAM approach to improve stakeholder orientation in EA is to explicitly model stakeholders and their concerns. In our opinion concerns like *"functionality, performance, reliability, security, […], cost, schedule, quality of service [7]"* can hardly inform the definition of architecture viewpoints for an architecture description. We define a notation for the goal-oriented elicitation of stakeholders' requirements and the derivation of concerns from these requirements, facilitating a more precise understanding of stakeholders' concerns.

Fig. 2 illustrates our concepts of GORE-based concern modeling in conformance to the ISO 42010 conceptual model of architecture description.

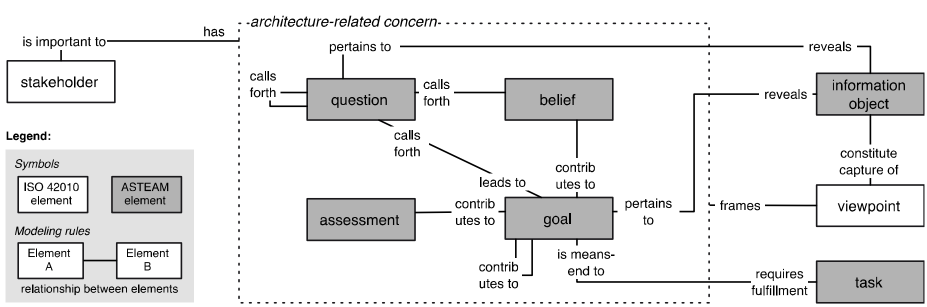


Fig. 2. ASTEAM conceptual model of GORE-based concern modeling in conformance to the ISO 42010 conceptual model of architecture description

A *stakeholder* [has] *architecture-related concerns*, which s/he [considers important]. Groupings of the elements of *goal, belief, question* and *assessment* make up the concept of architecture-related concern. This is indicated by the dotted line drawn around these four elements. Each of these four elements can be related to a stakeholder (i.e., [has / is important to]). Question, belief, assessment or another goal can be related to a goal. A common case is one goal [contributing to] another goal. An assessment describes a current situation, which positively or negatively [contributes to] (i.e., influences) a stakeholder's goal. A belief can be related to a goal in so far, that it indicates expectance of a positive or negative [contribution] (i.e., influence) to this goal. A question helps to formulate new goals (i.e., [leads to]) and supports goal refinement (cf. [24]). Moreover a question provides rationale for the existence of a goal. A question can be evoked (i.e., [called forth]) by a belief, a goal or another question. The *task* element is directly connected to the goal element. This is a [means-end] relationship. A task is required to be fulfilled in order to accomplish a stakeholder's goal. We do not consider the task element to be inside the concern boundary because we believe a stakeholder will not care about, who accomplishes a goal or how it is accomplished. By identifying stakeholders' concerns we are able to [reveal] their information demands in the form of *information objects*, which are relevant to them. According to our experiences, especially the question and goal elements are helpful in that regard. Therefore, these elements are directly connected to information object. Information objects are identified based on stakeholders' concerns, represented by their goals and questions. To actually develop an EAD, *viewpoints* are chosen, that [frame] the stakeholders' concerns.

Subsequently we explain the elements that are used in the ASTEAM concern model to render the concept of architecture-related concern more precise.

Goal. This is the central aspect in GORE and present in all goal-oriented modeling approaches. "A goal is an objective the composite system should meet" [25].

Belief. This element is inspired by the aforementioned practitioner workshop and the i* modeling notation [26, 27]. It is best described with the following quote: "A belief is a condition about the world that the actor holds to be true" [28].

Question. The *question* element is inspired by discussions led at the practitioner workshop. We aim to capture stakeholders' information demands with the question element, which tend to be expressed in the form of questions. We observed two things about questions: First, they can often be traced to a goal and second, they often lead to refined goals. This observation is backed by the Goal/Question/Metric method [24]. Thus, we find questions to provide valuable rationale for goal refinement.

Assessment. The Merriam Webster online dictionary defines an assessment as "the action or an instance of assessing: appraisal" [29]. Appraisal is defined as "a valuation of property by the estimate of an authorized person" [30].

Task. The *task* element is inspired by i* [27, 28], GRL [31] and KAOS [32]. The Merriam Webster online dictionary defines a task as "a usually assigned piece of work often to be finished within a certain time" [33].

Information Object. Information objects are actual domain elements captured in different models defined by architecture viewpoints. Information objects are important in regard to concern elicitation since they are the actual, real-life objects of stakeholder interest that are to be captured in an EAD [34].

We would like to explain how we developed our notation. GORE has been identified as a possible means to reach an improved stakeholder orientation in EA by facilitating a more precise understanding of stakeholders' concerns (cf. [4]). We design ASTEAM to be compatible with the ISO 42010 standard representing typical EA concepts and integrate common GORE concepts due to our proposition that these concepts hold for a reasonable representation of the notion of *concern.*

The concepts used in our notation originate from three streams of literature: (1) The ISO 42010 and related EA literature. We consider the standard itself as well as literature discussing it, which we found with a Google search for the terms *"enterprise architecture iso 42010"* and *"enterprise architecture ieee 1471"*. (2) Common and widely cited GORE frameworks and approaches: i* [27, 28], GRL [31] and KAOS [32]. (3) Articles discussing the adoption of GORE in EA. We searched for such articles using various combinations of "GORE" and "EA" in abbreviated and non-abbreviated form using Google Scholar, IEEE Xplore and AIS Electronic Library. Few scholarly publications exist about leveraging GORE and GORM in enterprise architecture [15, 35]. Our search yielded one modeling approach: ARMOR [15], a goal-oriented requirements modeling language for enterprise architecture.

Our conceptual model contains these elements we identified in our literature analysis which are as well considered relevant by participants of our aforementioned workshop. Table 1 illustrates the respective concepts and their origins.

Table 1. GORE and EA concepts and their origin

Concept	i*	GRL	KAOS	ARMOR	ISO 42010 [7, 36]	Software cartography [34]
Assessment	n	n	n	y	n	n
Belief	y	y	n	n	n	n
Concern	n	y	n	y	y	y
Goal	y	y	y	y	n	n
Information Object	n	n	y	n	n	y
Question	n	n	n	n	n	y
Stakeholder	y	y	y	y	y	y
Task	y	y	y	n	n	n
Viewpoint	n	n	n	y	y	y

"y" = concept exists in approach, "n" = concept does not exist in approach

4.2 Modeling Notation for Concern Elicitation

This section introduces the ASTEAM concern modeling notation. It is based on the i* visual syntax (cf. [26, 28]) because i* is a well-accepted GORM notation that allows to take goals of different stakeholders into account. We keep elements and symbols already present in i* and add elements for *question, assessment* and *information object*. Fig. 3 depicts the element symbols defined for our concern modeling notation.

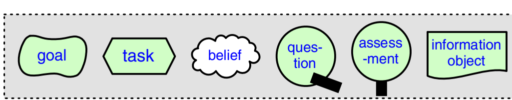


Fig. 3. Element-symbols used in an ASTEAM concern model

The modeling rules of our modeling notation used in the concern models is illustrated in Fig. 4. The relationships of *goal contribution, task decomposition* and *means-end* exist in the original i* notation [26, 28]. According to our conceptual model we add the following relationships: *assessment contribution, question decomposition* and *calls-forth question*. Elements and relationships we propose to elicit concerns have been explained on the basis of our conceptual model in section 4.1.

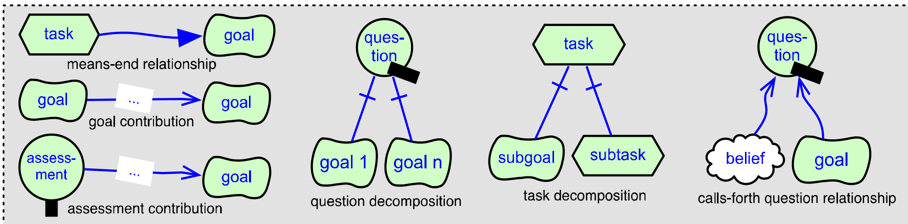


Fig. 4. Modeling rules for an ASTEAM concern model

4.3 Concern Elicitation

Aim of the concern elicitation step is to collect and capture stakeholders' requirements in an ASTEAM concern model in a GORE-based style. We assume that stakeholders have been identified prior to concern elicitation.

This phase identifies stakeholders' goals, questions, beliefs, assessments and tasks and the relations between them. Activities to collect these requirements are for instance workshops, interviews or document reviews. These requirements are captured in models as described in section 4.1 and 4.2. For each requirement the connection to the issuing stakeholder needs to be documented. The ASTEAM approach extends typical requirements engineering activities conducted in an EA undertaking.

We exemplify concern elicitation with sample models from our practice project. Our sources of requirements were interviews, strategy and contract documents.

The context for our sample models is as follows. Our IP's department is responsible for the "transfer" of aircraft computer units' software and the development environments required to build these software components. A number of computer units (LRU; line replaceable unit) are operated in both types of aircrafts for different purposes, making up the avionic system. Many different manufacturers produce these LRUs and their software. "Transfer" means our client has to be able to reproduce software builds, originally created by other manufacturers. These software builds have to be identical to the originally built software (SW). To ensure good software quality and meet strict aviation industry standards in software development, the IP's department has decided conduct an EA project describing the department's baseline architecture to:

- Investigate review processes and the state of documentation guiding review activities;
- Inform the harmonization of processes as well as process documentation in case necessary; and

- Evaluate the options for information system support for review accomplishment and evaluation.

We consider two goals in our example – i.e., *Comprehensive taxonomy of "Reviews"* and *Clearly defined review actions.* These goals are part of the overall objective to understand what reviews are conducted by teams of the department and how these reviews are conducted. The first author led and analyzed interviews as well as strategy and contract documents to gather requirements as done in traditional RE. Gathered requirements were analyzed for goals, questions, beliefs, assessments and tasks. Partly, requirements were rephrased in this process to exhibit a model-handleable form. In Fig. 5 we show the exemplary model capturing the two goals *Comprehensive taxonomy of "Reviews"* and *Clearly defined review actions.*

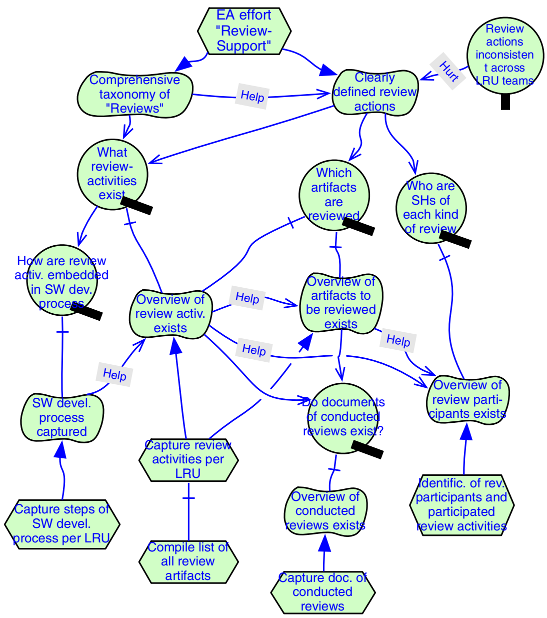


Fig. 5. Sample of the project's concern model

The customer wanted us to discover which teams conduct what types of reviews and to clearly define these review actions. These objectives were named in interviews as well as the contract document. Both goals are in a means-end relationship with the task *EA effort "Review Support"* because they represent main requirements of the IP. Since both goals are rather imprecise, they are further refined. The goal *Comprehensive taxonomy of "Reviews"* leads to a question (i.e., *What review activities exist*). This question calls forth another question (i.e., *How are review activities embedded in SW dev. Process*). Decomposition of these two questions yields the more precise goals *SW development process captured* and *Overview of review activities exists*. Further on, means-end tasks are captured, which shall be performed to achieve the respective goals. The refinement of the goal *Clearly defined review actions* works in a similar way and is not discussed in detail for the sake of brevity.

Up to now the model is mostly identical to what a traditional goal model would yield – apart from some additional concepts being captured (e.g., question elements providing additional rationale for goal refinement). We find that this model captures motivation and vision of the EA undertaking much explicit than often done in EA.

For each model element we maintain meta information, most importantly associated stakeholders. This way we can trace requirements back the respective stakeholders allowing us to identify who has which goals or questions; or who is considered responsible for a certain task. Capturing this information in the form of a stakeholder element directly in the model would make the models more difficult to read and understand; maintainability would be impaired as well. To capture structured attribute information about goals and beliefs we use business goal scenarios as described by Clements and Bass [19, 20].

Once stakeholder requirements are captured in a concern model it forms the foundation for the next step – i.e. concern identification.

4.4 Structural Patterns for Concern Identification

We define a model-structure-based concern identification method, which offers guidance to an otherwise merely subjective concern identification process. We propose structural patterns guiding concern identification for every possible element link (e.g., contribution, decomposition, etc.). These patterns are a result of our concern identification experiences in the project as well as based on common sense considering the link semantics. Three exemplary patterns are presented in Fig. 6. The direction of traversal defines in which direction concern identification is conducted.

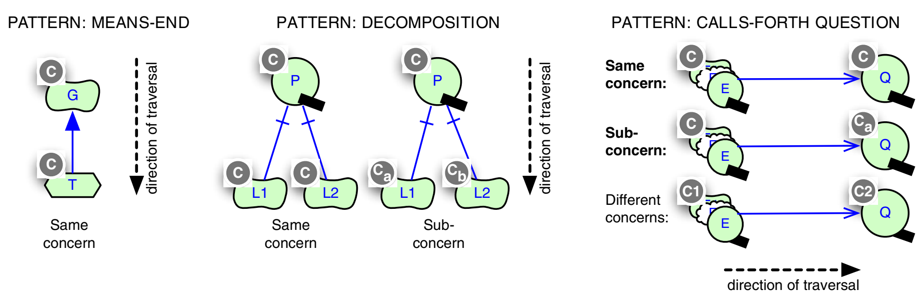


Fig. 6. Structural patterns guiding concern identification

The *means-end pattern* represents a conditional statement: If task T is means-end to goal G, then T is of the same concern as G. This means if G is considered to belong to a certain concern, T belongs to that concern as well.

The *decomposition pattern*, which is relevant for tasks and questions, indicates that an element's sub-elements are of the same concern or of a sub-concern (cf. [37] on aggregation of concerns): If composition-parent P is of concern C, composition leafs L1-Ln are most likely of concern C as well. Depending on individual judgment, L1-Ln might belong to sub-concerns of C (i.e., C_a, C_b, etc.). Using this pattern concern identification still depends on the discretion of the architect but it provides guidance by determining that L1-Ln to not belong to a completely different concern.

The *calls-forth question* pattern is relevant for all element combinations that describe the forth calling of a question (i.e. goal, belief, question). Based on a goal, belief or question another question is formulated which represents a stakeholder's demand for more information. From our experience the pattern indicates the identification of the same or a sub-concern for the questions being called forth. However, both elements (i.e., the source element E and the question Q being called forth) might as well be of different concerns if the architect's rationale suggests so.

4.5 Concern Identification

Aim of the concern identification step is to identify architecture-related concerns based on the requirements captured in an ASTEAM concern model. We define a method for concern identification comprising the following steps:

1. Determination of concern-belonging for the concern model elements. This is conducted in a depth-first search (DFS) manner combined with use of our patterns introduced in section 4.4. Concern-ids are assigned to model elements, where different numbers represent different concerns. Elements with multiple concern-ids assigned are taken into account for the generalization of multiple concerns.
2. Highlighting of concerns (i.e., groupings of requirements elements).
3. Phrasing of concerns – i.e., giving the concern groups a name.
4. Analysis of inter-concern dependencies.

As a result every concern model element is assigned at least one concern.

We continue our example (cf. section 4.3) by conducting the aforementioned steps resulting in the model depicted in Fig. 7.

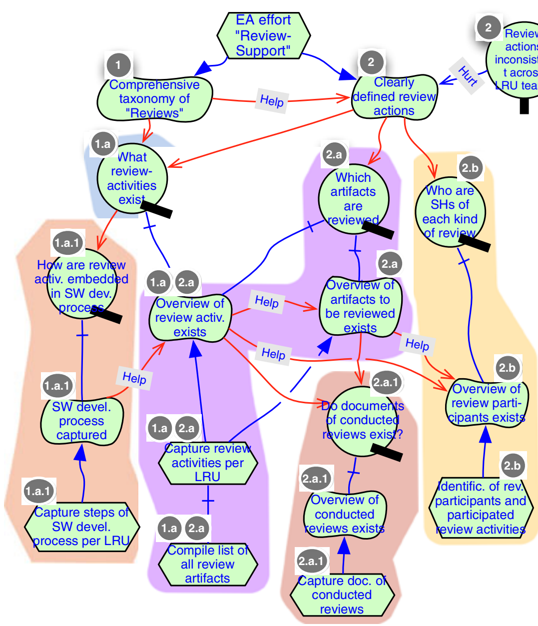


Fig. 7. Concern model after concern identification is completed

We start to assign concern-ids with the top-left-most element, connected to the task that represents our EA effort. Thus, we start with the goal element *Comprehensive taxonomy of "Reviews"*, which is assigned a (1). The calls-forth pattern is used to assign a concern-id to the connected question *What review activities exist*. We consider it a sub-concern and thus assign the id (1.a). The contribution relationship to goal *Clearly defined review actions* is not used for concern identification, since this relationship is not considered helpful to guide concern identification; the goal is not assigned a concern-id at this point. For concern identification of the question's (i.e., *What review-activities exist*) related elements we use the *calls-forth* and *decomposition* pattern. We consider the question *How are review activities embedded in SW development process* representing a sub-concern (1.a.1); asking about how review activities are anchored in the whole of activities conducted during SW development. It is therefore a more specific aspect of the original concern regarding review activities in general. The question is further decomposed to the goal *Overview of review activities exists*. We decide not to annotate a sub-concern but keep the concern (1.a). We use the decomposition and means-end patterns for concern identification for the remaining elements of this part of the graph. Once the last element in a part of the graph is reached, concern identification continues with the element at the top-left-most position, which has not yet been assigned a concern-id; DFS will inform the decision which element to take. In most cases a new concern-id will be assigned to this element because top-level goals are often rather distinct and thus belong to different concerns. In our case it is the goal element *Clearly defined review actions*, which we assign concern-id (2). Further pattern-based decomposition leads to the resulting concerns shown in Fig. 7.

We group concerns with equal concern-ids by drawing colored boundaries around them. The colors are used to distinguish different concerns and have no special meaning. Table 2 summarizes the concerns we identify based on the underlying concern model. Concern names are chosen by discretion of the architect. Our conceptual model and the chosen concern subjects represent a concern understood as a stakeholder's area of interest in a canonical, goal-oriented and practitioner-friendly way. We call it practitioner-friendly because the elements making up the concept of concern can easily be perceived and captured in interviews or meetings with relevant stakeholders. Note that we do not try to provide an answer to the question what a concern is in an epistemological sense.

Table 2. Concerns identified on the basis of the concern model

Id	Concern
1	Taxonomy of "reviews"
1.a	Review activities
1.a.1	Anchoring of review activities in SWM process
2	Review actions
2.a	Review-relevant LRU artifacts
2.a.1	Conducted reviews overview
2.b	Review participants

We identify inter-concern dependencies, which are colored red in Fig. 7. Inter-concern dependencies are helping viewpoint identification in a later step of the ASTEAM method. The relationships between concern model elements help to determine relationships between different concerns. We utilize *goal contribution* and *calls-forth* links to investigate inter concern relationships, since these links connect model elements belonging to different concerns. For instance a goal belonging to concern A contributing to a goal belonging to concern B indicates an inter-concern relationship between these two concerns. Another indication of two or more concerns being in a relationship with each other is one model element being assigned two or more concern-ids. This is the case with three elements in the model displayed in Fig. 7.

The immediate benefit of the concern model after the conduct of our method is the stakeholder-oriented identification of concerns and a thorough understanding these concerns. Moreover the inter-concern dependencies help to inform the selection and/or definition of architecture viewpoints. A convenient by-product of the model is the identification of a project's work packages – in case they have not been defined at this stage of the project; the task elements of the concern model resemble a large part of the work packages defined for our project.

5 Conclusions and Discussion

The concept of architecture-related concern is rather dim and hard to observe or investigate in practice. Although EA frameworks and procedure models acknowledge the importance of RE, they hardly consider method support specifically guiding elicitation and identification of stakeholders' concerns. Typically, it also remains unclear how knowledge about concerns should be captured and documented.

In the article at hand we present the concern elicitation and identification part of our ASTEAM approach for stakeholder orientation in EA modeling. We propose a conceptual model, a modeling notation and a method to guide and facilitate a more precise elicitation and identification of EA stakeholders' concerns.

Our goal is to achieve a stronger stakeholder orientation in EA. Can we achieve that with our concern modeling approach ASTEAM? Our initial application in a practical EA project shows promising results. Concerns are captured and derived from stakeholders' requirements allowing for traceability between stakeholders, requirements and concerns. When faced with communication of architectural knowledge related to a certain concern it is immediately clear which stakeholder(s) to address. Adopted in the practical project our approach yielded reasonable concerns and at the same time facilitated a precise understanding of these concerns and the inter-concern dependencies. This thorough understanding of concerns facilitated the determination of stakeholder-appropriate architecture viewpoints. Moreover our models provide valuable information about the vision for an EA undertaking – lack of such is named as a frequent issue in EA [3, 15, 35]. For these reasons we consider an ASTEAM concern model a worthwhile addition to an enterprise architecture description.

The additional workload caused by the adoption of ASTEAM was reasonable and in our opinion well worth the effort. ASTEAM is compatible to the ISO 42010 standard [7], which is well accepted by EA practitioners and research alike. It can be combined with existing EA frameworks without any problems or changes to the framework. We want to emphasize that argumentation, exemplification and results of this work are fitted for the enterprise architecture domain. As ISO/IEC 42010 [7] holds for all software intensive systems, we assume that a generalization to the field of system architecture is feasible and sensible.

The results of the adoption of ASTEAM look promising so far. We intend to obtain further empirical validation to support these first insights and eventually enhance ASTEAM.

References

1. Aier, S., Kurpjuweit, S., Riege, C., Saat, J.: Stakeholderorientierte Dokumentation und Analyse der Unternehmensarchitektur. In: Hegering, H., Lehmann, A., Ohlbach, H., Scheideler, C. (eds.) INFORMATIK 2008: Beherrschbare Systeme – dank Informatik, vol. 2, pp. 559–565. GI/Köllen, München (2008)
2. Kurpjuweit, S.: Stakeholder-orientierte Modellierung und Analyse der Unternehmensarchitektur unter besonderer Berücksichtigung der Geschäfts- und IT-Architektur. Inst. für Wirtschaftsinformatik, doctoral thesis, Universität St. Gallen, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG), St. Gallen, p. 419 (2009)
3. Lucke, C., Krell, S., Lechner, U.: Critical Issues in Enterprise Architecting – A Literature Review. In:16th Americas Conference on Information Systems, Paper 305, Lima, Peru (2010)
4. Lucke, C., Lechner, U.: Goal-oriented requirements modeling as a means to address stakeholder-related issues in EA. In: Bernstein, A., Schwabe, G. (eds.) 10 Internationale Tagung Wirtschaftsinformatik, Zürich, Switzerland, pp. 714–723 (2011)
5. Niemi, E.: Enterprise Architecture Stakeholders - a Holistic View. In: 13th Americas Conference on Information Systems, Paper 41 (2007)
6. The Open Group: TOGAF Version 9 - The Open Group Architecture Framework (TOGAF). The Open Group (2009)
7. International Organization for Standardization: ISO/IEC 42010 (WD4). Systems and Software Engineering — Architecture Description (2007)
8. Cheng, B.H.C., Atlee, J.M.: Research Directions in Requirements Engineering. In: 2007 Future of Software Engineering, pp. 285–303. IEEE Computer Society, Los Alamitos (2007)
9. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Conference on The Future of Software Engineering, pp. 35–46. ACM, Limerick (2000)
10. Lamsweerde, A.v.: Goal-Oriented Requirements Engineering: A Guided Tour. In: 5th IEEE International Conference on Requirements Engineering, pp. 249–262 (2001)
11. Bender, G.: Designing a Stakeholder-Specific Enterprise Architecture Management based on Patterns. sebis, master's thesis, p. 136. TU München (2009)
12. Chief Information Officer Council: A Practical Guide to Federal Enterprise Architecture (2001)
13. Lankhorst, M., Quartel, D.: Architecture-Based IT Valuation - Supporting portfolio management and investment decisions. Via Nova Architectura (2010)
14. Op't Land, M., Proper, E., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture: Creating Value by Informed Governance. Springer, Berlin (2009)
15. Quartel, D., Engelsman, W., Jonkers, H., van Sinderen, M.: A Goal-Oriented Requirements Modelling Language for Enterprise Architecture. In: IEEE International Enterprise Distributed Object Computing Conference, pp. 3–13 (2009)
16. Schekkerman, J.: How to survive in the jungle of enterprise architecture frameworks: creating or choosing an Enterprise Architecture Framework. Trafford Publishing (2004)
17. Buckl, S., Ernst, A., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical Report TB0801, Chair for Informatics 19 (sebis), Technische Universität München (2008)

18. Chair for Informatics 19 (sebis). EAM Pattern Catalog, http://wwwmatthes.in.tum.de/wikis/eam-pattern-catalog/home (accessed June 6, 2011)
19. Clements, P., Bass, L.: Using Business Goals to Inform a Software Architecture. In: 18th IEEE International Requirements Engineering Conference (RE), pp. 69–78 (2010)
20. Clements, P., McGregor, J.D., Bass, L.: Eliciting and capturing business goals to inform a product line's business case and architecture. In: Bosch, J., Lee, J. (eds.) SPLC 2010. LNCS, vol. 6287, pp. 393–405. Springer, Heidelberg (2010)
21. Bortz, J., Döring, N.: Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler. Springer, Berlin (2006)
22. Mitchell, R.K., Agle, B.R., Wood, D.J.: Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. The Academy of Management Review 22, 853–886 (1997)
23. Spewak, S., Hill, S.C.: Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. QED Publishing Group (1993)
24. Solingen, R.v., Berghout, E.: The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. McGraw-Hill, New York (1999)
25. van Lamsweerde, A.: Handling Obstacles in Goal-Oriented Requirements Engineering. IEEE Transactions on Software Engineering 26, 978–1005 (2000)
26. Yu, E.: Modelling Strategic Relationships for Process Reengineering. Graduate Department of Computer Science. Doctor of Philosophy, p. 178. University of Toronto (1995)
27. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: 3rd IEEE Int'l Symposium on Requirements Engineering, pp. 226–235 (1997)
28. i* Wiki, http://istar.rwth-aachen.de (accessed August 7, 2010)
29. Free Merriam-Webster Dictionary. Assessment, http://www.merriam-webster.com/dictionary/assessment (accessed February 9, 2011)
30. Free Merriam-Webster Dictionary. Appraisal, http://www.merriam-webster.com/dictionary/appraisal (accessed February 9, 2011)
31. International Telecommunication Union: Z.151: User requirements notation (URN) - Language definition (2008)
32. Lamsweerde, A. v., Letier, E.: From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. In: Wirsing, M., Knapp, A., Balsamo, S. (eds.) RISSEF 2002. LNCS, vol. 2941, pp. 325–340. Springer, Heidelberg (2004)
33. Free Merriam-Webster Dictionary. Task, http://www.merriam-webster.com/dictionary/task (accessed February 9, 2011)
34. Lankes, J., Matthes, F., Wittenburg, A.: Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000. Software Engineering, 43–54 (2005)
35. Yu, E., Strohmaier, M., Xiaoxue, D.: Exploring Intentional Modeling and Analysis for Enterprise Architecture. In: 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (2006)
36. Emery, D., Hilliard, R.: Every Architecture Description Needs a Framework: Expressing Architecture Frameworks Using ISO/IEC 42010. In: European Conference on Software Architecture, pp. 31–40 (2009)
37. Buckl, S., Krell, S., Schweda, C.M.: A formal approach to architectural descriptions – refining the ISO standard 42010. In: Albani, A., Dietz, J.L.G. (eds.) CIAO! 2010. LNBIP, vol. 49, pp. 77–91. Springer, Heidelberg (2010)

Capture the Current State

Gregory Uppington

YouSoGetMe Pty Ltd

The paper describes the evolution of an enterprise engineering / architecture approach and solution through three case studies, emphasising the contributions of lessons learnt from each case study to this evolution, each presented with results used as input for refinement. The decade of experience with what could very much be dubbed 'enterprise modelling', the extent of the effort combining frameworks and methods to model and capture the current state of the enterprise, and the apparent maturity (if only in practical terms) of the 'solution' is well described and it is easy to follow the advancement throughout the entire process. Along the way the author has ran into a number of known, quite formidable challenges in the field of business analysis and enterprise engineering. The case studies focus on practical orientation and highlights lessons learnt for each step. The paper results in an interesting and detailed conclusion.

1 Introduction

Corporations, institutions and others competing in the global economic and trade environments require organisational efficiency to survive. Understanding of the enterprise is therefore essential for management to connect and combine people, processes, systems, and technologies to ensure that the right people and the right processes have the right information and the right resources at the right time. While there are many ways to deliver this understanding, this paper describes how a web-based solution called 'the Company Bookshelf' combines enterprise engineering and architecture (**EE/A**) with business analysis to address some of these challenges. From the model first developed in 1998 alongside the Generic Enterprise Reference Architecture and Methodology (**GERAM**), the Bookshelf has not been implemented in any organisation yet as the time has been spent profiling organisations, hospitals and government departments, gauging how the solution meets their requirements, assessing how the solution solves common organisational problems, and seeing how it compares to other systems. The approach, frameworks and the Bookshelf have been re-thought and refined through many scenarios in local, national and multi-national organisations, both real and virtual, however this paper presents only three of the case studies. The Bookshelf has its foundation in GERAM, and utilises the Generic Reference Model (**GRM**) - a model based on the Federal Reference Architecture (**FEA**) Reference Models (**RM**). The author believes automating the GRM through a web-based system creates an EE/A solution suitable for the majority of organisational forms, particularly government institutions. Whilst the aim of the research in 1998 was to provide a design-model for multi-national corporations (**MNCs**) involved in

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 162–177, 2011.

a Keiretsu, the Bookshelf has since evolved into a solution that captures and shows the current state of an enterprise (what each part of the company does and uses). Of more significance to the author are the challenges that had to be overcome during the course of its evolution:

– How to represent and capture the enterprise: the practical significance.
– How to use what was captured: the transferability of results.

The author believes the approach used to overcome these challenges have evolved enough to become a suitable and mature solution intended to benchmark industry best practice, and make the solution suitable for most organisational forms.

2 The Results: Using What Was Captured

The second challenge (how to use what was captured) is provided first as the screens show how the frameworks are held within the solution. The Business Screen in figure 1 shows the business view of *'using what was captured'*.

2.1 The Business Information

Whilst there is nothing remarkable about this screen, it shows the organisational hierarchy. The authorised user enters the different descriptors of each level, and names their business entities at each level. On the Business Screen of the Bookshelf, the user selects the Architecture Office in circle 1.

Fig. 1. The Business Screen

Each business area on each level usually has one management position, and two purposes:

– Purpose A: to provide its products and services,
– Purpose B: to manage itself, so it can achieve purpose A (Bernus, 1997).

The functions (shown on the tabs of circle 3) are one of two types:

- The 'product and service' (PS) business-functions (2a), or
- The 'management and control' (MC) business-functions (2b).

The tabs in circle 3 are the names of the business functions, provided by the business area selected in circle 1 ('Architecture Assessments' function). When the user selects a function, they are able to view its' processes, and tasks (circle 4). The user views (circle 5) documentation, templates, artefacts and information deemed important for the task by the actors responsible for that part of the business, within its' context, shown in figure 2. These results are useful when the organisation gives authorised access to their areas of responsibility, then new executives, managers and employees coming into the organisation could become functional earlier and contribute to its revenue much faster. This practice (called on-boarding) significantly increases productivity rates and throughput, decreases coordination costs and employee anxiety by permitting employees to identify with their new employer and understanding the company's values and priorities. The Profile Screen in figure 3 shows how the Bookshelf 'uses what was captured' in a different manner, showing an 'EE/A profile view' for a selected part of the business.

Fig. 2. Resources of the Task

2.2 The Profile Screen

The solution extends the Business, Information, Application, Technology (**BIAT**) view from the Australian Government Architecture (**AGA**) to show a profile of a business area, its people, processes, technologies, and (hard and soft) information to management and executive stakeholders. This BIAT is a commonly used perspective in government departments in Australia and the AGA forms the foundation for many key initiatives such as e-health and enterprise transformation. The AGA is based on the FEA. In the Profile screen the user selects a part of the business or a function, process or task in circle 1, and then selects what they want to see about it in circle 2. The example in figure 3 shows the details of the information systems used by the Architecture Office. The user has selected to see current-state details about the integration overview document, the user devices, the infrastructure pattern, and the non-functional requirements of the SOE, in the context of the Architecture Office. Whilst this seems quite straightforward, this lets the user view any item of description of any architectural element about any application used by the selected part of the business, and about any part of the business. The author applies the BIAT domains where each domain is subdivided and categorised to hold relevant information. The objective of the Bookshelf is to allow organisations to capture the important details

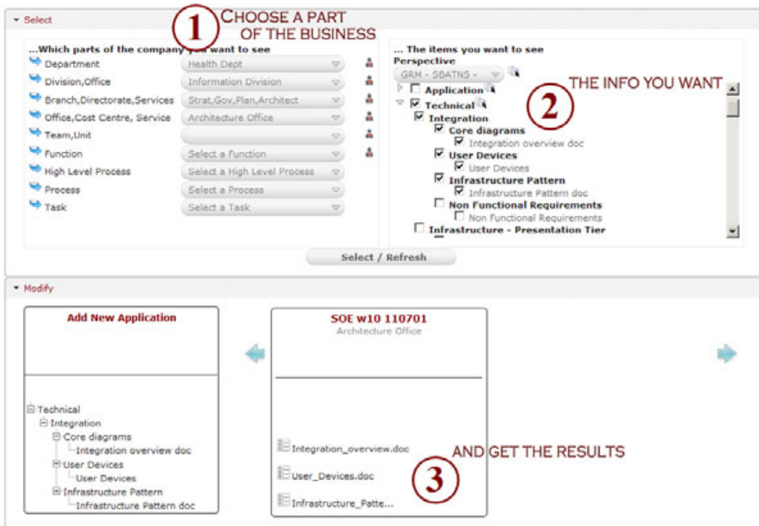


Fig. 3. The Profile Screen

of their enterprise from the organisational structure down to specific processes while showing the interconnections between the components of the enterprise. This baseline profile of the enterprise is a current-state snapshot showing the work done in each area of the business and what resources (documents, info, roles, systems, technologies, networks, etc.) are required. The aim is to use one system that knows the current state of the enterprise information, so you can capture it once and use it often in other systems. This profile perspective was created during the first case study, and addresses the challenge of how to capture and represent the enterprise.

3 The Background Experience

The approach used to capture the current state of the enterprise stems from the authors related work in the Globeman 21 (**GM21**) project. The GM21, started in 1994, was a part of the Japanese **IMS** (Intelligent Management Systems) program till mid 1999 (Bernus 1999). Participants were:

- Twenty industrial partners including Toyo Engineering Corporation (**TEC**), Hawker de Havilland, Toyota, Mazda, Omron, NCR Norge, Mitsui Engineering & Shipbuilding, and Newport News Shipbuilding;
- Eight research institutions including Fraunhofer Institute, CSIRO Australia, and Japan Institute of Industrial Science; and
- Six universities including Griffith University (Australia), University of Toronto, University of Tokyo, and University of Virginia and Helsinki University.

The GM21 was an industry-led project for product life-cycle driven plant engineering and demonstrated the application of life-cycle architectures to large scale enterprise engineering for virtual enterprises (VE)(Bernus 1999). Together with others, the author created the model showing the design of all organisations involved in a multinational Japanese Keiretsu.

The model identified each relevant enterprise business entity (**EBE**) of each organisation involved in the Keiretsu, the key part each EBE played, their relationships and feedback loops with other EBEs, and the elements within each phase of each EBEs design. This was called the **VRIDGE** model (Virtual and Real Information Technologies Driven Global Engineering) and gave a logical place-holder for any activity, document, artefact or information from any phase of architecture, design or operation for each business entity (Bernus et al, 1997).

4 Early Aims of the Solution

From engagements to provide EE/A understanding of enterprises between 1998 and 2003, the author found nearly every project began with analysis of the current state. Unfortunately after a number of engagements, the author had difficulty remembering precisely which document had the right information about a selected part of the business, where it was, and its status for each project. The author was looking for a system that showed the current state of the enterprise, where the information was always current. To solve this issue, in 2003 the author created the first version of the system which held the VRIDGE model. The system provided empty electronic placeholders for each phase of the design of the business. This version helped capture, organise and manage the complex information for a large-scale business transformation for a telecommunications company, but it had a number of issues. As part of the analysis, the author found many applications and systems that could show one or more representations of the organisation in its own way (Uppington, 2011), but the only 'system' that could be used enterprise-wide, and hold all different descriptions of the enterprise was the classic 'one-large-folder' on a corporate drive, usually subdivided by division and then by project or business area or both, and hardly ever life-cycle managed. In 2005 the author designed a mockup solution to capture and present the current state description of each business area of an organisation in its correct context, using the language of the business, to accept information and artefacts created by any other system, feed information to other systems, be non-proprietary, affordable, and keep the information current and applicable, and easy enough to use. Product comparisons at the time let the author discover there was no solution that satisfied the above requirements, let alone be affordable and understandable for a cost centre director and their employees to use easily; and decided to build one. During engagements for MNCs, the author used GERAM to provide the life-cycle phases of enterprise design, showing how the phase outputs fit into a hierarchy of the business area. Each enterprise design phase produces documents describing the relevant parts of the business, and the management, strategies, business processes, information, applications and technologies for each business area. Figure 4 shows the outputs of each life-cycle phase for any entity. The information within these outputs usually describes the resulting EBE, and belongs to a single domain of description. For example, the Mission Statement of an entity is a document that belongs to the Strategic Domain of that entity. This hierarchy is noted as the domains of the Generic Reference Model (GRM), used to overcome the first challenge: how to capture and represent the enterprise, and was identified during the first case study.

GERA Life-Cycle Phase	GERA Life-Cycle Phase Outputs	Enterprise Business Entities	Domains of the Generic Reference Model
Identification of the EBE	Strategic EBE details, PS & MC details	Highest level EBE, customers, Suppliers, buyers, ...	Strategic
Concept Phase	Enterprise MVV, Policies; identification of sub-EBEs PS& MC details	Branch (1st subdivision)	Strategic
Definition Phase	Strategic details of each EBE, PS & MC details	Department (2nd), Division (3rd) Cost Centre (4th),Program (5th)	Strategic
Design Phase: Functional Design	Details of functions of each EBE, PS & MC details	Each Program EBE	Business
Design Phase: Detailed Design	Decomposition of Functions to Tasks, resources, roles, artefacts, etc of each EBE	Each Program EBE	Business, Business-info, Application
Construction and Installation Phase	Business and Technical Architecture, change mgt artefacts, etc of each EBE.	Each Program EBE	Business, Business-info, Application, Application-info, Technical, Network
Operations Phase	Business and Technical Operational detail of each EBE	All EBEs	Business, Business-info, Application, Application-info, Technical, Network
Renewal, recycle or Disposal Phase	Business and Technical Change management details, projects	Any EBE	Change Management of every domain

Fig. 4. Influences on the Generic Reference Model

5 Practical Significance 1: Represent the Enterprise

Case study one was the catalyst for defining an approach and a perspective that could profile each part of the business as an entity, using a generic model in a consistent and measurable manner.

5.1 Case Study One: Healthcare Enterprise A

The organisation involved in the first case study was a government healthcare department. Its structure followed the shared-model concept where some ICT support is provided by one division for most other divisions. The hierarchy of the department had four levels of management from the CEO (highest management level) to the director of each cost centre (second management level). The sub-hierarchies of each cost centre differed with some having extra levels of management between a director and employees. The enterprise architecture (**EA**) function was provided by a single division to all other divisions. This division was managed in numerous directions by successive directors. All artefacts and documents were kept in the classic main storage folder on a corporate drive organised first by division then by folders and then accessible to selected employee levels only. This information-management model proved to be cumbersome and difficult particularly when sections of the storage folder belonged to disparate divisions conducting projects on the same areas of the business. It was evident that knowledge of the current state of the enterprise was largely expert knowledge. Most documented information could not be confirmed as current due to overlapping projects. When areas of the enterprise changed as a result of a project, not all other areas knew of the change. At this time the author decided to define the organisations' EA framework to show each part of the enterprise. The definition of the EA framework resulted in an approach that could apply the BIAT framework onto each business entity on each level of the existing organisational hierarchy. This provided an orthogonal view of EE/A and gave a way for enterprise projects to be assessed for their degree of impact in the BIAT domains for each business area involved in the project. The approach could be applied to each entity in a consistent and measurable fashion due to the identification of each function type, described below.

5.1.1 Identifying the Enterprise

GERAM identifies a number of definitions of the enterprise on numerous levels of abstraction (Williams and Li 1999). It has been found that each business area on each level of the organisational hierarchy is a strategic EBE with its own performance requirements, objectives and drivers (Kaplan and Norton, 1996). GERA supports the EBE can be any part of an enterprise: a multinational corporation, an organisation or government department, a division, a branch, a small-medium business, a cost centre, office, a business unit, team, or a program (Williams and Li 1999). Each EBE can be positioned on the organisational hierarchy where each subsequent level represents a lesser authority (Clark and Thrift 2005). Therefore using GERAM we can say each business area on each level of the hierarchy is an entity. Successful management of a business entity should include planning and measurement. Because planning is strategic in nature, then each management entity has its own strategic domain. The author used the FEA PRM to position the strategic elements of each management entity, showing where its measurement was applied. Since each entity had a strategic domain, the author stacked the AGA BIAT domains underneath it to form a profile for each entity. Whilst the approach was correct, the mistake uncovered a few years later showed there was not enough distinction between the functions of a Management and Control (**MC**) entity, compared to those of a Product and Service (**PS**) entity.

5.1.2 Identifying the Function Types of Each Entity Type

The GERAM approach gives structure to each business area of the enterprise, to identify the MC entities, their strategies, management functions, systems and technologies used; and the same for the PS entities (Bernus, et al, 1997). Whilst this may seem trivial, the distinction is useful when describing the functions of each type of entity. The organisation is the sum of its entities and most entities provide (A) some form of product or service, and are (B) managed to do so. On closer analysis, it can be seen that each entity on each level of the enterprise hierarchy is managed (B), but the products and services (A) of any entity on any level are only provided by the functions of a cost centre belonging to that level. In short, managers perform mostly management functions for each hierarchy level, and employees perform mostly product and service functions at the lowest hierarchy level. This understanding was important for designing the interface in figure 1: the Business screen above, as selecting 'Level Management' presents the MC functions of each layered entity in the hierarchy; and selecting 'Products/Services' presents the PS functions of each nested entity in the hierarchy.

- If the user selects an EBE (a Branch), and selects the Products / Services (2a), then the solution will show all Products and Services (PS) functions for all Cost Centres within the Branch.
- If the user selects any EBE and selects Level Management (2b), then the solution will show the level management (MC) functions for the selected EBE on that hierarchy level only.

This captures the organisational hierarchy and separates the functions of the MC entities from those of the PS entities. The MC functions are used for reporting the performance, while the PS functions are used for providing the performance. The author relates these to the FEA Performance RM and Business RM.

5.1.3 Identifying the Hierarchy of Functions

The function has a hierarchy from its highest level: function, to its lowest level of decomposition: task. The task has a single human role, numerous resources attached to it, can be described but cannot be subdivided. Even if the task is known by many other names, it is still grouped with likewise tasks into a process, which are grouped further as desired. The function name is the connector of the task to the entity which manages it. The hierarchy of the function is important because:

- the PS function is performed by the lowest level PS EBE, which is an entity and is managed;
- the task is a part of the function, and is performed by a person;
- functions are described differently than tasks, which are again different to process descriptions;
- the business information and information system are used by the task;
- the application-information is connected to the information system.

The functional hierarchy is identified because it is where the business processes are described and held. Because of this, the author allocated the functional hierarchy and its contents to the Business Domain. The information, applications, technology and networks were seen to be in place because of the business processes, and therefore were layered underneath the business domain.

5.1.4 Results of Case Study One

The paper-based EA framework was accepted by the authors director who was unfortunately removed from office before implementation. The new broom swept clean and it was never discovered if the approach was correct or not. These following results were found some months later when designing the system for the approach and numerous frameworks to work together:

- The approach captured each part of the enterprise, but unsuitably mixed purpose A and purpose B together. A way was needed to more clearly recognise and portray the difference between management and employee functions, and to identify those roles and their scope of influence.
- The solution needed to capture the hierarchy of the enterprise, and be able to change any name or description of any of its levels.

The approach was extended to include these findings, and was tested in manufacturing and service organisations which focused on capturing the part each business area played in projects, and gave additional findings:

- a way was needed to capture each project schema, and to map it to each entity involved in the project, thus creating the timeline,
- a way was needed to map each project artefact to where it resides in the enterprise profile, and

5.1.5 Functional Subjectivity

The author found the grouping of related activities on selected hierarchy levels is subjective and according to the desires of its creator regardless of whether the function is made up of one or many tasks. In case study one a process-model seven layers deep was re-organised to a four level hierarchy. This showed the author the functional hierarchy for MC and PS functions can be the same because their representation is subjective. There are numerous names for each level and a different number of levels of decomposition for each function in some organisations. This proved to be a key difficulty in achieving process standardisation and interoperability between business areas in case study one. However because the layering is subjective, and the function always has a highest level (function) and a lowest level (task), the author suggests using four levels because the basic tenet for standardisation of most things is to standardise its representation. Identifying the grouping and layering of items into levels as being subjective allows the Bookshelf to use one interface for the organisation to standardise their functional-hierarchy levels, and then for each business area to complete the details. In this way a Keiretsu involving numerous contractor SMEs and one main guiding MNC can create their own four layer pattern for interoperability purposes. The approach and the solution were reworked and refined to include these requirements in a web-based prototype. The prototype was used in case study two by other researchers as the base to create industry specific programs. The author has drawn on their published articles and his own experience for the second case study.

5.2 Case Study Two: Multinational Mining Enterprise

The second case study was to use the prototype in an organisation where architecture was a topic confined mainly to the IT Department. The prototype was tested and verified against stringent business requirements:

1. Capture all entities of the organisation.
2. Give management a view of the business areas they were responsible for.
3. Map the entities impacted by projects before the project begins.

4. Create and manage the architectural domains, their multiple frameworks and artefacts (models, templates, assets and other forms of documents and deliverables, their version control, check in / check out, and life-cycle management functionality), and programs of work (assignment of subordinate projects, roles, tasks, activities and steps).
5. Manage programs, projects, roles, resource utilisation, and performance.
6. Create and manage a workspace allowing each user various views per role.
7. Automate and integrate the TOGAF framework into the operations of the enterprise by pre-populating EA frameworks (TOGAF) and their document repositories for various workgroups based on their project type, and allowing management review of all project requirements including roles, artefacts and other resources against milestones and critical review points.
8. Map similarity between artefacts, helping achieve standardisation.
9. Help manage the delivery of programs and their projects, including task management and status.

5.2.1 Results of Case Study Two

The prototype satisfied the business requirements. It correctly mapped the business functions occurring in the different levels of the organisation, and allowed local naming so as to be suitable for all areas and domains of the enterprise. However, through discussions some additional findings suggested some desirable changes.

1. Stakeholders preferred variants of the information depending on their role.
2. Each actor wanted to use their own words and methods to map more naturally their work protocol, and for information to be communicated in a form appropriate for them.
3. A preference was expressed for information to be viewed per area of responsibility (programs, projects, and geographical areas), and for responsibility to be viewed per domain in relation to those EA items that would change consequent upon an event in a project on any part of the EBE.

Unfortunately the researchers report the lines of communication between technical and business personnel were not amenable to productive debate. There were strongly held, incompatible views of the architecture function and progress depended upon *how useful the line business units find [EA] in supporting the business run initiatives and how well they are able to maintain and upgrade core EA assets (global EA Governance Framework) and communicate them in a timely and effective manner* as described by Turner et al (2009) *there was confusion or misunderstanding as to where all key architectural artifacts reside*. Case study two showed the prototype was deficient and a refined approach and solution was developed. The solution was holistically redesigned allowing for modular interoperability, numerous existing functionalities were refined and simplified, some were removed and some added. The solution had to be usable by most employees, provide relevance as perceived by internal factions, and allow each in-house expert or SME to express their expertise.

5.3 Case Study Three: Healthcare Enterprise B

The third case study spent a year verifying (but not implementing) the Bookshelf with a multi-campus healthcare organisation where the design, operation and understanding of architecture was a topic easily discussed by numerous divisions and departments, just using different names. The enterprise structure followed the shared-model concept where nearly all ICT support was provided by one division for all other divisions. The hierarchy of the organisation had four levels of management from the CEO (highest level) to the director of each cost centre (second bottom level). Each EBE had consistency in the number of management levels within each cost centre. This provided uniformity within the enterprise and how it managed its numerous layers of customers and clients. Architecture was seen by the enterprise as complex, but able to be organised and controlled largely due to the efforts of its information division. The dominant ethos within this division was aimed towards effective internal communication for the successful management and refinement of the delivery of services to the business, so the business could deliver its services. Knowledge of the current state of the enterprise was expert knowledge openly discussed and also captured in documents for each particular project. Customer-based (patient or care) knowledge was captured differently to enterprise-based (strategic, tactical and operational business management) knowledge, yet all knowledge was confirmed by other actors as part of daily work practices. The documents describing the enterprise were held in one central storage area with access controlled by division, then by folders, and then to selected employee roles. This information-management model proved to be cumbersome and difficult particularly when the storage area belonged to disparate divisions and information efforts overlapped. It was a history of the past, where the present was unknown. Due to the questionable currency of documents, each area of the enterprise to be impacted by any new project was re-analysed.

5.3.1 Results of Case Study Three

The Bookshelf was assessed against more than twenty projects over a year but never implemented due to contractual reasons. This led to further refinement of the approach and to another redesign of the solution, because:

- A way was needed to connect the organisational hierarchy with the functional hierarchy and better accommodate the difference in the management levels within each cost-centre.
- A way was needed to accommodate the different languages used by different levels of users in different business areas.
- Users needed the ability to view all business functions involved in the end-to-end solution, drill down to any level of architectural description, and to capture that description.
- Users needed to see the different entities in each value chain, supply chain, business case workflow, patient workflow, patient flow or info stream.

- A method was needed to create and show the repository for employees to enable capture of their documents, artefacts and EA descriptions for each domain in each entity, to capture and show the policies and standards for each entity, to capture and show the QA and review processes for each information item, and to update these as part of work practices.
- The EA governance and management models required integration into the approach and the solution to ease the burden of compliance for these issues.
- The information and knowledge contained in the enterprises data is subject to government-imposed confidentiality, to corporate confidentiality agreements, and to very high social expectations, therefore the Bookshelf had to be secure and able to operate within the clients firewall.

5.4 Combining the Findings

The aim of case study one was to provide a common profile for each business entity, which would combine to form a picture of the entire enterprise including buyers and suppliers. Case study one showed:

- Each entity in a hierarchy has its own Strategic Domain.
- Managers perform management functions for each hierarchy level and employees perform product and service functions at the lowest level.
- The functional hierarchy of each business entity could be captured in a common way for standardisation, comparison and business modelling.
- All business-function information belongs to the Business Domain.
- Adding the other AGA domains underneath these creates a profile view.
- Each domain has a number of layers, which have a number of categories, which are also subdivided.

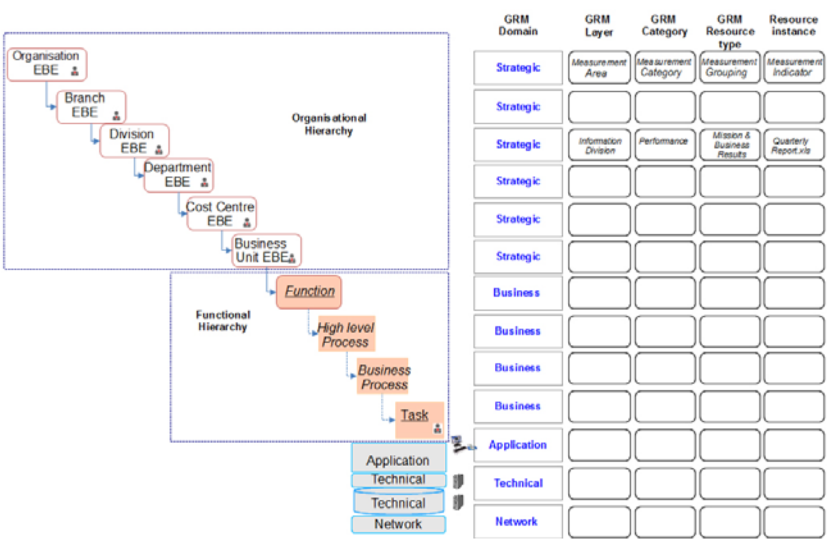


Fig. 5. The hierarchies define the GRM

Case study two showed the solution had to be useful for managers as well as technologists, and had to accommodate corporate politics and be resilient to the socio-technical and socio-political phenomena within the enterprise.

Case study three showed the domains of the GRM should be the 'strategic, business, application, technology, and network' domains, to capture the enterprise profile relevant to each task and to where it resides in the organisation. The task is tied with the information system used. Because of this connection, the Information Domain is organised as two sub-domains, as Business-Information and as Application-Information (an extension of Kilpelinen 2007).

Figure 5 shows connecting the organisational and functional hierarchies creates a line of sight upwards from the task through it's management layers, and downwards to the application and infrastructures that support the task. When profiled, it creates the GRM. How the contents of the GRM are organised is described below.

5.4.1 Defining What's Important

Most of the layer names were drawn from the FEA RMs which provide the taxonomy and ontology for describing resources within the enterprise. While case study one showed a lot of information could be put into each domain for each business area, case study two showed the information describing each business area in the domain of reference had to be collected and classified according to established frameworks for that domain, and in the users language. The author used the four layer FEA PRM and TRM hierarchies to create the four layer GRM hierarchy.

Table 1. Table 2: The GRM hierarchy

Layer	**View: PRM**	**TRM**	**GRM**	**GRM example**	**GRM example**
1st	Strategic	Technical	Domain	Strategic	Technical
2nd	Measurement Area	Service Area	Layer	Architecture office	Integration
3rd	Measurement Category	Service Category	Category	Performance	Core Diagrams
4th	Measurement Grouping	Service Standards	Resource Type	Customer Results	Integration Overview
Instance	Measurement Indicator	Service Indicator	Resource Survey.doc	Client	Integration.doc

Because each business entity is a management entity, each has its own strategic domain. Therefore the name of the entity is the name of the only layer within the strategic domain. The author categorises each entity's strategic information into the Description, Direction, Performance and Portfolio categories. This approach allows each business area on different hierarchy levels to follow the same hierarchy of classification:

Whilst there is no definitive answer to the names of each layer within each domain, the author found the most suitable is *what the user feels is important, and*

Table 2. Table 3: aligning FEA and GRM hierarchies

Layer	View: GRM	GRM	GRM	GRM
1st	Strategic	Strategic	Strategic	Strategic
2nd	Health Dept.	Information Division	Strategy, Governance, Arch Services	Architecture Office
3rd	Performance	Performance	Performance	Performance
4th	Business, results	Business, results	Business, results	Business, results
Instance	Annual report.pdf	Quarterly Report.xls	Performance.doc	Project Results.doc

what the specialist knows where to put. The solution was likened to a bookshelf where users could name any shelf (the domain), any category in any domain, the 'type' of information required for that category, and the document, artefact or information that belongs there in context to the business area being described. The solution was named the 'Company Bookshelf'. Putting the enterprise profile into a web-based system gives a mechanism for every level of management reflection showing who controls what, and the management and decision structures (Bernus, 2011). This is meant to allow organisations to capture their business-as-usual processes to be verified, measured, refined and improved.

6 Practical Significance 2: Capture the Enterprise

This paper has so far described how the Bookshelf profiles a business entity, however the question of its suitability remains. To be suitable a solution has to be useful, easy and to do its function. Unfortunately the connections of the concepts used within the solution are not easy to immediately understand but the author hopes that once understood it becomes simple; hopefully akin to learning how to drive. The key function of the approach is to let users define what is applicable and important for each domain and then to provide the evidence from each business area. This captures the current state of the enterprise. Although the solution may be capable of capturing the necessary components of an enterprise, doing so makes light of the fact that people in the organisation would have to be motivated and disciplined enough to achieve this monumental feat. It was tested in case studies one and three, and in other engagements that capturing the ideal state of accurate, complete, and up-to-date articulation of the enterprise is not a smooth exercise when self-directed. However the author found this exercise is easier completed when driven by projects or consultants. This is because the business analysts or consultants enter business areas and ask the *what do you do, why, what do you use, why, what do you report, why?* business and systems analysis questions in their current state analysis. This type of approach allows the governance and discipline for implementation of the Bookshelf to be addressed in a step-wise manner as the work, tasks and resources of each enterprise actor in each business area are added to their enterprise profile.

When employees, including management and executives see their processes in their context, the processes are much easier to be followed and maintained. The author believes as the people working in each business entity maintain their part of the Bookshelf, it would allow an always-updated information schema of the organisation and the elements within it. This would answer the questions of how to capture the current state of the enterprise, and how to keep the information current. The solution is multi-user for all organisational levels and is designed to support people in the enterprise in a distributed manner. Unfortunately, the author feels this effort holds no convincing argument yet, as the layout of this paper describes a succession of directed case studies. Collectively, these detail how the failings of numerous versions resulted in an approach, method and system for an organisation to capture information that people think is important. The author believes the Bookshelf has matured to a level which supports the representation, analysis, exchange and beneficial manipulation of business and technical information by being the mechanism that provide the coherency between the information and the talking. This fine point is important as it's where the convincing argument lies.

7 Conclusion

The author does not purport to provide a solution to the problem of creating a complete and universally applicable reference model for all the architectural elements of an enterprise, as this (for the purposes of classification) has been respectively been collectively provided by the FEA, GERAM (Bernus et al 1997) and numerous other solutions (Uppington, 2011). Instead the author proposes a solution that captures the structure of most organisational forms and allows each to attach their preferred functional hierarchy, which is then added to a generic reference model. This creates the Enterprise Profile, against which each business entity within that organisation is mapped. This results in a profile for each business entity on each level of the organisation, allowing MC and PS entities to be compared on the same plane or combined with others in business-model simulations. Each profile is editable by the actors of each entity, who can enter their own names, terms, documents and artefacts to describe their world in their way. This captures the information that its' people think is relevant for their organisational history, operations, planning, and decision making. The author feels the success of this research and its convincing argument will only be determined when the approach, method and system has been used and assessed within a number of organisations. The author hopes the approach and the solution described in this paper will help organise the complexity of enterprise architecture and will give insight for organisations to improve on what they already have. This insight would not only make a leaner system, but a much more efficient system where staff have a clearer picture of what their goals are, what they have to achieve, how they can achieve it and what tools they can use.

References

1. Bernus, P., Nemes, L.: Requirements of the GERAM. Annual Reviews in Control 21, 125–136 (1997)
2. Bernus, P.: (1999), http://www.cit.griffith.edu.au/~bernus/ims/gm21/tg4wp1/models/
3. Bernus, 2011: personal communication
4. Clark, G.L., Thrift, N.: The Return Of Bureaucracy: Managing Dispersed Knowledge In Global Finance. In: The Sociology of Financial Markets by Karin Knorr-Cetina, Alex Preda, pp. 229–249 (2005)
5. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
6. Kilpelinen, T., Nurminen, M.: Applying Genre-Based Ontologies to Enterprise Architecture. In: ACIS 2007 Proceedings (2007)
7. Turner, P., Gøtze, J., Bernus, P.: Architecting the Firm – Coherency and Consistency in Managing the Enterprise. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 162–171. Springer, Heidelberg (2009)
8. Uppington, G.: EA Framework Inventory (2011), www.companybookshelf.com (last viewed August 2011)
9. Williams, Li: PERA and GERAM - enterprise reference architectures in enterprise integration. In: DIISM 1998 (1999)

Towards a Method for IT Service Management

Heiko Kattenstroth and David Heise

Information Systems and Enterprise Modelling Research Group
University of Duisburg-Essen
Universitaetsstr. 9, 45141 Essen, Germany
{heiko.kattenstroth,david.heise}@uni-duisburg-essen.de

Abstract. To cope with the challenges in IT service management, methods are required that purposefully reduce the complexity inherent to enterprises – with regard to both business and IT –, facilitate communication among groups of stakeholders and support the management of IT services along their life cycle. In this paper we investigate the potentials of an enterprise modelling approach to IT service management and reflect upon design alternatives for corresponding modelling constructs. We present research in progress that is intended as foundation for discussion with and discursive evaluation by peers and domain experts.

Keywords: IT service management, service life cycle, domain-specific modelling, enterprise modelling.

1 Motivation

Today's IT organisations are confronted with remarkable challenges: First, they have to deal with the tremendous *complexity* of present day enterprises and their IT. A complexity that results from the multitude of IT platforms, networks, and information systems as well as their interrelations. At the same time they are expected to efficiently support the business and to drive innovations, which requires IT managers to account for the multifaceted *dependencies* between IT and the enterprise's goals, organisational structures, and business processes. Furthermore, aligning the IT to the business demands the participation of various stakeholders from business and IT, such as executives, end-users, and IT experts. Each of them has a different professional background, perspective on IT and business, and – even more important – technical language. As a result IT managers have to deal with language barriers, which hamper *communication* and collaboration between the stakeholders and consequently compromise the efficiency of the IT organisation.

In response to these challenges, the concept "IT service" emerged as an innovative approach to manage the relationship between business and IT. In current business practice, IT services are widely considered to be of outstanding relevance for IT management, as they, for example, represent the IT organisation's outcome that is used in the company's business processes and serve as a reference between IT and business stakeholders as well as between the different stakeholders inside the IT organisation. Unfortunately the introduction of this additional

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 178–192, 2011.

concept seems to have an ambivalent effect: While it is suited to reduce complexity on the one hand, it also adds to the complexity of IT management on the other. IT organisations are, thus, urged to enforce development, implementation, and maintenance of their IT services – the IT service management (ITSM) – in a structured and systematic manner.

To cope with these challenges, methods are required that purposefully reduce the complexity inherent to enterprises – with regard to both business and IT –, facilitate communication among groups of stakeholders, and support the management of IT services along their life cycle. In this respect, several authors highlight the prospects of supporting ITSM through conceptual models in general and enterprise models in particular (e.g., [1,14]). While current enterprise modelling approaches already provide preliminary conceptualisations for IT services (i.e., provide dedicated abstractions that, among others, reduce the complexity), their support for IT service management is still limited. Against this background we propose to extend a method for enterprise modelling to specifically addresses the requirements and challenges of IT service management. In this paper we present a part of this research in progress: First, we investigate the potentials of an enterprise model-based support for ITSM. In contrast to prior work, we deliberately analyse the potentials throughout the different phases of the service life cycle. And second, we propose corresponding modelling constructs in extension to existing approaches for enterprise modelling. Therefore, we synthesise and reflect upon promising enhancements and discuss design alternatives. The results of our analysis are intended as working drafts and foundation for discussion with and discursive evaluation by peers and domain experts.

In the next section we introduce the ITSM domain. In Section 3, we discuss approaches for enterprise modelling and how they can be used as a foundation for ITSM. The general prospects of such an approach are investigated in Section 4. In Section 5, we reflect on current work on domain-specific modelling constructs for ITSM and discuss design alternatives. Section 6 presents concluding remarks.

2 IT Service Management: Life Cycle and Requirements

IT service management seems to have evolved to an essential part of IT management that focuses on development, implementation, and maintenance of IT services. While in the last century many authors conceptualised (IT) services as "counterpart" to physical goods and products – applying the criteria of intangibility, heterogeneity, inseparability, and perishability – we follow Vargo and Lusch's [32] arguments in that a segregation between goods and services cannot be maintained. Rather we see "IT service" as a broad term that covers *any IT-related outcome of the IT organisation* that is used by internal (e.g., business departments) or external customers (e.g., companies at the market). In this diction, the service term incorporates both tangible and intangible aspects and corresponds to 'product'. Examples for IT services can be found at different levels of abstraction, ranging from fine-granular IT services like "Edit Customer Data" to broad ones like "Providing ERP Application". A method

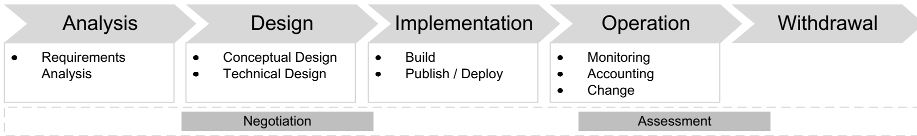


Fig. 1. Idealised life cycle with exemplary activities

for ITSM should therefore provide a conceptualisation of IT service that covers the various, generic interpretations of service, their dependencies and other interrelations.

Moreover, IT service management can be understood as a paradigm that acknowledges the importance of aligning IT with the business ("IT/Business Alignment"). Thus, the key focus of IT management shifted from technology, i.e., hardware and software, to IT as an instrument that aims to support people in creating a business value and is strongly motivated by the demand for cost control and customer orientation. Thereby it seems accepted that IT does not provide a business value per se, but "must be part of a business value creating process with 'other' IS and organisational factors operating in a systematic manner" and that it includes "IT people and management, routines, and policies or the organisational system including non IT-people and management, business processes, knowledge assets, relationship assets, culture structure, and policies" [18, p. 26]. Accordingly, IT is not an end in itself. Instead, it is supposed to support an organisation's business processes, enterprise goals, and – in general – its competitiveness. Hence, adequate management of IT services requires a profound understanding of the organisational context and interdependencies between business and IT. At the same time, ITSM has to account for the technical context, too, since many IT services will be realised by hardware and software. Consequently, a method for IT service management has also to address the technical perspective and provide concepts representing the IT domain.

ITSM frameworks such as ITIL [21] aim at systematically managing the offerings of the IT organisation as services for the business by means of managing the service portfolio. Service portfolio management focuses, amongst others, on reuse of existing services and the strategic development of new offerings and capabilities to serve the customers, which recommends managing IT services during all phases of their life cycle. Moreover, if instruments do *not* support IT service management at all stages of the life cycle – i.e., different instruments have to be applied to cover the different phases – negative effects such as lack of integrity, loss of information, and frictions in communication are likely that finally will hamper the efficiency of IT service management in general. Based on the work from Jin and Ray [15], Kohlborn et al. [17], and Riedl et al. [24], Figure 1 illustrates a synopsis of IT service life cycle phases along with exemplary activities. A method for ITSM thus has to provide support for all phases.

Table 1 summarises the requirements for a method that aims to support IT service management. Based on our experience with ITSM frameworks, methods, and tools, we conclude that there is currently no instrument or methodical

Table 1. Requirements for a method in support of IT service management

Requirement
Req. 1 – Reduce complexity inherent to ITSM domain
Req. 2 – Foster communication among business and IT stakeholders
Req. 3 – Provide concepts for IT services and their interrelations
Req. 4 – Account for the organisational context
Req. 5 – Offer concepts of the IT domain (technical perspective)
Req. 6 – Provide support over entire service life cycle

support available that addresses all requirements. For instance, the ITIL framework provides high-level guidelines for IT service management processes and explicitly spans the service life cycle, but does neither address the business context nor provide concepts at detailed technical level; approaches like "Model Driven SOA" [35] support the technical design and implementation of (web) services, but do hardly address the complexity of the ITSM domain, foster communication with business or account for organisational context. Our research is therefore based on the assumption that both demand and potentials exist for methods and tools that fulfill the described requirements and support IT service management.

3 Conceptual Background

Enterprise modelling (EM) approaches such as ARIS [25], MEMO [11] or ArchiMate [19,29] seem to provide a promising foundation for several reasons: (1) They serve to structure an enterprise by providing purposeful abstractions of IT and the surrounding action systems (*Req. 1*). (2) They, therefore, make use of domain-specific modelling languages (DSMLs), thereby offering a consistent, intuitive and semantically rich modelling [16]. (3) They usually build upon an elaborate meta model that already includes organisational and technical context, such as for goal, business process or organisational structure modelling on the one hand and for modelling the IT organisation on the other (*Req. 4 & 5*). (4) They provide stakeholders with specific and illustrative views on a company at various organisational levels, such as on value chains, business processes, or IT landscapes. Since the DSMLs are seamlessly integrated, they allow for comprehensive analyses fostering IS/Business Alignment and communication among business and IT stakeholders (*Req. 2*). Figure 2 illustrates an example of an enterprise model based on the MEMO family of modelling languages. The excerpt shows models at different levels of an organisational hierarchy, which is intended to address different stakeholders: A value chain and goal model at strategic level, a business landscape and business process model at organisational level, and models of the IT organisation at IT level.

In this regard, some approaches to enterprise modelling already provide preliminary meta types representing an "IT service", though their conceptualisations of the (IT) service term, the resulting meta types and the corresponding

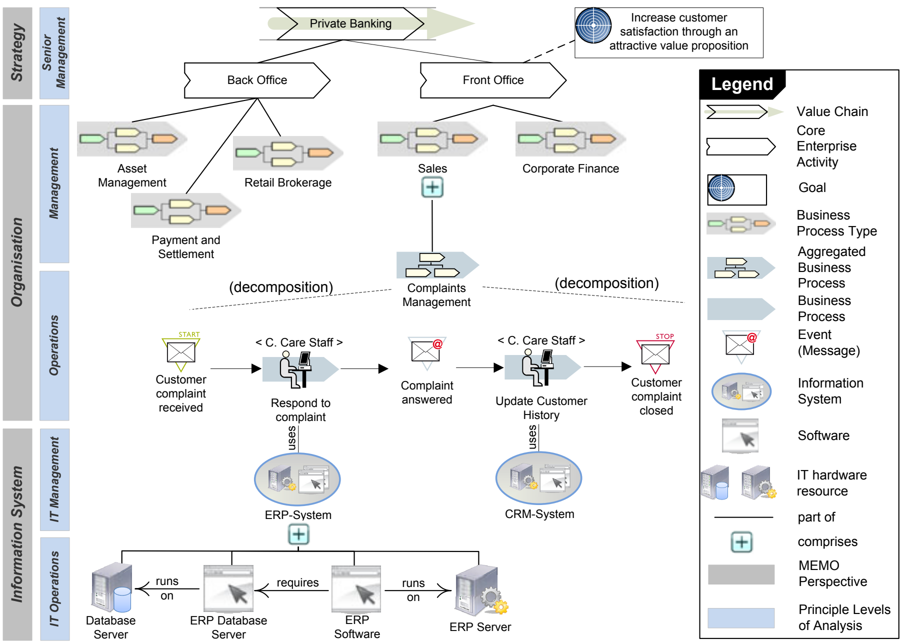


Fig. 2. Example of an Enterprise Model that integrates models of an IT landscape with models of associated business processes from various perspectives

associations vary notably. Furthermore, these concepts have usually not been developed with dedicated ITSM support in mind. Moreover, to our best knowledge, approaches that propose an enterprise modelling-based approach as foundation for IT service management, such as Braun and Winter [1] and Frank et al. [14], do not illustrate how to support IT service management in detail, for instance, for service portfolio management.

At the same time, various authors outside the EM community propose specific DSMLs – at least in the broadest sense – for conceptual modelling of services. For instance, Quartel el al. [22] introduce modelling concepts and a preliminary notation that allow to model services as sequences of user-provider interactions. Hence, they conceptualise services as counterpart to products and goods; technical or organisational context is not considered. An approach that specifically addresses *IT* service management is presented by Correira and Abreu [3,4] as work in progress. They suggest a DSML that aims to support the management of service level agreements over their life cycle; a meta model is not (yet) provided. A meta model for ITSM is, instead, provided by Uebernickel et al. and Ebert et al. respectively [7,31]. Their metatype "IT Service" is embedded in organisational and technical context in that it is associated to meta types representing business processes and users as well as IT resources (e.g., "Information", or "Hardware"). It seems their meta model is not intended to be used for graphical modelling, since they do neither discuss a notation nor provide examples of how to apply the concepts. As attributes are missing from the meta model, the semantics of its concepts are mainly up to interpretation.

Against this background, the paper at hand extends prior work in two aspects: (1) We exemplarily illustrate how enterprise modelling can support ITSM during different phases of the life cycle; (2) we enhance current work by synthesising, reflecting, and enhancing current language specifications for IT service management as well as discussing corresponding design alternatives. Our research follows a design research process that is grounded on a research method configured for the epistemological particularity of research on modelling methods (for a detailed description, see [10] and [27]) and applies a method for designing domain-specific modelling languages [12].

4 Outline of EM-Based IT Service Management

This section illustrates the prospects of supporting life cycle spanning IT service management with a dedicated method, which is based on a domain specific modelling language and is integrated with an approach for enterprise modelling as described in Sec. 3. With respect to space restrictions, the example is limited to selected activities of the life cycle. The application scenario, as well as the specification of the design alternatives in Sec. 5, are based on the MEMO approach. Thus the present models are illustrated presuming modelling languages and notation of the MEMO language family, for instance, for strategic, organisational, and IT landscape modelling [11,14] as well as specific extensions like for risk or indicator modelling [27,28]. It is, however, important to note that shown diagrams and meta models are not intended to predetermine a specific enterprise modelling approach; instead, they serve as an illustration of principle application and design decisions in context of enterprise model-based ITSM.

Figure 3 illustrates the application scenario: In the *analysis* phase the desired functionalities, constraints, and expectations of the business stakeholders have to be captured as requirements. With regard to *Req. 4* it seems promising to use models of the organisational context (e.g., business processes, organisational structure, and business goals) since they provide an overview of the business processes the IT service will be used for and the business goals (and restrictions) it aims to support. These models are enriched with a description of the required functionalities (see ① in Fig. 3). In contrast to common requirements engineering approaches such as [5,34] these requirements are *integrated* with the enterprise models in that they are associated to the concepts they affect. The required functionalities are then structured as potential IT services ("service candidates") based on their scope, purpose and similarities. To foster reuse, the resulting candidates are mapped against existing IT services, and the functionalities they offer, using the IT service model (②). The IT service model contains the services and their functionalities, dependencies (such as "uses" or "requires") as well as similarities ("similar to") and specialisation/generalisation relationships between IT services. In this regard, the objective of this service mapping is to identify services that can be reused, extended or adapted to fulfill the requirements. This allows for a comprehensive and sustainable management of the

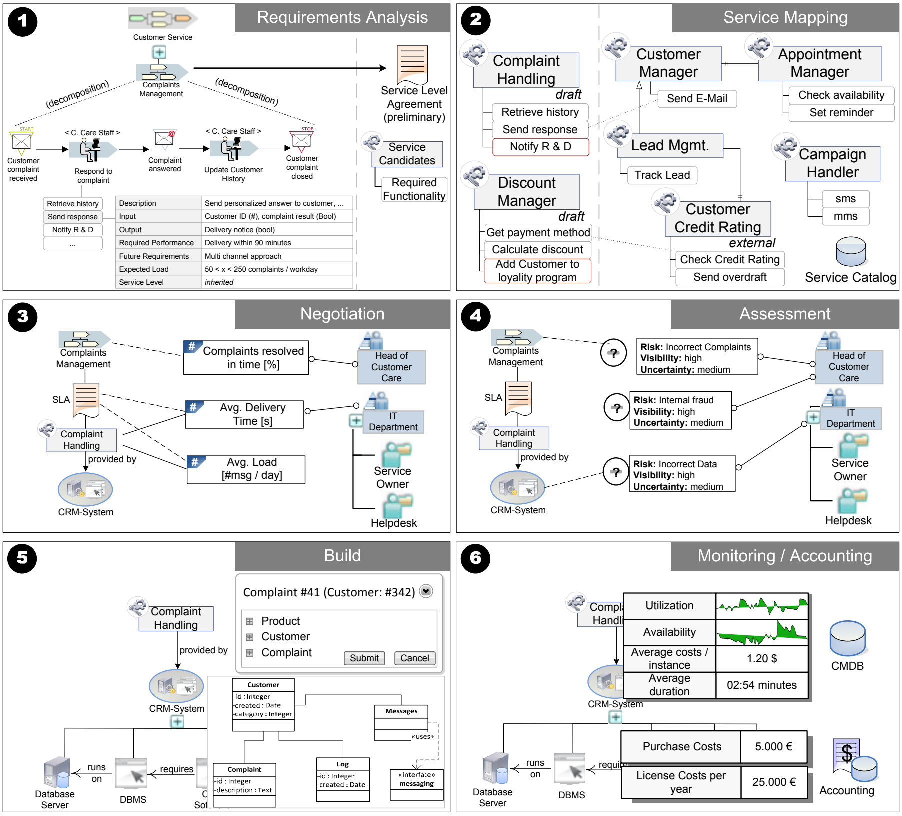


Fig. 3. Example for a model based ITSM

IT service portfolio. To support the negotiation between service provider and service customer, service level agreements act as contracts that specify responsibilities, additional IT services (e.g., a help desk) and, in general, their expected quality. For this purpose the IT service model is enriched with, for instance, (key performance) indicators and corresponding thresholds the service provider has to comply with (③). Enterprise models provide an overview of potential sources from which risks originate and thereby support risk identification. The identified risks and corresponding cause-and-effect relations can be added to the models (④). This allows for evaluation and prioritisation of risks related to IT services, e.g., to make decisions on their treatment and to implement appropriate measures. For the implementation of new IT services and functionalities, the IT service model – along with corresponding SLAs and architectural and implementation standards – is used for the conceptual and the technical design of the prospective service. Furthermore, the model can be enriched with implementation details, e.g., about hardware, software, or network resources as well as data models, interfaces, and GUI drafts (⑤). Again, models promise to foster the communication, in this case between different groups in the IT department and the "service owner", for example, to get feedback and to ensure the alignment between business requirements and service implementation. For higher productivity, models can also be used to generate (parts of) the service realisation such as interface descriptors, schema for databases, and program code [16]. Once deployed and running, the IT services need to be monitored, maintained and evaluated. In this context, Frank et al. [13] employ the idea of using *models at run-time*. Accordingly, the service model is used for operations in the run phase (⑥), for instance, to enforce quality and availability as guaranteed in the SLA. Whereas models in the plan and build phases remain on type level, the models in the run phases are enriched with data about the service execution (i.e., instance data). Thus the models are used as frontend to instance data ("dashboard"), which promotes a more proactive IT service management. Presuming an integration with corresponding operational information systems, these models could also be enhanced by their running costs, current availability and utilisation as well as other relevant information.

5 Considerations on Method Design: Reflections on Service Modelling

Development of a method for IT service management requires reconstructing the key concepts of the domain as modelling concepts and providing a process model that guides the application of these concepts. The key domain-specific concepts seem obvious: The method has to provide concepts that represent IT services (*Req. 3*), the business context the IT services are used in (*Req. 4*) as well as concepts that represent the underlying IT infrastructure (*Req. 5*). The conceptualisation in detail, however, depends on design decisions between alternative specifications. In this section we therefore discuss considerations on the language specification and design alternatives (Sec. 5.1), which are based on existing conceptualisations, and outline open issues for a method for IT service management (Sec. 5.2). The specifications are presented as metamodel excerpts in notation of the MEMO Meta Modelling Language [9]. The meta models are intended as working drafts and foundation for discussion with and discursive evaluation by peers and domain experts. The reuse of modelling concepts from existing modelling languages in the MEMO language family is visualised by a colored rectangle attached to the meta type header (as suggested in [9]).

5.1 Outline of a Metamodel: Design Alternatives

IT service concept. Pertinent literature, ITSM approaches and ITSM standards such as [1,2,21,23] apply a generic definition wherein an IT service is used as an abstraction over any (IT-related) outcome of the IT organisation that is used by a customer. This is mainly for one reason: A vague definition of "IT service" enables stakeholders from different hierarchical levels (e.g., management vs. operations), organisations, and domains (e.g., business vs. IT) to communicate *without* clarifying their (maybe different) conceptualisations in detail. Current

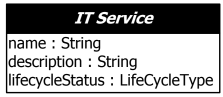


Fig. 4. Basic Conceptualisation of IT service

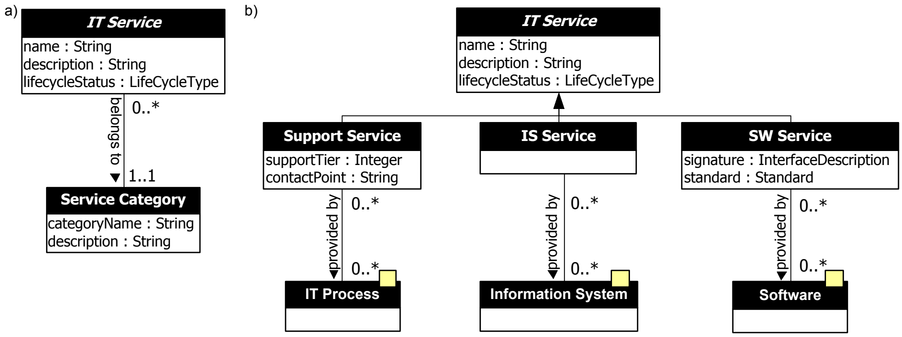


Fig. 5. Design alternatives for IT service

modelling languages like [1,14] seem to acknowledge this practice and offer a generic service concept as illustrated in Fig. 4: A meta type *IT service* serves to describe a service at least by means of a name, a description and its status in the life cycle (such as "planned", "in implementation", "operated" or "inactive").

However, despite the advantages of such a broad service concept it also comes with various deficiencies: Though stakeholders might be able to communicate for a start, it bears the risk of (maybe unrecognised) misunderstandings, especially if stakeholders apply a different conceptualisation of IT service in detail; and since it does not provide much semantic, it does not allow for a specific support throughout the different life cycle phases as illustrated in Sec. 4. Thus it seems reasonable to provide a more specific conceptualisation that accounts for different connotations of "IT service". We reason that the IT-related service term is used on at least three different levels of abstraction (for a similar conclusion, see [22]). On the one hand an IT service is used as an abstraction over *processes* performed by the IT organisation. Examples for such IT services are "Help Desk" or "PC installation" (e.g., [30]). On the other hand, it refers to a more technical conception with a focus on features that are provided from *information systems*, like "Invoice Printing" or "Manage Customers" offered by an ERP system. Finally, the IT service term is used to describe functions (i.e., operations) of *software*; prominent examples are services in context of SOA as in [8] (e.g., "Retrieve Customer" or "Add Customer").

Against this background, we recommend providing a differentiated conception of IT service in the modelling language as well. One solution would be to offer a meta type for *IT service categories* that allows organisations to define categories of IT services, such as "IS-based services" or "software-based services" (cf. fig. 5a). Though this solution supports stakeholders in interpreting the semantic

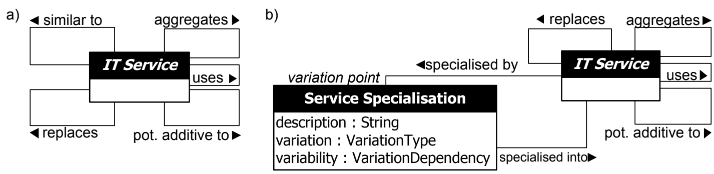


Fig. 6. Design alternatives for service relations (cardinalities & constraints ommitted)

of IT service conceptualisations, is does still not provide service-specific semantic and thus no support for tool-based analyses or code generation. Figure 5b illustrates a design alternative that reconstructs the different types of IT services *as meta types* (a similar conceptualisation has been chosen in ArchiMate [29]). In contrast to the service categorisation this alternative offers semantically rich concepts, since it allows to (1) specify IT service type-specific attributes and to (2) associate the meta types to other meta types that further describe them, e.g., associate *IS-based services* to a meta type that represents information systems etc. This promotes a more precise modelling, enables tool-supported analyses and supports code generation, e.g., for software services.

IT service relations. Analysing and managing IT services recommends to account for the various relations between IT services. For instance, Braun and Winter [1] suggest an *isPartOf* relation between services; Brocke et al. [2] differentiate the service meta type into core services and associate services, that customise the core services; while Weigand et al. [33] classify services into, e.g., core, enhancing, complementary and support services. Since this distinction might be a matter of perspective – while one customer regards a service as complement, others might consider it core – we refrain from such separation in the language specification; instead we regard them as context specific *roles*. However, it might be promising to introduce an association *potentiallyAdditiveTo* between IT services. Following the proposal from Ebert et al. [7], we also suggest to differentiate the *isPartOf* relation: On the one hand an IT service might be an abstraction over other IT services, without providing own functionality (*aggregates*). On the other hand an IT service might, in turn of its execution, optionally or mandatory revert to other IT services (*uses*). In extension to current work, two further relations seem promising: In support of the withdrawal of IT services an IT service can be dedicated to *replace* other IT services; and an IT service can be *similarTo* other IT services, which might be feasible especially for larger corporations with several local IT organisations. Figure 6a illustrates the design alternative.

With respect to ongoing research in product modelling (see, e.g., [26]), one design alternative seems worthwhile that builds upon the idea of generalisation/specialisation. IT services that are similar to a large extent are generalised to an IT service that describes their common features; additionally, features that vary among the specialised IT services are explicitly modelled as “variation points” (fig. 6b). This alternative contributes to transparency and fosters communication with business stakeholders, as it supports abstracting over all the different variations. Moreover, it specificially promotes a systematic and

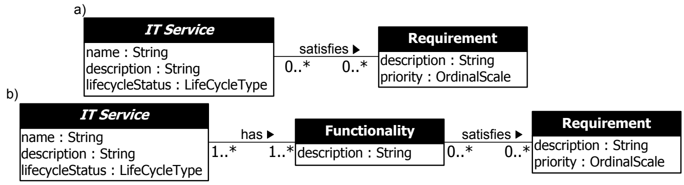


Fig. 7. Design alternatives for service functionalities

elaborate portfolio management of IT services. However, although fig. 6 indicates corresponding language specifications (and, for instance, ArchiMate already offers service specialisation), the semantic of service specialisation, service variation points and similar-to relations has not been clarified yet and is subject to further research. As for similar-to relations, we assume that this issue might – at least partly – be addressed by particularising IT services into *functionalities*.

Functionality. An IT service is used as abstraction over IT processes, information systems and software. At the same time an IT service is used to describe and bundle functionalities that address customers' requirements. Two design alternatives are feasible to represent a service's functionality. First, the concept of functionality is considered being part of the IT service, esp. the description attribute (fig. 7a); this alternative conforms to most current state of the art in EM approaches, e.g., [14,29]. Second, a dedicated meta type represents the functionalities of an IT service (fig. 7b), as suggested by Brocke et al. [2]. The first alternative requires organisations to formulate their services in a specific, functionality-oriented manner and, thus, restricts to describing one functionality. As a result, an IT service that offers several functionalities has to be modelled by using the above mentioned aggregation mechanism, i.e., the IT service has to comprise several other IT services that each provide one of the required functionalities. The second alternative, in contrast, supports to (1) assign several functionalities to one IT service and (2) allows to separate a general, maybe more business-intuitive description of the IT service and the description of the functionality. Furthermore, a separate modelling of functionalities might be a prerequisite to using service-specialisation and variation points – though this depends on the semantic of these concepts.

Business context. To account for the organisational context IT services are used in, we propose the integration with existing modelling languages, e.g. for business process and goal modelling. Besides "traditional" concepts such as business processes, goals, and organisational structures, we further propose to integrate concepts for risks and indicators as these aspects play a crucial role for today's IT management (fig. 8): Against this background, we suggest integrating the IT service concept with concepts representing risks [28] and indicators [27].

Service level agreement. The relations between a service provider, the IT services it provides and the customers of these IT services are often contractually agreed

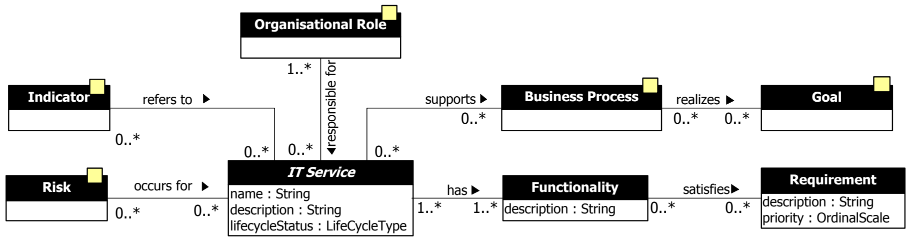


Fig. 8. Business context for IT services

in service level agreements (SLAs). At least two alternatives seem feasible: First, SLAs are considered to govern IT services in that IT services have to be provided with the agreed quality (fig. 9). Similar conceptualisations have been proposed in [1,6]. Since a SLA might be specific for the relation between service provider and service customer, a second design alternative is to refine the association between IT service and business process, which acts as a surrogate for the service customer (fig. 9). Whereas Moser and Bayer [20] propose to associate SLA with an IT service and a business unit, we use the concept for business processes as this more precisely describes the point of use of the service.

In both alternatives a SLA is constituted, amongst others, by a set of indicators, responsible organisational roles, and goals. Usually these aspects will a be subset of indicators, roles, and goals of the IT service the SLA refers to, which recommends corresponding constraints to assure consistency in the model. Additionally a SLA can contain billing details for the IT service (e.g., *costs*). Although the second solution promotes a more precise definition of the relation between IT service and business process and thereby of a SLA, this alternative might be counterintuitive for users that are accustomed to assign IT services directly to a business process.

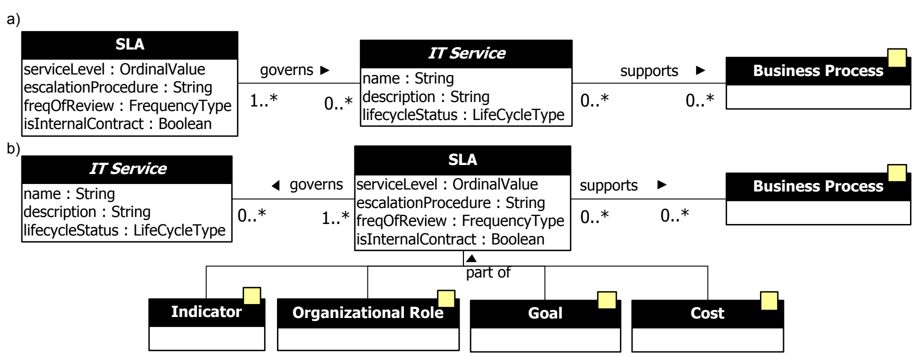


Fig. 9. Design alternatives for service level agreements

5.2 Open Issues and Research Agenda

The presented reflections on IT service modelling already indicate challenging questions that need to be investigated. Although we presented alternative design decisions and, as far as reasonable, decision criteria and rationales, we highlighted just a selection of open questions. Development of an enterprise modelling method for IT service management requires considering further design issues. Among others, open research questions are:

- Some authors introduced a differentiation of IT services and IT products (e.g., [7]). While this separation seems reasonable – a product bundles services to be sold, thus applying a marketing perspective – it entails a problem if the specialisation/variation point concepts (cf. fig. 7b) are applied: Assuming that these concepts make sense for product modelling, too, the relation between product, specialised products, IT services and specialised IT services has to be investigated.
- In fig. 5 we assumed a meta type representing IT processes. However, yet there seems no consensus – nor analysis – whether there is need for a specific meta type or if existing modelling concepts and languages for business processes can be reused to also model IT processes.
- With regard to the requirements analysis it needs to be investigated (1) to what extent requirements can be derived from process models and (2) how additional details such as capacity requirements and anticipated changes can be added to process models and used in later life cycle phases.
- Similar and reusable services (and functionalities) can only be identified if they are described *consistently* and at the same level of abstraction – at best using (semi-)formal concepts that support tool-based comparison. In context of enterprise modelling such concepts and corresponding heuristics are, to our knowledge, subject to future research.
- Finally, modelling SLAs, as indicated in fig. 9, recommends accounting for juridical aspects, too. However, we are not aware of work that addresses the formalisation juridical aspects, yet.

Beside issues pertaining the language specification, the application of the presented modelling concepts needs to be embedded in a process model that envisions the use of the concepts; in addition, heuristics and guidelines are recommended. These aspects were not in focus of this paper.

6 Conclusions

This paper investigates the potentials of an enterprise modelling based method in support of IT service management. In this regard, our contribution in this paper is twofold: First, we illustrate consistent support for ITSM over the entire life cycle of IT services. Second, we prepare for further research on such a method by reflecting key considerations and decisions pertaining to IT service modelling. Therefore, we focus on language specification and discuss design alternatives as enhancements to existing approaches as research in progress. However, developing a method requires further considerations, for instance, guidelines, an accompanying process model, and support of a modelling tool. Since corporate culture and the distribution of power in an organisation have an impact on IT management [23,30] it should also be investigated, whether these aspects, which apparently resist against formalisation, could be somehow accounted for in the method, too.

References

1. Braun, C., Winter, R.: Integration of IT Service Management into Enterprise Architecture. In: Proc. of the 2007 ACM Symposium on Applied Computing (SAC 2007), pp. 1215–1219. ACM, New York (2007)
2. Brocke, H., Uebernickel, F., Brenner, W.: Managing the Current Customization of Process Related IT-Services. In: Proc. of the 43rd Hawaii Int. Conf. on System Sciences (HICSS-43), Koloa, Kauai, Hawaii (2010)
3. Correia, A., Abreu, F.B.E.: Integrating IT Service Management within the Enterprise Architecture. In: 4th Int. Conf. Software Engineering Advances (ICSEA 2009), pp. 553–558. IEEE, Los Alamitos (2009)
4. Correia, A., Abreu, F.B.E.: Defining and Observing the Compliance of Service Level Agreements: A Model Driven Approach. In: Proc. of the 7th Int. Conf. on the Quality of Information and Communications Technology, pp. 165–170. IEEE, Washington, DC, USA (2010)
5. Davis, A.M.: Software Requirements: Objects, Functions, and States. PTR Prentice Hall, Englewood Cliffs (1993)
6. Debusmann, M., Kroger, R., Geihs, K.: Unifying Service Level Management Using an MDA-Based Approach. In: IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), pp. 801–814. IEEE, Piscataway (2004)
7. Ebert, N., Uebernickel, F., Hochstein, A., Brenner, W.: A Service Model for the Development of Management Systems for IT-Enabled Services. In: Proc. of the 13th Americas Conference on Information Systems, ACIS 2007 (2007)
8. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall Professional Technical Reference, Upper Saddle River (2005)
9. Frank, U.: The MEMO Meta Modelling Language (MML) and Language Architecture. Technical report 24, ICB, University Duisburg-Essen
10. Frank, U.: Towards a Pluralistic Conception of Research Methods in Information Systems Research. Technical report 7, ICB, University Duisburg-Essen
11. Frank, U.: Multi-Perspective Enterprise Modeling (MEMO): Conceptual Framework and Modeling Languages. In: Proc. of the 35th Hawaii Int. Conf. on System Sciences (HICSS-35), Honolulu (2002)
12. Frank, U.: Outline of a Method for Designing Modelling Languages. Technical Report 42, ICB, University Duisburg-Essen (2010)
13. Frank, U., Heise, D., Kattenstroth, H.: Use of a Domain Specific Modeling Language for Realizing Versatile Dashboards. In: Proc. of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM 2009), Orlando, Florida, USA (2009)
14. Frank, U., Heise, D., Kattenstroth, H., Ferguson, D., Hadar, E., Waschke, M.: ITML: A Domain-Specific Modeling Language for Supporting Business Driven IT Management. In: Proc. of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM 2009), Orlando, Florida, USA (2009)
15. Jin, K., Ray, P.: Business-Oriented Development Methodology for IT Service Management. In: Proc. of the 41st Hawaii Int. Conf. on System Sciences (HICSS-41), Waikoloa, Hawaii, USA, p. 99 (2008)

16. Kelly, S., Tolvanen, J.-P.: Domain-Specific Modeling: Enabling Full Code Generation. Wiley-Interscience and IEEE Computer Society, Hoboken, N.J. (2008)
17. Kohlborn, T., Korthaus, A., Rosemann, M.: Business and Software Service Lifecycle Management. In: 13th Int. Conf. on Enterprise Computing (EDOC 2009), Auckland, New Zealand (2009)
18. Kohli, R., Grover, V.: Business Value of IT: An essay on Expanding Research Directions to Keep Up with the Times. J. of the AIS 9(1), 23–39 (2008)
19. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communiation and Analysis. Springer, Heidelberg (2005)
20. Moser, C., Bayer, F.: IT Architecture Management: A Framework for IT Services. In: Proc. of the Workshop on Enterprise Modelling and Information Systems Architectures (EMISA), pp. 137–151 (2005)
21. Office of Government Commerce. ITIL – Service Design. London (2007)
22. Quartel, D.A.C., Steen, M.W.A., Pokraev, S., Sinderen, M.J.: COSMO: A Conceptual Framework for Service Modelling and Refinement. Information Systems Frontiers 9(2-3), 225–244 (2007)
23. Rands, T.: Information Technology as a Service Operation. J. of Information Technology 7(4), 189–201 (1992)
24. Riedl, C., Böhmann, T., Rosemann, M., Krcmar, H.: Quality Management in Service Ecosystems. ISeB 2(7), 199–221 (2009)
25. Scheer, A.-W.: Architecture of Integrated Information Systems: Foundations of Enterprise Modelling. Springer, Berlin (1992)
26. Sinnema, M., Deelstra, S., Nijhuis, J., Dannenberg, R.B.: COVAMOF: A Framework for Modeling Variability in Software Product Families. In: Nord, R.L. (ed.) SPLC 2004. LNCS, vol. 3154, pp. 197–213. Springer, Heidelberg (2004)
27. Strecker, S., Frank, U., Heise, D., Kattenstroth, H.: MetricM: A Modeling Method in Support of the Reflective Design and Use of Performance Measurement Systems. ISeB (2011)
28. Strecker, S., Heise, D., Frank, U.: RiskM: A Multi-Perspective Modeling for IT Risk Assessment. Information Systems Frontiers (2010) (forthcoming)
29. The Open Group. ArchiMate 1.0 Specification. Van Haren Pub., Zaltbommel (2011) (forthcoming)
30. Trienekens, J.J., Bouman, J.J., van der Zwan, M.: Specification of Service Level Agreements: Problems, Principles and Practices. Software Quality J. 12(1), 43–57 (2004)
31. Uebernickel, F., Bravo-Sanchez, C., Zarnekow, R., Brenner, W.: Is Service-Engineering: A Process Model for the Development of is Services. In: European and Mediterranean Conference on Information Systems, EMCIS (2006)
32. Vargo, S.L., Lusch, R.F.: The 4 Service Marketing Myths: Remnants of a Goods-Based, Manufacturing Model. J. of Service Research 6(4), 324–335 (2004)
33. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-Based Service Modeling and Design: Toward a Unified View of Services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 410–424. Springer, Heidelberg (2009)
34. Wieringa, R.: Requirements Engineering: Frameworks for Understanding. Wiley, Chichester (1996)
35. Zhang, L.-J., Zhou, N., Chee, Y.-M., Jalaldeen, A., Ponnalagu, K., Sindhgatta, R.R., Arsanjani, A., Bernardini, F.: SOMA-ME: A Platform for the Model-Driven Design of SOA Solutions. IBM Systems J. 47(3), 397–413 (2008)

Analysing Enterprise Models from a Fractal Organisation Perspective – Potentials and Limitations

Kurt Sandkuhl[1] and Marite Kirikova[2]

[1] University of Rostock, Department of Computer Science
Albert-Einstein-Str. 22, 18059 Rostock, Germany
Kurt.Sandkuhl@uni-rostock.de

[2] Riga University of Technology,
Kalku 1, Riga, LV-1658, Latvia
Marite.Kirikova@cs.rtu.lv

Abstract. The paper focuses on the use of the fractal paradigm in enterprise modeling. It investigates whether the properties of fractal organizations can be applied in business analysis and whether this results in useful outcomes and new insights. Based on an adaptation and operationalization of properties of fractal organizations, two real-world cases are analyzed using the adapted properties. The contributions of this paper are (1) to adapt fractal organisation properties for use in analysis of enterprise models, (2) to present practical examples from two cases showing the application of the fractal organisation properties, and (3) to identify potentials and limitations of using fractal organisation perspective in enterprise model analysis.

Keywords: Enterprise Modelling, Fractal Organisation, Practices, Business Analysis.

1 Introduction

Work presented in this paper brings together approaches and experiences from two different research fields in computer science; enterprise modelling and fractal organisations. In enterprise modelling, one of the traditional application purposes has been to understand the current situation in an enterprise or organisation under consideration in order to identify or explain business problems and to propose improvements [1]. Many methods, approaches, tools and work practices aiming at supporting this purpose were developed in areas such as business process reengineering [2], process improvement [3], enterprise knowledge modelling [4] organisational renewal [5], or and information systems development [4]. This large body of knowledge forms one basis for the work presented in this paper. In this context, our focus is on business analysis, i.e. on analysing enterprise models to identify organisational improvement potential.

High turbulence of business environment requires from organizations such features as agility [11], and viability [12]. Both of these features require means for achieving a good balance between complexity and simplicity in organizational management and operations. Fractal organizational structure is proposed by several researches, e.g. [6], [12], and [13] as an enabler of agility and vitality.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 193–207, 2011.

Fractal organisation structures have received much attention in areas like manufacturing industries or enterprise engineering. Among the advantages of fractals, their flexibility, robustness and easy adaptation to new business challenges are considered as interesting for many application domains. Does it make sense to apply fractal organisation principles when analysing businesses i.e. is it feasible to do this and does it give useful results pertinent to analysis purpose? What are the potential benefits and limitations of doing this?

The contribution of this paper is (1) to adapt fractal organisation properties for use in analysis of enterprise models, (2) to present practical examples from industrial cases showing the pertinence of the fractal organisation properties, and (3) to identify potentials and limitations of using the fractal organisation perspective in enterprise model analysis.

The paper is structured as follows: Section 2 will introduce the fractal paradigm as the background for our research. Section 3 investigates how the properties of fractal organisations can be interpreted in analysis of enterprise models, i.e. an adaptation of the properties of fractal organisation for enterprise modelling is proposed. In Section 4, we use the adapted properties proposed in section 3 for analysing real-world enterprise models in order to illustrate pertinence. Section 5 discusses the results of applying the fractal organisation perspective and identifies potentials and limitations of using this perspective. Section 6 summarizes the work and gives an outlook on future work.

2 Background of Fractal Paradigm

The start of the use of fractal paradigm in context of organizational structure and behaviour can be traced back to 1992 [13]. The overall number of research papers available on this topic is not very large, the number of books is 2 [12], [13]. In the first edition of [13] H. J. Warnecke suggests fractal organization of enterprise as essential means for survival in turbulent environment. In this work a fractal is defined as independently acting corporate entity whose goals and performance can be precisely described. H. J. Warnecke defines and describes the following basic properties of fractal organization:

- Self-similarity - fractals are self-similar, each performs services;
- Self-organization – fractals practice self organizations (1) operatively – procedures are optimally organized by applying suitable methods and (2) tactically and strategically – fractals determine and formulate their goals in dynamic process and decide upon internal and external contacts. Fractals restructure, regenerate and dissolve themselves.
- Goal orientation – the system of goals that arises from the goals of individual fractals is free from contradictions and must serve the objective of achieving corporate goals.
- Dynamics (in other sources named as dynamics and vitality [10]) – (1) the fractals are networked via an efficient information and communication system. They themselves determine the nature and extent of their access to data; (2) the performance of a fractal is subject to constant assessment and evaluation.

P. Hoverstadt's book [12], written 16 years later than [13] applies the fractal paradigm as the basis for use Viable Systems Model [14] in enterprise management.

In different researches dated from 1993 to 2011 the fractal paradigm has been applied in the following contexts:

- Manufacturing/product development [19], [20], [25], [26]
- Organizational networks [21], [22], [23], [26], [31]
- Service oriented systems [15], [30], [32]
- Agent oriented systems [28], [33]
- Business process and workflow management [25]
- Competence, responsibility, motivation, and goal management [16], [17], [18], [24], [27], [31]
- Knowledge and decision making management [21], [24]
- Quality control [19], [28]
- Enterprise architecture [29]
- Information systems and software engineering (a survey of related work in information systems development is included in [34]), [35]

The above list of topics shows that many organizational aspects are researched from the point of view of the fractal paradigm; however there is no analysis on applicability of fractal paradigm in enterprise modelling practice. Usefulness of the paradigm in each of the above mentioned enterprise perspectives suggests that use of the paradigm could bring particular benefits in enterprise modelling. The fractal paradigm does not focus on isolated aspects of an organisation, but defines properties cross-cutting processes, organisation structures and information flows, to name just a few examples. Such a multi-perspective approach including different aspects is also essential for enterprise modelling, since in enterprise modelling it is important to understand dependencies and connections between different organisational aspects (e.g. processes, organisation structure, competences or enterprise architecture), because organisational changes or problems usually affect several aspects.

3 Fractal Organisation Properties for Enterprise Model Analysis

Based on the properties and characteristics of fractal organisations described in section 2, this section investigates the adaptation of these properties for use in business analysis. The assumption made for work presented in this paper is that the "as is" situation in an enterprise already has been captured and documented in a model, i.e. we focus on analysis of models rather than on developing them. In this context, adaptation of properties includes two aspects:

- How to interpret the property in the context of business analysis?
- How to operationalize this interpretation for practical use, i.e. what questions to investigate in an actual analysis case and how to perform the analysis?

In the remaining part of this section, we will discuss the properties of fractal organisations from section 2 with their interpretation in business analysis and the operationalization. The focus will be on self-similarity, goal orientation, self-organisation, and dynamics.

Self-similarity

As discussed in section 2, self-similarity is a repetition of a particular pattern of organization structure at different scales of a particular organizational dimension or at different scales of several organizational dimensions simultaneously. If such a pattern exists in some organization unit of the same scale, but not in all units of this scale, the reason for this should be investigated. Are pattern-compatible units performing better or worse than non-pattern units? Should there be an adaptation towards the pattern of the non-pattern structure?

Based on the above interpretation of the property, the proposed *operationalization* includes the following questions to be investigated during business analysis:

SS-1: Do organisation patterns repeat on different scales of the organisation in the dimension "organisation structure"? If so, does the repeating pattern has advantages compared to other structures and should be implemented on all scales?
SS-2, SS-3, and SS-4: Same as SS-1, but for product structure (SS-2), process structure (SS-3), resources structure (SS-4).

This property also includes that information should no longer be monopolized, but be made generally available. In practice support for this aspect has to be established in the enterprise architecture and implemented by information systems in the enterprise. More obvious solutions would include access to essential information from all organisational functions. Enterprise models include relations from functions to information systems, i.e. we could use these relations as indicators.

Operationalization (continued):
SS-5: Is the information system structure included in the model? If so, do all organisation units have access to essential information systems?

Self-organisation

Self-organisation includes that fractals restructure, regenerate and dissolve themselves based on goals in a dynamic process and internal and external contacts. If the model of the enterprise under consideration indicates delegation of decision rights regarding enterprise strategy implementation to organization units, these units should have continuous improvement and adaptation processes in place and it also has to include adaptation. Furthermore, there should be organizational roles responsible for this task.

Operationalization:
SO-1: Is delegation of responsibilities in the organization reflected in the model? If so, do continuous improvement and adaptation processes exist?
SO-2: Have organizational roles responsible for continuous improvement been established?

In the literature on fractal organisation, so called "process patterns" characterizing fractal organisations are described. An example is the pattern described by [10], which identifies monitoring, analyzing, reporting, planning, executing as the main functions in project-oriented fractal companies. Such process patterns for fractal organisations could serve as templates for analysing models in order to detect self-organisation structures.

Operationalization (continued):
SO-3: Can all processes from the process pattern be found in the enterprise model under consideration? If so, can additional properties of fractal organisations be identified?
SO-4: Can the majority of the sub-processes of the process pattern be found in the model? If so, are the missing processes – if not named in a different way - the starting point for improving the organisation into this direction?

Goal Orientation

Goal orientation includes different aspects, like that the goals of individual fractals are free of contradictions from the goals of the overall organization, serve the objective of achieving corporate goals and involve all units concerned. Goal orientation as a property does not need adaptation for use in business analysis, since even non-fractal organisations should follow this principle. However, to determine whether this property is implemented is not a trivial task if only the enterprise models are available. It would basically require propagation of the goals to different organisational division and the refinement of relevant goals for each organisational unit

Operationalization:
GO-1: Are the enterprise's goals included in the enterprise model? If so, are the goals broken down for use in different organisational units? Should this be done in order to reach acceptance, as it is recommended in a balanced scorecard with its sub-scorecards?

Dynamics and Vitality

The operationalization of this property overlaps with *SO-1*, *SO-2*, SS-5, and *GO1*. Additionally fractals with identical goals and input and output variables can have quite different internal structures. If organization units with identical input/outputs and goals exist, is one of the internal structures superior to the others and should it be adapted by the others?

Operationalization:
DV-1: Do different processes, activities, or tasks with identical input/output variables exist in the model? If so then if they have different internal structures, is there a superior one performing better?

The above operationalization of self-similarity, goal orientation, self-organisation, and dynamics focuses on the questions to ask in business analysis. Furthermore, we also need to define, how to use these questions in the actual business analysis. As discussed in Section 3, we assume that an enterprise model exists capturing the "as is" situation. Although there are different ways how to use the questions for analysing an enterprise model, we recommend the following procedure and use it in the cases presented in Section 4: The business analysis should be performed by a team consisting of at least one expert in fractal organisation properties familiar with the questions and one enterprise modelling expert familiar with the model under consideration. The analysis should start with an introduction of scope and purpose of the model, the method and notation used, and a walk-through of the actual model, which starts with an overview to the main processes and structures, and includes at least one part of the model in full detail. Afterwards, the questions are analysed as follows:

- SS-1 to SS-4: the analysis team jointly browses different perspectives (process, organization, product, resources) of the enterprise model on different levels. Similarities are documented as pattern candidates. After finishing the browsing, all candidates are revisited in order to decide whether they are patterns.
- SO-1, SO-2 and GO-1: the analysis team checks the meta-model for entity types or relationships types matching the wanted ones (i.e. role, delegation, goal). For the matching entity types, the instances are browsed to answer the questions.
- SO-3 and SO-4: the analysis team browses the instances of the process-related entity-types in the model for instance names matching the names of the sub-processes in the process patterns.
- DV-1: same procedure as for SS-1 to SS-4, but limited to those parts of the model containing processes, activities or tasks.

4 Application of the Adapted Properties in Real-World Cases

This section focuses on evaluating the adapted properties introduced in section 3 regarding the applicability and pertinence by applying them in two cases. One case is taken from the public sector and the other one from industry. Section 4.1 presents and discusses the application of the adapted property in the public authority case; section 4.2 covers the industrial case. An interpretation of the results and comparison of the cases is included in section 5.

4.1 Public Authority Case

The Public Authority case model was developed in 2005. Its purpose was to establish a vision of new information systems of a university. Currently the information system has been developed. Its functionality is taken into consideration in the enterprise model analysis. The enterprise model was developed using Enterprise Knowledge Development methodology [7]. The model consists of five sub-models reflecting perspectives of Goals, Processes, Actors and Resources, Concepts, and Information Systems Components and Requirements.

The results of analysis of the model are presented according to the four properties of fractal organizations and their operationalization introduced in Section 3.

Self-similarity

SS-1: Do organisation patterns repeat on different scales of the organisation in the dimension “organisation structure”? If so, does the repeating pattern has advantages compared to other structures and should it be implemented on all scales?

The pattern at different scales was identified in Actors and Resources sub-model. It was Unit level *i*, Management of Unit level *i*, binary relationship between Unit level *i* and Management of Unit level *i*. The levels in decreasing order of granularity were University, Faculty, Institute, and Department. A representative of University IT department participating in the model analysis session admitted that clear repetition of the same pattern at all scales would give an opportunity to state clearer and richer information systems requirements.

SS-2: Do structural patterns repeat on different scales of the organisation in the dimension "product structure"?

Product structure was not represented in the model.

SS-3: Do process patterns repeat on different scales of the organisation in the dimension "process structure"? If so, does the repeating pattern has advantages compared to other structures and should it be implemented on all scales?

In the business process model which was developed at a high level of abstraction common patterns where not found except of some similarities with respect to maintenance of different Information systems registers related to each level of organizational structure. Relationships between the registers were not specified.

SS-4: Do structure patterns repeat on different scales of the organisation in the dimension "resource structure"?

Resources (excluding organizational structure) were represented at a high level of detail and did not contain repeating patterns.

SS-5: Is the information system structure included in the model? If so, do all organisation units have access to essential information systems?

Information systems structure was partly represented in Concepts sub-model. Indirect relationships to all organizational units were identified in the model.

Self-organisation

SO-1: Is delegation of responsibilities in the organization reflected in the model?

Model showed only which information is received or transferred by which actor. Delegation of responsibilities was not represented.

SO-2: Have organizational roles responsible for continuous improvement been established?

The activity of continuous improvement was identified in the model, but was not assigned to any actor. This can be explained with the fact that in 2005 there was no specific department responsible for continuous improvement. Some years later a University Strategy Department was established which partly deals with issues of continuous improvement. An interesting fact is that this department has its own information system that operates on different principles than other parts of the information system and causes lots of problems for users due to request of inputs that do not fit the work structure. The Strategy Department's system was outsourced from another public institution. The difficulties currently perceived emphasize that it is necessary to represent who and how is responsible for continuous improvement in the enterprise model to avoid problems that can arise if this issue is not considered.

SO-3: Can all processes from the process pattern be found in the enterprise model under consideration?

Planning process was missing.

SO-4: Can the majority of the sub-processes of the process pattern be found in the model?

Processes of the process pattern (except of planning) were represented in the model, however they did not form fractal structure.

Goal Orientation

GO-1: Are the enterprise's goals included in the enterprise model? If so, are the goals broken down for use in different organisational units? Should this be done in order to reach acceptance?

The models included a sub-model dedicated to the goals of the university; however, the goals were not broken down for use in different organizational units. Particular goals for units at all levels would be beneficial so as to achieve clearer and richer requirements and reach user acceptance for information systems solutions.

Dynamics and Vitality

DV-1: Do different processes, activities, or tasks with identical input/output variables exist in the model? If so, then if they have different internal structures, is there a superior one performing better?

On the highest level of abstraction it was possible to identify such inputs and outputs for processes related to different registers. These processes were not shown in detail and thus it was not possible to analyze their internal structure. In another project related to university information system [35] such structures were identified and recommended for implemention at several branches of the fractal organization.

4.2 Industrial Case

The industrial model analyzed was developed in the EU-FP6-project MAPPER [9] during 2006 - 2008. The model was produced in the automotive supplier use case of MAPPER and focused on the department of advanced engineering and the 'process of innovation', which basically involves several organizational functions in developing new technologies or components for future products of the company. The method used was C3S3P [5, 6]. Two model versions were available: the model of the 'as is' situation, also called scoping model, and the model slowing the 'to be' situation, also called solution model.

The models covered the POPS* perspectives [8], which include the enterprise's processes (P), the organization structure (O), the product developed (P), the IT system used (S) and other aspects deemed relevant when modeling (*). The analysis of the models from MAPPER is presented in the following and divided according to the four properties and their operationalization introduced in Section 3.

Self-similarity

SS-1: Do organisation patterns repeat on different scales of the organisation in the dimension "organisation structure"?

Solution model and scoping model included only the advanced engineering department in full detail. Other organisation structures were not included. Thus, similarity between organisation units could not be detected.

SS-2: Do structural patterns repeat on different scales of the organisation in the dimension "product structure"?

The scoping did not contain any product structure, but it was decided to include the product decomposition structure, requirements, technical characteristics, materials, etc. in the solutions model. Structural similarities between different levels of the product structure were detected, but these similarities were based only on common characteristics of different component types. Further unification of the structures on different levels wouldn't make sense since the different kinds of product components (sensors, harness, wires, etc.) have significantly different structures and features.

SS-3: Do process patterns repeat on different scales of the organisation in the dimension "process structure"? If so, does the repeating pattern has advantages compared to other structures and should it be implemented on all scales?

Regarding the work processes, the scoping model did not contain any similarities between different levels, which were not surprising since the processes were only very roughly defined without details and refinements. In the solution model, similarity exists on two levels. First, all tasks aiming at "establishing" a specification start with "preparing" the task and continue by "developing an initial draft", "establishing test methods" and the "final specification". The second similarity was detected on a refinement level. Wherever a material, a test method or a design approach had to be selected, first the existing ones were checked for suitability, then the decision was made whether to develop a new one in-house or to outsource this.

SS-4: Do structure patterns repeat on different scales of the organisation in the dimension "resource structure"?

The only resources included in the scoping model were the major IT systems used in the enterprise. This was primarily to identify the systems and did not show their internal structure or relationships. The solution model included more structure and refinements of this aspect, but similarity on different scales was not detected.

SS-5: Is the information system structure included in the model? If so, do all organisation units have access to essential information systems?

To improve and promote access to information and knowledge, and sharing of such information was one of the main intentions of the project. Thus, the solution model contains many details of which view of what information is needed for what role in what task. All roles and all organisation units have access to essential information. This property of a fractal organisation is clearly visible in the model.

Self-organisation

SO-1: Is delegation of responsibilities in the organization reflected in the model? and SO-2: Have organizational roles responsible for continuous improvement been established?

Mechanisms and structures for continuous improvement and adaptation of the organisation structure were outside the scope of the modelling. The model focused on the core process "Process of Innovation" of one organisation unit "advanced engineering" but did not cover the whole organisation unit or the whole organisation.

SO-3: Can all processes from the process pattern be found in the enterprise model under consideration? and *SO-4: Can the majority of the sub-processes of the process pattern be found in the model?*

Such structures are not part of the model. The model did not cover the whole organisation, nor the whole organisation unit. Whether the process pattern exists in the organisation or not cannot be concluded from the model.

Goal Orientation

GO-1: Are the enterprise's goals included in the enterprise model? If so, are the goals broken down for use in different organisational units? Should this be done in order to reach acceptance?

The models included a sub-model dedicated to the goals of the MAPPER project and the goals of the company for the POI. These goals were captured as goal hierarchy, the goals were linked to processes and organizational units related to goal fulfillment.

This goal orientation and the traceability of the goals were implemented in the solution model since a number of specific goals regarding knowledge sharing and innovation were very important for the company and should be addressed. In the scoping model, goals were not included and the specific goals not defined. Thus, the case is a good example for the importance of goal orientation in industry and a confirmation of the relevance of this property of fractal organizations.

Dynamics and Vitality

DV-1: Do different processes, activities, or tasks with identical input/output variables exist in the model? If so, then if they have different internal structures, is there a superior one performing better?

On a high aggregation level, many processes exist with identical input and output, since most processes of the POI have the product knowledge as input and the changed product knowledge as output. However, on a lower level, these identical inputs are no longer visible, since only different part of the product knowledge is used.

5 Discussion

This section is dedicated to discussing the experience acquired in the two cases presented in section 4. The approach used for this discussion is to compare the results of analysing the enterprise models of the two cases with the results of the original business analysis performed in these cases, i.e., the analysis results of the enterprise models developed in these real-world cases are considered as "gold standard" when comparing the results of the analysis with the adapted properties. For both cases, a table will be presented summarizing two aspects of the evaluation:

- Applicability: Was it possible to apply the property and its operationalization in practice?
- Usefulness: Was the use of the properties and its operationalization of any help in business analysis?

Regarding *applicability* four categories will be distinguished:

- it was possible to apply the operationalization and the *property was detected*
- it was possible to apply the operationalization, but the *property was not detected*
- It remains *unclear*, whether the operationalization can be applied (e.g. if the model does not contain information needed for applying the property)
- it was *not feasible* to apply the operationalization

Regarding *usefulness*, five categories will be distinguished:

- *New insights*: using the property gave new insights and other opportunities to improve the organization as compared to the result of the original business analysis
- *Confirmed*: using the property resulted in confirmation of the results of the original business analysis, i.e. the same or compatible results were achieved
- *Contradiction*: using the property resulted in recommendations contradicting the results of the original business analysis.
- *Not relevant*: the property was not relevant for the business analysis
- *Not applicable*: the property could not be analyzed, e.g. due to missing data

Table 1 shows a summary of the analysis of the public authority case.

Table 1. Summary of public authority case

Property	**Applicable?**	**Usefulness**
SS-1	Property detected	New insights (how to organize work at lower levels of institutional granularity)
SS-2	Property not detected	Not applicable (property not detected, however the product perspective could be useful and give new insights with respect to information systems requirements)
SS-3	Property detected (to some extent)	New insights (more detailed analysis of these properties could help to identify particular requirements at different levels of fractal organization)
SS-4	Property not detected	Not applicable (property not detected)
SS-5	Property detected	New insights (direct instead of indirect relationships could give better chances for interface requirements definition)
SO-1	Not detected	Not applicable (aspect missing in the model except of information flows to and from actors)
SO-2	Property detected (to some extent)	There is a process for continuous improvement but there is no delegation: New insight: modelling the delegation is essential in case of continuous improvement.
SO-3	Not detected	Not applicable (property not detected)
SO-4	Property detected	New insight (related to *SO-2;* it is essential to represent planning process in an enterprise model
GO-1	Property detected (to some extent)	New insight: it is beneficial to propagate goals in organisation structure
DV-1	Property detected (to some extent)	Confirmation: in another research of university information system preferable process structures were found.

This summary shows that it was not possible to detect fractal properties *SS2*, *SS4*, *SO1*, and *SO3*. Some properties, namely, *SS3*, *SO2*, *GO1*, and *DV1*, were detected to some extent. Nevertheless, the results of analysis of fractal properties showed that utilization of them in enterprise models could provide richer models and more detailed information systems requirements.

Table 2. Summary of industrial case

Property	Applicable?	Usefulness
SS-1	Property not detected	Not applicable (property not detected)
SS-2	Property detected	Confirmation and contradiction: some part of the structures should repeat, but unification of product structures wouldn't make sense.
SS-3	Property detected	Confirmation: processes with similar structure
SS-4	Property not detected	Not applicable (property not detected)
SS-5	Property detected	Not applicable (property not detected)
SO-1 to SO-4	unclear	Not applicable (aspect missing in the model)
GO-1	Property detected	Confirmation: goals have to be propagated in organisation structure
DA-1	Property detected	Not applicable (aspect missing in the model)

Table 2 shows a summary of the analysis of the industrial case. This summary shows that it was possible to apply the operationalization of the properties in the industrial case. Only the properties for self-organization could not be utilized since the scope of the model was too narrow as it was limited to one department and one value creation process. Furthermore, most of the other properties were not detected in the model, indicating that the enterprise under investigation does not show many characteristics of a fractal organization.

However, those properties which were detected confirmed the results of the original business analysis, i.e. utilization of self-similarity for the process and product perspective (SS-2 and SS-3) and goal orientation (GO-1) are relevant and valuable for the enterprise under consideration. Self-Similarity in the product perspective also showed a case where it is important to see and respect limitations of self-similarity, since some levels of the product structure have similarities, but these similarities disappear with increasing specialization.

The presented research focused on a most common canonical list of the properties of fractal systems, namely, self-similarity at different levels of scale, self organization, goal-orientation, and dynamics and vitality [36]. In [37] there are other properties of fractal systems listed, such as emergence, co-evolution, sub-optimality, requisite variety, connectivity, and simple rules. Operationalization of these properties may generate some new questions besides the ones described in Section 3. However, further research is needed to see to what extent the analysis of these additional properties is possible and useful in the context of enterprise modelling.

6 Conclusions and Future Work

The aim of the paper was to investigate whether the properties of fractal organizations can be applied in business analysis and whether this results in useful outcomes and new insights. Based on an adaptation and operationalization of properties of fractal organizations, two real-world cases were analyzed using the adapted properties. The first case showed many properties of fractal organizations and led to new insights regarding further improvement potential of the organization. In the second case, the results were not as positive as in the first case, but still showed the applicability of the properties and importance of properties, like goal orientation and self-similarity.

Although the use of fractal organization properties in business analysis showed some promising results, the potential would probably be bigger if we started to use them already when capturing the "as is" situation in an enterprise. This could help to avoid certain shortcomings in analysis models, like missing delegation relations between roles or improvement processes, but the methods for enterprise modeling might have to be adapted. An example is the property of self-similarity of the processes in the industrial case: during development of the solution model in this case, the processes were designed separately from each other and only after several iterations of refinement and validation a similarity between them developed. This refinement process probably would have been shorter with the advantages of self-similarity in mind. One of the main issues in this context is the method support. Enterprise modeling methods like EKD or C3S3P do not include activities supporting the implementation of certain organization paradigms, like fractal organizations, and probably should not do so, since this would limit the applicability and the situational adaptability of the methods. However, for business analysis and process design activities, it might be beneficial to raise awareness on the analysis side for advantages of fractal organization forms and to offer additional method support, like a method component for fractal organization based process design.

The main limitation of the work is the small number of cases considered in the evaluation and the limited number of fractal organisation properties applied. Basically we can conclude that it is possible to apply our operationalization and that this was contributing to business analysis in these two cases, but we should not even try to generalize these results. Since the results from these first two cases are promising, future work will have to focus on identifying characteristics of cases or organisations, where the use of fractal organisation properties in business analysis can be recommended. More cases are required in order to reach conclusions on this question. Furthermore, the operationalization has to be subject to a more thorough quality check including (a) do we need to include more properties or a different interpretation of fractal organisation properties? and (b) can the operationalization be made more precise, complete and easier to apply? This aspect requires involvement of more practitioners.

Acknowledgment. We acknowledge Ilze Birzniece, Ligita Businska, Janis Drobisevskis, and Peteris Rudzajs for analysis of the of public authority enterprise model. We acknowledge the work in the MAPPER project for developing the industrial model.

References

1. Harmon, P.: The Scope and Evolution of Business Process Management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management. Springer, Heidelberg (2009)

2. Davenport, T.H.: Process Innovation. Reengineering Work Through Information Technology. Harvard Business School Press, Boston (1993)
3. Humphrey, W.S.: Software Process The improvement – A Personal View: How it Started and Where it is Going. Software Process The improvement and Practice 12, 223–227 (2007)
4. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises. Springer, Heidelberg (2008)
5. Warner, B.W.: Organisational Development - A Process of Learning and Changing. Addison Wesley, Reading (1994)
6. Sandkuhl, K., Lillehagen, F.: The Early Phases of Enterprise Knowledge Modeling: Practices and Experiences from Scaffolding and Scoping. In: Stirna, J., Persson, A. (eds.) IFIP PoEM 2008. LNBIP, vol. 15, pp. 1–14. Springer, Heidelberg (2008)
7. Bubenko, jr., J., Persson, A., Stirna, J.: EKD User Guide. Appendix B to Deliverable 3 of EU-funded project "HyperKnowledge" (#IST–2000-28401) (2001)
8. Lillehagen, F.: The Foundations of AKM Technology. In:10th International Conference on Concurrent Engineering (CE) Conference, Madeira, Portugal (2003)
9. Johnsen, S., Schümmer, T., Haake, J., Pawlak, A., Jørgensen, H., Sandkuhl, K., Stirna, J., Tellioglu, H., Jaccuci, G.: 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)
10. Canavesio, M.M., Martinez, E.: Enterprise modeling of a project-oriented fractal company for SMEs networking. Computers in Industry 58(8-9), 794–813 (2007)
11. Jordan, J.A.: Requirements for next generation manufacturing systems. In: Proceedings of the IEEE/CPMT International Manufacturing Technology Symposium, pp. 247–255. IEEE, Los Alamitos (1994)
12. Hoverstadt, P.: The Fractal Organization: Creating Sustainable Organization with the Viabable System Model. John Wiley & Sons, Chichester (2008)
13. Warnecke, H.J.: The Fractal Company: A Revolution in Corporate Culture. Springer, Heidelberg (1993)
14. Beer, S.: Platform for Change. John Wiley, Chichester (1978)
15. Zhao, Y., Wu, J., Shu, H.: The fractal management of SOA based services integration. In: Proceedings of International Conference on Information Management Innovation Management and Industrial Engineering, pp. 420–424. IEEE, Los Alamitos (2008)
16. Deng, X., Peng, J., Huang, H.: Research on the fractal company modelling based competence, pp. 2136–2140. IEEE, Los Alamitos (2009) 978-1-4244-3672-9/09
17. Klopp, M., Kibling, W., Splavack, M.: Fractal lines, Manufacturing and Engineering, pp. 82–85 (April 1997)
18. Li, H., And Li, X.: An analysis of the double module fractal structures of enterprises and its forming motivations and mechanisms. In: Proceedings of International Workshop on Chaos-Fractals Theories and Applications, pp. 422–425. IEEE, Los Alamitos (2009)
19. Qin, Y., Zhao, L., Yao, Y., Xu, D.: A study of inter-enterprise quality control function self-organization reconfiguration based fractal networks. In: Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, pp. 1733–1737. IEEE, Los Alamitos (2007)
20. Bin, H., Gangyan, L.: Configuration unit for product self-organizing configuration design and its application. In: Proceedings of Fifth International Conference on Natural Computing, pp. 468–472. IEEE, Los Alamitos (2009)
21. Xu, J., Chen, C.: Similarity in tacit knowledge of supply chain and interfaces, p. 5. IEEE, Los Alamitos (2010) 978-1-4244-7161-4/10

22. Fan, X., Chen, H.: Research on the self-organization model of fractal supply chain, pp. 1–4. IEEE, Los Alamitos (2008) 978-1-4244-2108-4/08
23. He, X.: Research on evaluation model for self-similarity of fractal supply chain. In: International Conference on Educational and Information Technology (ICEJT 2010), pp. VI-279–VI-281 (2010)
24. Lei, Z., Shouju, R., Zuzho, L.: Self-organization based DSS framework, pp. 609–614. IEEE, Los Alamitos (2000) 0-7803-6583-6/00
25. Strauss, R.E., Hummel, T.: The new industrial engineering revisited, information technology, business process reengineering, and lean management in self-organizing fractal company. In: Proceedings of 1995 Engineering Management Conference, pp. 287–292. IEEE, Los Alamitos (1995)
26. Danli, D., Leng, J., Hongyan, Z.: A ship manufacturing model for fractal-based green supply chain. In: Proceedings of MSIE 2011, pp. 22–25. IEEE, Los Alamitos (2011)
27. Mun, J., Shin, M., Jung, M.: Trust evaluation model for fractal-based virtual enterprises using goal achievement probability. In: Proceedings of the 2007, IEEE IEEM, pp. 1532–1536. IEEE, Los Alamitos (2007)
28. Xu, D., Zhao, L., Yao, Y.: Fractal and mobile agent-based inter-enterprise quality tracking and control (2008)
29. Malan, R., Bredemeyer, D.: The art of change: Fractal and emergant. Enterprise Architecture 13(5), 1–25 (2010)
30. Traverson, B.: Contract management in viewpoint specifications: Experiment in the Electrical Network and SOA domains. In: Proceedings of the 14th International Enterprise Distributed Object Computing Conference Workshops, pp. 361–367 (2010)
31. Mun, J., Shin, M., Wi, H., Lee, K., Jung, M.: A VO formation process of a project-based organization with trust evaluation using goal achievment probability. In: Proceedings of PICMET 2008, pp. 256–265 (2008)
32. Nayak, N., Nigam, A.: Modeling business services for implementing global business services delivery platforms. In: The Proceedings of the 9th International Conference on E-Commerce Technology and The 4th International Conference on Enterprise Computing, E-Commerce and E-Sevices (CEC-EEE 2007), 8 p. (2007)
33. Gustavsson, P.M., Planstedt, T.: The road towards multi-hypoyhesis intention simulation agents architecture- Fractal information fusion modeling. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (eds.) Proceedings of the 2005 Winter Simulation Confernece, pp. 2532–2541 (2005)
34. Kirikova, M.: Towards multifractal approach in IS development. In: Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (eds.) Information Systems development: Challenges in Practice, Theory and Education, vol. 1, pp. 295–306. Springer, Heidelberg (2009)
35. Stecjuka, J., Makna, J., Kirikova, M.: Best practices oriented business process operation and design. In: Proceedings of the 9th Workshop on Business Process Modeling, Development and Support Business Process Life-Cycle: Design, Deployment, Operation & Evaluation Proceedings of the BPMDS 2008 Workshop held in Conjunction with the CAiSE 2008 Conference Montpellier, France, June 16-17, pp. 171–184. CEUR (2008)
36. Ryu, K., Jung, M.: Fractal approach to managing intelligent enterprises. Creating Knowledge Based Organisations. In: Gupta, J.N.D., Sharma, S.K. (eds.), pp. 312–348. Idea Group Publishers, USA (2003)
37. Fryer, P., Ruis, J.: What are fractal systems: A brief description of complex adaptive and emerging systems (2006), http://www.fractal.org

Enterprise Modeling Practice in ICT-Enabled Process Change

Anniken Karlsen[1,2]

[1] Department of Engineering and Maritime Studies, Aalesund University College, Norway
[2] Department of Information Science and Media Studies, University of Bergen, Norway

Abstract. This paper presents and discusses findings from a study where the use of enterprise modeling has been empirically investigated in eight combined process change and information technology initiatives. Artifacts, guidelines and tools used in enterprise modeling practice are identified. We identify three types of barriers to enterprise modeling: Challenges, Resistance and Moderators. We compare the way the modeling activities are organized with modeling maturity of different groups of project stakeholders. Our results indicate that the distribution of modeling maturity between project stakeholders affects how the modeling activities are carried out.

Keywords: Enterprise modeling, modeling tools, modeling use, barriers to modeling.

1 Introduction

Enterprise modeling can be seen as the art of externalizing knowledge which adds value to the enterprise or needs to be shared, and are often, as done in the following, used as a catch-all title to describe the activity of modeling any pertinent aspect of an organization [1]. Enterprise modeling can be used to represent the structure, behavior, components and operations of a business entity to understand, (re)engineer, evaluate, optimize and control business operations and performance [5, 6].

There are many commercial tools which have come into the marketplace in recent years to assist with architecture visualization and modeling [10]. Persson and Stirna [14] emphasize that while much research has been done on developing enterprise modeling methods, research concerning enterprise modeling in practice has been more or less neglected by the research community. A similar situation can be seen within process modeling, which can be seen as a specialized field of enterprise modeling [20]. Sedera, Gable, Rosemann and Smyth [15] emphasize that while there has been much research on process modeling techniques and corresponding tools, there has been little empirical research into important factors of effective process modeling and post-hoc evaluation of process modeling success.

This paper presents findings from a multiple case study of enterprise modeling practice in ICT-enabled process change. The paper supplements another publication where it is shown that different types of modeling initiatives produce a broad variety

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 208–222, 2011.

of modeling benefits [21]. The paper provides insight and answers to the following research questions:

(1) How is the modeling process organized?
(2) How is participation and involvement in the modeling process?
(3) Which tools, languages and guidelines are used for modeling?
(4) Which artifacts are produced in each type of modeling initiative?
(5) What might influence the selected way of organizing the modeling process as for example workshops with oral participation or workshops with active participation?
(6) Are there any barriers to modeling to be identified?

In the following the paper explains the motivation for our inquiry in section 2. Thereafter follows section 3 explaining how the research project was designed and conducted together with a short description on how the collected research material was analyzed. Thereafter follows section 4 where the questions above are attended, by using the questions as subsection headings. In section 5, our findings are discussed. Finally, in section 6 limitations of our work are emphasized and further work suggested.

2 Motivation

Our research and publication are motivated by both the work of writers like Davenport [3] focusing on information technology as a crucial enabler of process innovation and researchers of modeling practice, here represented by a few:

Davies, Green, Rosemann, Indulska and Gallo [12] conducted a study of conceptual modeling practice using the aspects of conceptual modeling as defined by Wand and Weber [22] to guide their work. Davis et al [12] state that conceptual models are developed and used during the requirements analysis phase of information systems development. Through their study they found that the top six most frequently used modeling techniques and methods were ER diagramming, data flow diagramming, systems flowcharting, workflow modeling, UML, and structured charts. They also found that the highest ranked purposes for which modeling was undertaken were database design and management, business process documentation, business process improvement, and software development.

Persson [13] has described situational factors and their influence on adopting a participative approach in enterprise modeling practice. Through her study she came up with recommendations for use of enterprise modeling in information systems development, particularly in the requirements engineering stages of the development process.

Vernadat [23] has written a book advocating a systematic engineering approach for modeling, analyzing, designing and implementing enterprise systems. In the book a large set of knowledge on tools and methods to achieve business process reengineering and business integration is presented.

Eikebrokk, Iden, Olsen and Opdahl [16] have conducted a study giving insight into Norwegian model-supported process-change practice, focusing especially on process modeling. As part of their study they introduced an a priori process-modeling-practice

(PMP) model [17] and a revised PMP model [18]. Their analyses indicate that a combination of technological, social and organizational factors explain the outcome of model-based project change projects.

Motivated by the fact that little is known about enterprise modeling in practice and with an initial aim to test and further explicate the conceptualizations of the PMP model into another setting, our study was initiated to focus on enterprise modeling in ICT-enabled process change. ICT-enabled process change is a term that denotes the use of information and communication technology as an enabler to change the way organizations work, including changes to business processes to make them more efficient and timely and covering the provision of enhanced information to support better decision making [9]. The dual focus built into the term ICT-enabled process change made us, at the onset of our inquiry, expect that different types of enterprise models would be developed and/or used as part of the combined process change and information technology initiatives under study.

3 Research Method

Case research is beneficial in the study of 'why' and 'how' questions because these deal with operational links to be traced over time rather than with frequency or incidence [2]. With our overall research question stated as: 'How is EM used and how can it be used to support ICT-enabled process change in Norwegian companies?' it was decided that a multiple case study would serve our purposes. Yin [19] defines a case study as an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.

According to Miles and Huberman [6] highly inductive, loosely designed studies make good sense when experienced researchers have plenty of time and are exploring exotic cultures. On the other hand, Miles and Huberman [6] say, pointing to Wolcott [4], it is not possible to embark upon research without an idea of what one is looking for and it is also foolish not to make that quest explicit.

Looking into an area with little prior empirical research it was decided to develop *a research model for enterprise modeling practice*, building on categories and subcategories from the related field of process modeling practice incorporating additional aspects found in literature. In addition a pilot study was conducted to provide additional input to the model. By incorporating the PMP model into the research design, we had an additional opportunity to test and further explicate the PMP models conceptualizations into a new setting in accordance with suggestions found in Miles and Huberman [6]. The enterprise modeling practice research model is presented in Karlsen [7].

The enterprise modeling research model was built up of three main categories: Enterprise modeling (EM), Context and Outcome, where Context was defined as the setting of the project comprising organizational characteristics, project specific characteristics and project participant characteristics and Outcome was defined as the phenomena that follow or are caused by enterprise modeling, including attainment of purpose and the effect of enterprise modeling on the ICT-enabled process change project solution. The EM category, which is the focus of this paper, addresses both the development of new models and the additional usage of existing models in relation to the ICT-enabled process change project.

EM was further elaborated by the subcategories (1) Management support, (2) Modeling Guidelines, (3) Modeling tools, (4) Individual modeling or workshop, (5) Participation and involvement, (6) Resistance, (7) Modeling languages and (8) Modeling artifacts [7]. The work of Eikebrokk et al. [16, 18], Davenport [3], Sedera et al. [15] and Sommar [11], were used to motivate both these definitions and the expected outcomes of enterprise modeling.

Having designed a research model to focus and bound the collection of data, in accordance with Miles and Huberman [6], *we then conducted an exploratory /explanatory multiple case study* on combined process change and information technology initiatives.

We used the telephone and internet to search for relevant cases, and ended up with an inquiry of eight Norwegian cases, defined as a constellation of (1) a main organization or (2) a consulting company and/or an IT-vendor. The main organizations of these cases were related to the construction industry (case C1), the marine sector (cases C2 and C4), the maritime sector (cases C3 and C8), the offshore sector (case C5), a wholesaler within the food sector (case C6) and the banking sector (case C7).

To prepare for the case study an interview guide was developed, containing semi-structured open-ended questions based on the categories of the enterprise modeling research model that was developed in the initial stages of the project. A total of thirty informants were interviewed as part of our investigation, generating 40 hours of tape recordings: two 'expert informants', six informants at the pilot stage to underpin the research model and twenty-two informants related to our eight cases. In addition a rich variety of material was collected in the form of model prints, reports and historical material, as recommended by Yin [19]. Organizational information was additionally downloaded from the internet. We also visited the various companies and got demonstrations of the software solutions involved.

It was decided that a criteria for being included in the study was that the organizations should be "available and willing", in the sense of being available and willing to provide in-depth insight into enterprise modeling practice via interviews and supplemental information. The second selection criterion was that the respondents defined their projects as ICT-enabled process change.

We had no initial knowledge of the enterprises and their modeling practice at the onset of the inquiry. With such limitations on what to find we chose to use the term enterprise modeling in a broad sense to capture how the companies in fact used modeling; possibly by using both formalized and non-formalized languages, simple tools etc.

All interviews were transcribed and transferred into Nvivo 9, a computer-assisted qualitative data analysis software package, generating more than 500 pages of transcribed text together with links to all other types of material for analysis. Nvivo 9 provided opportunities to run a variety of built-in queries and helped in keeping track of all material collected by providing database facilities.

The research model that originally guided data collection was also used in the initial computer-assisted analysis by providing initial constructs on characteristics of context possibly influencing on the modeling process, constructs on characteristics of the modeling process and the outcome of modeling.

To guide the analysis we used "Qualitative analysis: An Expanded Sourcebook" by Miles and Huberman [6]. This book gives a thorough explanation of coding as analysis, where coding is described as tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study.

The coding process of the interview transcripts started by building a node tree in Nvivo 9 containing the initial constructs of the EMP research model. Then a read through of all interviews was conducted whereby passages of text in the interview transcripts were linked to the appropriate initial nodes. In this process text passages which did not fit the initial nodes were in vivo coded and later specified under new appropriate nodes after a process of revealing appropriate new constructs.

To increase the quality of the coding process the material was re-read to check that nothing important had been missed in the reading process. Missed text sequences were linked to existing or new nodes. Thereafter followed a process where all material linked to each node was controlled to ensure consistency between selected text and the node assigned. Thereafter followed a process where material connected to a particular node was challenged to see if it should be broken into sub-nodes. If a subclassification seemed appropriate the divide was done.

The coding process ended up with an array of different constructs representing the findings done in conjunction with the questions we raised, concerning characteristics of context possibly influencing on the modeling process, characteristics of the modeling process and outcomes of modeling.

4 Modeling Practice

Our initial analysis focused on the case distribution of our eight cases among different constellations of ICT initiatives, process change main focus and modeling objectives.

This analysis led to the identification of five different types of enterprise modeling initiatives in our study which we called Strategy, Industry, Dataflow, Work and Support [21].

The 'Strategy' initiative (S) was identified and defined as modeling to reach a change strategy in a long term business change initiative with a mixed focus on improving work practice via physical intervention and improving information flows via IT. With reference to the tables and figures in this paper, case C1, C2 and C3 apply to this type of modeling initiative. The 'Industry' initiative (I) was identified and defined as modeling to reveal the build-up of market leaders' IT solutions to develop a joint industry specific IT solution and modeling as input to a preliminary report to communicate the necessary alignment between this joint solution and specific actor needs. With reference to the tables and figures in this paper, case C8 applies to this type of modeling initiative. The 'Dataflow' initiative (D) was identified and defined as modeling to reveal AS-IS and as input to a requirement specification in a change effort to improve information flows. With reference to the tables and figures in this paper, case C4 and C5 apply to this type of modeling initiative. The 'Work' initiative (W) was identified and defined as utilizing vendor supplied models to reveal differences between a wearable voice-directed warehouse application system and the organization in a change effort to improve work practice by technology. With reference to the tables and figures in this paper, case C6 applies to this type of modeling initiative. The ‘Support’ initiative (Q) was identified and defined as modeling to fill a quality system with process descriptions based on a specific guideline, focusing on developing a business support environment where it is foreseen that in the long run shared common models of work practice will improve business processes. With reference to the tables and figures in this paper, case C7 applies to this type of modeling initiative.

Across our cases a broad variety of different benefits of enterprise modeling were identified in ICT-enabled process change [21]. We will now take a closer look into the characteristics of the enterprise modeling process of each case and type of enterprise modeling initiative under study, thereby answering the research questions used as subheadings in the following.

4.1 How is the Modeling Process Organized?

Analysis identifies different ways of organizing the various modeling activities as shown in Table 1.

Table 1. Individual modeling or workshop etc

	C1	C2	C3	C4	C5	C6	C7	C8
	S	S	S	D	D	W	Q	I
Workshop with oral participation	+	+	+	+	+			+
Workshop with active participation							+	
User forum							+	
Supply your input							+	
Group-based model use						+		
Individual modeling		+			+			+

At the type of modeling initiative level, Table 1, illustrates that in the Strategy (S), in the Dataflow (D) and Industry (I) modeling initiatives, modeling activities were organized as workshops with oral participation, meaning that the modeling was written down by an external consultant, whilst participants of the main organization provided oral inputs to the modeling process. This was not the case concerning the Work (W) or Support (Q) initiative.

In the Support initiative they chose to use workshops with active participation in the modeling activity, where employees did concrete mapping of business processes. In addition the quality system initiative was supplemented with the possibility for all employees in the bank to provide inputs to model layouts via a digital mailbox-system named “Supply your input”. The bank also organized a specific user forum where modelers from each business area were represented. The user forum made decisions whether specific process change suggestions collected via the “Supply your input” should be universally applied in the banks’ preferred process portfolio. If so, the corresponding process model in the quality system got changed.

Group based modeling was used in the Work initiative, where a group of representatives from the main organization and external representatives, compared vendor supplied models with what was going on in the warehouse building. Differences were subject to debate and lead to necessary tweaks between system and process layouts.

4.2 How is Participation and Involvement in the Modeling Process?

Comparing the cases further indicates that even though people are not directly involved in the actual drawing of the models, their participation and involvement are evaluated as satisfactory or very good in all cases.

4.3 Which Tools, Languages and Guidelines are Used for Modeling?

By analyzing our cases we identify a varied use of tools, languages and guidelines as illustrated in Table 2.

Table 2. Guidelines, Languages and Tools

Case	C1	C2	C3	C4	C5	C6	C7	C8
Type of modeling initiative	S	S	S	D	D	W	Q	I
Tools:								
The quality system application							+	
Word	+		+	+				
Visio		+					+	
PowerPoint		+			+			
Excel	+	+						
Guidelines:								
Had guidelines	+	+		+	+		+	
Had no concrete guidelines	+		+	+				+
Use of vendor supplied models						+		
Languages:								
Modeling language used		+					+	
Modeling language not used	+		+	+	+	+		+
No specific modeling tool is used		+	+	+		+	+	+

In the construction case, C1, Microsoft Excel and Word are used as the tools for modeling. In the Marine subcontractor case, C2, no specific modeling tool is used, but comes in a flavor of Excel, PowerPoint and Visio made models. In the Maritime subcontractor case, C3, it is stated that no specific modeling tools are used. A model example from the case shows a "rich picture" type of model made in Word. In the Marine laboratory case, C4, Word is identified as the common modeling tool used in the project. In the Off-shore subcontractor case, C5, PowerPoint is the tool chosen. C6 relates to the Work initiative where modeling is defined as utilizing vendor supplied models when implementing a standardized ICT system. In the Banking case, C7, the quality system application itself is used for modeling. In addition one can state that in general no specific modeling tools are used due to a highly varied practice in the bank across departments and project participants. In the Industry case C8 tool use is said to differ between enterprises adopting the industry specific enterprise resource planning solution.

Concerning guideline use, analysis reveals that this can vary along the time-axis of the project lifecycle and among project participants. In the Support initiative they had a common framework on how to build the quality system for modelers and facilitators. They also used external consultants in each business area to make sure that the modeling standard was followed. In cases C2 within Strategy and C4 and C5 within Dataflow, external consultants used a consultant variant modeling guideline in their work. In case C4 employees reported that before the consultant entered the company no concrete guidelines were used. In C1, process description from a similar enterprise was used as a template to set up a description of the company's own processes. But in general no specific modeling guidelines were used.

Concerning modeling language the majority of cases report that no specific language was used. In cases where modeling language were reported to be used, it turned out that they spoke about some sort of a "consultant variant".

4.4 Which Artifacts Are Produced in Each Type of Modeling Initiative?

With reference to Table 3, analysis shows that in all cases process descriptions are made as part of the process change process (except for the Work initiative where models are used). Technological models are developed in three cases: In C2 Use Cases are developed, in C4 database models are developed and in C5 a system draft evolves in parallel with the development of the process descriptions. In C8, marked with a *, technological models of different solutions were used years ago, when developing the joint industry-specific solution. The construction case, C1, is the only case where models from other sources, textbooks and downloaded documents from the Internet, were adapted to be used as part of the process change process. In one example the consultants adapted a model from a textbook to illustrate to employees in the main organization what they meant by a holistic enterprise understanding.

Table 3. Artifacts

Case	C1	C2	C3	C4	C5	C6	C7	C8
Type of modeling initiative	S	S	S	D	D	W	Q	I
Process descriptions	+	+	+	+	+		+	+
Meta models							+	
Organization charts		+						
Technological models		+		+	+			*
Adapted models from text books and other sources	+							

4.5 What Might Influence the Selected Way of Organizing the Modeling Process as for Example Workshops with Oral Participation or Workshops with Active Participation?

By comparing the respondents' answers on modeling maturity, the main organization's and the externals' modeling capability and experience of modeling, analysis indicates that in most cases the externals' capability of modeling is seen and reported as high, or at least much higher than what is the situation in the main organization. In one case, C6, the capability and experience with modeling is reported as low both in the main organization and among the externals. In C7 modeling capability is reported as generally high in the main organization, but that it of course varies. In C4 and C8 the capability of modeling in the main organization in general are seen as low, but that there are persons that have some modeling experience from previous projects. By combining these findings with the organization of modeling activities in terms of using workshops with oral participation or workshops with active participation etc., the relationships are revealed as shown in Figure 1.

			Organizational modeling maturity:			
	External			Internal		
Organization of modeling activites:	High	Low	High	Low	Medium_Low	Variable
User forum	C7		C7			C7
Group-based model use		C6		C6		
Individual modeling	C2, C5, C8			C2, C5, C8	C8	
Supply your input	C7		C7			C7
Workshop with active participation	C7		C7			C7
Workshop with oral participation	C1, C2, C3, C4, C5, C8			C1, C2, C3, C4, C5, C8	C4, C8	

Fig. 1. Modeling maturity versus organization of modeling activities

The matrix indicates the following relationships between the ways of organizing the modeling activities related to the modeling maturity of different project participants:

(1) In cases where the modeling maturity of the external representative is reported as high and the modeling maturity level of the main organization as low or medium to low, workshops with oral participation are used to organize the modeling efforts. This way of organizing the modeling activities is in some cases supplemented with individual modeling, whereby the external representative sits down and do modeling by him-self based on interview inputs.

(2) In the case where modeling maturity is reported as high both in the main organization and among the external participant, workshops with active participation are used.

(3) In the case where the modeling maturity level is reported as low both in the main organization and among the external participant, group-based model use is applied. In this instance lack of knowledge on modeling does not stop the participants from finding vendor supplied models useful in the project.

4.6 Are There Any Barriers to Modeling to be Identified?

In the initial research model ‘Resistance’ was one of the sub-categories of the enterprise modeling process. Analysis reveals that there are in fact different types of barriers to modeling which we have grouped into: (1) *Challenges*, (2) *Moderators* and (3) *Resistance*. We identify and define ‘Challenges’ as barriers to modeling related to the actual act of model making. ‘Resistance’ is identified and defined as negative feelings associated with modeling. ‘Moderators’ is identified and defined as barriers to modeling that hinder the actual use of modeling in ICT-enabled process change.

Analysis shows the distribution of challenges, moderators and resistance among our cases and different types of modeling initiatives, as illustrated in Table 4, Table 5 and Table 6.

Table 4. Modeling challenges

Case	C1	C2	C3	C4	C5	C6	C7	C8
Type of modeling initiative	S	S	S	D	D	W	Q	I
Conceptual problem related to understanding graphical images							+	

As can be seen from Table 4, Support (Q) is the only initiative where *challenges* associated with understanding graphical images due to conceptual problems is reported.

Concerning the *moderators* of modeling, Table 5, analysis indicates that project participant characteristics, project specific issues, IT system issues, information issues and resource issues influence the modeling process. This is done by moderating, restricting or reducing, the modeling process in the different cases.

Project participant characteristics moderating the modeling initiative: Case 6 is the only instance where moderators associated with project participant characteristics are identified. In general, the main organization in this case works close with one specific IT specialist, serving their general needs for IT services. In this case the use of vendor supplied models is reported as a special event, a specific type of modeling initiative, in the everlasting improvement project where the IT service provider and the main organization live in what is called a symbiotic relationship. In general the IT service provider sees itself as well-informed about their customer and therefore sees little need for making models. The IT provider also pinpoints that a more directly focused work approach reduces the use of models in general. On the other hand, the situation of introducing “voice direction” was something new for all parties, and the vendor supplied models came in handy when the IT provider worked to adapt the organization to the way the system demanded and vice versa.

Project specific issues moderating the modeling initiative: In C6 the IT-provider do see the usefulness of modeling in some situations but emphasizes that in this case the history of the project is important and explains the reduced need for modeling in their day-to-day improvement work with the main organization.

Table 5. Modeling moderators

Case	C1	C2	C3	C4	C5	C6	C7	C8
Type of modeling initiative	S	S	S	D	D	W	Q	I
Project participant characteristics:								
Not being good at modeling reduces model making						+		
Knowledge of customers reduces the need for modeling						+		
Lack of historically good experiences with modeling reduces the modeling activity						+		
Some customers are not willing to spend time modeling						+		
Not being good enough to demand spending more time on planning reduces the modeling activity						+		
The fact that we are more directly focused reduces the use of modeling						+		
Project specific issues:								
The history of the project						+		
IT system issues:								
The desire to follow the sheep with the bell with respect to the ICT-solution reduces the need for process mapping								+
The IT system lays down guidelines for the modeling process								+
Information issues:								
Everything cannot be specified (like building a boat)								+
All information needs are not covered by process descriptions		+						
Resource specific issues:								
Available staff:								
Day to day activities are not designed for modeling work		+						+
Low staffing levels acts as a limiting factor	+	+						+
Money:								
Our level of ambition	+							
Bad economy acts as a limiting factor	+							
Costs associated with modeling	+		+					
Resource related reviews	+							+
Time:								
Time acts as a limiting factor		+					+	+

In general, concerning other customers, the IT-provider links reduced use of modeling to instances where customers are unwilling to pay time on modeling, and instances where they as an IT-provider is not "good enough" on demanding such spending.

IT-system issues moderating the modeling initiative: IT system issues are related to case 8, the type of modeling initiative where an industry specific solution is developed and implemented. In this case it is stated that the desire to follow the sheep with the bell, the leading organization in the industry, reduces the need for modeling. The reason is that the industry leader has been markedly engaged in developing the industry specific solution, so their processes are somehow embedded in the IT-solution. It is realized that by implementing the industry specific IT-solution one at the same time adopts the embedded business processes of a marked leader.

Information issues moderating the modeling initiative: Two cases report that their modeling initiative is moderated by information issues, C2 and C8. In C2 it is emphasized that all information needs are not covered by process descriptions and in C8, the case from the maritime sector, it is reported that everything cannot be specified, for example "building a boat".

Resource specific issues moderating the modeling initiative: As can be seen from Table 5 both the Strategy initiative, the Support initiative and the Industry initiative report on lack of resources as a limiting factor on modeling practice.

Table 6. Resistance

Case	C1	C2	C3	C4	C5	C6	C7	C8
Type of modeling initiative	S	S	S	D	D	W	Q	I
Yes, resistance present:								
Some people consider modeling high raving and theoretical							+	
Yes, because our job is to build boats								+
Yes, but the resistance has decreased:								
It was changed when they saw the system in practice						+		
Needed to see the point first	+							
Requires a sales job internally to avoid resistance							+	
The resistance changes from high to low	+							
You need to model a while before people see the point							+	
No resistance:								
Experienced no resistance		+*	+	+	+			

Concerning *resistance*, Table 6 shows that in four out of eight cases no resistance to modeling is experienced. In three of the cases resistance is experienced but has decreased. The reasons why resistance has changed can be seen directly from the table. The only case reporting on an ongoing negative feeling towards modeling is in B8 case, where it is stated that this is linked to what is their job focus; to build boats.

5 Discussion

Comparing our findings with the initial research model leads to an enriched picture of enterprise modeling practice. Concerning our question on how the modeling process is organized, our analysis shows that the EMP research model's category "Individual modeling or workshop" [7] should be more fine-grained to include the following constructs: Workshop with oral participation, Workshop with active participation, User forum, Supply your input, Group-based model use and Individual modeling.

Based on our analysis in section 4.5 on what might influence the selected way of organizing the modeling activities, as for example workshops with oral participation or workshops with active participation, we propose that the distribution of modeling maturity of project stakeholders influence the way the modeling activities are organized.

Concerning our question on participation and involvement in the modeling process, analysis shows that even though people are not directly involved in the actual drawing of the models, their participation and involvement are evaluated as satisfactory or very good. The key to these perceptions might be understood by the reported outcomes of modeling, where modeling is seen as an awareness-raising process in itself, as a communication tool or a thinking tool among others [21].

In [21] a broad variety of different benefits of enterprise modeling associated with the five types of modeling initiatives in our empirical investigation are reported. Looking into the artifacts made and the tools, languages and guidelines used for modeling, our inquiry indicates extensive use of the Microsoft Office application as a modeling tool across cases. In general no specific modeling guidelines are used, except for instances where one finds some sort of “consultant variant” guideline. Concerning modeling language the majority of cases report that no specific language is used. In cases where modeling language are reported to be used, it turns out that they again speak about some sort of a “consultant variant”. Concerning which artifacts are produced in each type of modeling initiative our analysis shows that in all cases process descriptions are made as part of the process change process. This finding is not surprising since we have investigated cases which by the interview objects have been understood and defined as ICT-enabled process change.

Comparing the tools, guidelines and languages used for modeling in our study with the modeling benefits produced, we conclude that even the simplest modeling tools and the simplest non-standard model-layouts can provide great value to project participants. The quality system initiative is the only instance where challenges associated with understanding graphical images due to conceptual problems is reported. In addition this is the only instance where it is reported that some see modeling as high raving and theoretical. Modeling to fill a quality system with models to be shared across time and space seem to raise the need for expressing models in a shared syntax [21], as opposed to other cases where models are made as part of a communication process which gives them their immediate meaning.

Despite the various benefits associated with modeling, analysis also reveal three types of barriers to modeling: (1) *Challenges*, (2) *Moderators* and (3) *Resistance*. This finding leads to the necessity to adjust the initial research model which only operated with Resistance as a subcategory. An interesting aspect revealed in the study was the saying that:

"*If you can tie modeling up against initial resistance, modeling actually helps because we can more easily see what the problems are*." [2. Interview, C3]

In this circumstance reduced resistance becomes an outcome or benefit of enterprise modeling.

6 Limitations and Further Work

Concerning our findings it must be emphasized that our qualitative study has aimed at painting a rich picture of enterprise modeling by investigating modeling practice in depth and within its real-life context. In an attempt to deal with well-known difficulties of case studies we have tried to focus and bound the collection of data by building and using an enterprise modeling research practice model and by applying an interview guide in the field in accordance with recommendations found in Yin and Miles & Huberman [6]. In general our study still is subject to various threats and limitations familiar to case-study research as described in Yin [19] and interviewing as discussed in Kvale [24] who states that the interview is neither an objective nor a subjective method. Focusing on gaining in depth insight from a few Norwegian cases has for instance limited our possibility to make large generalizations. Drawing heavily on related domains or what can be seen as specialized fields of enterprise modeling can be problematic due to context differences when designing a study. In addition, having focused especially on the use of enterprise modeling in ICT-enabled process change has led to a predominance of process modeling in the cases under study. This might be seen as problematic in relation to those who use a more restricted version of the enterprise modeling term: taking a more "total systems" approach, like Fraser [1] discusses.

As a next step in further work we suggest that a revision of the initial enterprise modeling practice research model is in demand, based on the findings of our empirical inquiry; some of which has been the subject of this paper. We also see the need for this revised model to be tested out in situations where projects use enterprise modeling from a more holistic approach than what has been practice in our cases. To increase the ability to make large generalizations we also see the need for large surveys. In such studies we hope our findings can provide useful inputs.

References

1. Fraser, J.: Managing Change through Enterprise models. In: Milne, R., Montgomery, A. (eds.) Applications and Innovations in Expert Systems II. SGES Publications (1994)
2. Benbasat, I., Goldstein, K.D., Mead, M.: The case research strategy in studies of information systems. In: Myers, M.D., Avison, D. (eds.) Qualitative research in information systems: a reader. Sage, London (2002)
3. Davenport, T.: Process innovation: reengineering work through information technology. Harvard Business School Press, Boston (1993)
4. Wolcott, H.F.: Differing styles on on-site research, or, "If it isnt't ethnography, what is it?". The review Journal of Philosophy and Social Science 7(1&2), 154–169 (1982)
5. Vernadat, F.B.: Enterprise Modeling and Integrations, principles and applications. Chapman & Hall, London (1996)
6. Miles, M.B., Huberman, A.M.: Qualitative data analysis – an expanded sourcebook. Sage Publications, Thousand Oaks (1994)
7. Karlsen, A.: A Research Model for Enterprise Modeling in ICT-enabled Process Change. In: Stirna, J., Persson, A. (eds.) PoEM 2008. LNBIB, vol. 15, pp. 217–230. Springer, Heidelberg (2008)
8. Petrie jr, C.J. (ed.): Proceedings of the First International Conference on Enterprise Integration Modeling. Scientific and Engineering Computation Series. The MIT Press, US (1992)

9. Manwani, S.: IT-Enabled Business Change, Successful Management. The British Computer Society, Publishing and Information Products, UK (2008)
10. Nightingale D. J., Rhodes, D. H.: Enterprise Systems Architecting: Emerging Art and Science within Engineering Systems, MIT Engineering Systems Symposium, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA (2004)
11. Sommar, R.: Business process modeling introduction, tutorial, developed by KTH (2006)
12. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58(3), 358–380 (2006)
13. Persson, A.: Enterprise modeling in practice: Situational factors and their influence on adopting a participative approach. Ph.D. Thesis. Department of Computer and Systems Sicences, Stockholm University/Royal Institute of Technology, Sweden, Report series No. 01-020 (2001)
14. Persson, A., Stirna, J.: Why Enterprise Modelling? An Explorative Study into Current Practice. In: Dittrich, K.R., Geppert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 465–468. Springer, Heidelberg (2001)
15. Sedera, W., Gable, G., Rosemann, M., Smyth, R.: A success modell for business process modeling: findings from a multiple case study. In: Proceedings Eight Pacific Asia Conference on Information Systems, Shanghai, China, pp. 485–498 (2004)
16. Eikebrokk, T., Iden, J., Olsen, D., Opdahl, A.: Process Modelling Practice: Theory Formulation and Preliminary Results. NOKOBIT, Molde (2006)
17. Iden, J., Olsen, D., Eikebrokk, T., Opdahl, A.L.: Process change projects: a study of Norwegian practice. In: Proceedings of ECIS, Gotenburg, Sweden, pp. 1671–1682 (2006)
18. Eikebrokk, T.R., Iden, J., Olsen, D., Opdahl, A.L.: Exploring Process-Modelling Practice: Towards a Conceptual Model. In: Proceedings of the 41st Hawaii International Conference on System Sciences (2008)
19. Yin, R.K.: Case study research: Design and methods, 4th edn. Applied Social Research Methods Series, vol. 5. Sage, Thousand Oaks (2009)
20. Andersen, B.: Enterprise Modeling for Business Process Improvement. In: Rolstadås, A., Andersen, B. (eds.) Enterprise Modeling: Improving Global Industrial Competitiveness. Ch. 10, pp. 137–157. Kluwer Academic Publishers, USA (2000)
21. Karlsen, A., Opdahl, A.L.: Benefits of different types of enterprise modeling initiatives in ICT-enabled process change. Manuscript submitted for publication
22. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modeling—A research agenda. Information Systems Research 13(4), 363–376 (2002)
23. Vernadat, F.B.: Enterprise Modeling and Integrations, principles and applications. Chapman & Hall, London (1996)
24. Kvale, S.: Interviews: an introduction to qualitative research interviewing. Sage Publications, Thousand Oaks (1996)

Modelling of the Ward Round Process in a Healthcare Unit

Adid Ali Fareedi and Vladimir Tarasov

School of Engineering, Jönköping University,
P.O. Box 1026, 55111 Jönköping, Sweden
alab08af@student.hj.se, vladimir.tarasov@jth.hj.se

Abstract. Information systems (IS) are nowadays extensively used to support all kinds of activities in healthcare organisations. Enterprise modelling can help to make the use of IS in healthcare more effective by providing process and domain models reflecting a particular healthcare unit. This paper proposes a model of the ward round process in a healthcare unit. The proposed model identifies the roles of medical professionals, tasks that can be performed according to the personnel's competences, and activities that are carried out as part of the tasks to achieve goals of the ward round process. A formal approach has been used to implement the modelling results in the form of an ontology. Such formal ontologies can support improvement and development of IS in healthcare. We learned that modelling workshops are important for development of models that can be formalized in a machine-readable form.

Keywords: Enterprise modelling, conceptual modelling, ontologies, healthcare process, ward round.

1 Introduction

Information systems (IS) are nowadays part of almost every activity or process in healthcare organisations. IS are intended to support the medical personnel and make their work more efficient. Yet, the use of IS in healthcare needs to be improved [1]. Current IS do not always meet the needs of healthcare professionals and patients. To achieve better use of IS, we need to understand work processes, which are to be supported by IS, the needs of the people who are going to use IS, and the details of the healthcare domain. Enterprise modelling provides means to model the current situation and to describe work processes, tasks, roles, and resources. Thus, enterprise modelling can help to make the use of IS in healthcare more effective by providing models reflecting a particular healthcare unit. Conceptual modelling is an important part of enterprise modelling [2]. The advantage of conceptual modelling is that the resulting model can be formalised as an ontology in a machine-readable form, that can be directly used to improve existing IS or build new ones. Ontological modelling methods have been used to support development of healthcare IS in may cases (e.g. [3,4,5]).

The purpose of this paper is to propose a model of the ward round process at Ryhov hospital in Jönköping. The model is implemented as a formal ontology.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 223–237, 2011.

Existing IS have not been able to provide full support to medical professionals in all different contexts of the healthcare processes at Ryhov hospital. Development of a formal model of the ward round process is needed to help to customize current IS to improve information flow, i.e. to provide the required information to an individual according to his/her role and competence in a specific context of the ward round at the Ryhov Hospital. The main contribution of the work is that it presents a detailed representation of the ward round that can be used for development or improvement of healthcare IS. Such IS can provide better support for the healthcare professionals to improve the quality of the patient's treatment. The ontology-based implementation of the model can be also directly utilised in IS to improve information flow in a healthcare organisation.

The paper is structured as follows. Section 2 briefly describes modelling in healthcare, explains the Ward Round process and presents the case. The method used for carrying out this study is presented in section 3. Section 4 details the developed model of the Ward Round process from the model design to the implementation. Section 5 discusses the lessons learned during the model development. The summary of the results and future work are presented in section 6.

2 Background

This section provides short overview of different modelling approaches applied in the healthcare area as well as briefly describes the ward round and its importance in the medical institutions. The section ends with an introduction of the modelling case.

2.1 Modelling in Healthcare

Enterprise modelling (EM) plays an important role in the healthcare sector because it promises to improve the use of healthcare information systems (IS). EM can be used to analyse patterns of healthcare activities, ensure fulfilment of the end-user's needs and requirements, build a systematic view of patient-centred processes, and help healthcare institutions to improve internal knowledge and understanding. EM also contributes to supporting IS that are intended to help healthcare professionals to manage patient treatment activities and document all the events of the care processes [6]. Enterprise modelling approach has been used for supporting a set of structured, goal/problem-driven models for capturing, structuring and representing organizational knowledge, and designing different perspectives of process modelling: functional, informational, organizational and behavioural [7,8,9].

New business demands imposed on enterprises and organisations require EM methods that are more formal, i.e. conceptual modelling methods [2]. Ontological modelling is the key tool in formal conceptual modelling since "*an ontology is an explicit specification of a conceptualization*" [10]. Ontology development is a method to formalize conceptual models in machine-readable form that can be utilized for improvement of existing healthcare IS or development of new IS.

Nowadays, the importance of ontological modelling has been acknowledged due to its usage in the healthcare sector. Ontological modelling helps to acquire knowledge in the machine-understandable form, which results in making IS more intelligent. This approach has been used in the healthcare area for modelling of the patient's treatment process and for improvement of the quality of healthcare services [3]. To achieve the latter, efficient healthcare workflow management systems are needed that support data exchange between different processes and provide relevant information timely, effectively and according to the demand of individuals. An ontological knowledge framework presented in [3] covers the hospital processes from patient records to hospital resources and administrative tasks. An adaptive workflow system constructs a process and resource model, which include different levels to represent the patient treatment processes: patient's admission, treatment planning, and discharge.

Different approaches are utilized for ontological modelling in the healthcare sector nowadays. The archetype pattern approach is employed to develop an ontology for a healthcare IS and to provide shared understanding of the healthcare processes [4]. The ontology-driven multi-agent approach provides a framework to help the medical professionals to interact and collaborate effectively [5]. The competence management approach is used to develop competence of healthcare personnel based on goals in order to enhance competence at the enterprise level. The case study of Municipal Hospital of Karlsruhe details this competence management approach [11].

2.2 The Ward Round Process

The ward round is the process intended for making decisions concerning the patient's treatment process by medical professionals and resource personnel [12]. It provides opportunity for multidisciplinary teams to carry out different activities: patient's examination, treatment and discharge from the hospital [13]. Ward rounds support patient treatment planning, prognosis formulation and analysis of social, psychological, rehabilitation and placement issues [12]. Learning opportunities are supported by ward round as well [14]. Ward round is categorized into traditional and modern ones. The traditional ward round is authoritatively led by senior hospital personnel who have responsibility for making decisions in the patient treatment process. On the contrary, the modern ward round relies on expertise of both senior and junior colleagues and aims at achieving consensus in order to improve the quality of healthcare from the patient perspective [12,15].

2.3 The Ryhov Hospital Case

The work described in this paper was partly carried out within the project "Bridging the Gaps". It is a research project aimed at improving healthcare and driven by The Jönköping Academy for Improvement of Health and Welfare[1].

[1] http://www.jonkopingacademy.se

The healthcare processes modelled in Sect. 4 represent the ward round at Ryhov Hospital in Jönköping, Sweden.

During the ward round, healthcare professionals with different competences, roles and skills gather at the appointed time to assess the patient's status, collect information for making diagnosis as well as plan the patient's treatment process. The head nurse consults the EBBA information system to get data about availability of vacant beds to enhance the treatment pace of normal patients, emergency patients, patients sent from other healthcare units, and patients admitted through the certain ward. Every morning the medical staff receive information regarding a particular patient in the form of medical reports from different information channels like Cambio Cosmic (electronic medical record system), EBBA (patient-admittance status system), and a database connected to the pathology laboratory. These systems are integrated and share information in real-time manner to support intelligent decision making during the ward round in a healthcare unit. The details about the case can be found in [15].

3 Method

Modelling of the ward round case (see Sect. 2.3) has been partly carried out as a collaborative activity with participants of different backgrounds such as modelling, medical, and IT ones. As soon as the goal was to create a healthcare process model as a formal ontology, the other part of the modelling was only performed by the modelling experts because it required very specialized expertise. The formal ontology was the intention since it could be used to support development and improvement of healthcare IS. During the modelling activity we followed a number of steps [15]:

- Arrange a modelling workshop with domain and modelling experts,
- Acquire knowledge about the ward round during the modelling workshop,
- Prepare the case data based on the modelling workshop,
- Support the case with the study of related literature,
- Analyse the results of the domain modelling and literature review,
- Determine appropriate tasks, processes, activities, and roles comprising the ward round,
- Describe steps needed to perform the tasks,
- Select a suitable methodology for implementation of the model as a formal ontology and perform the implementation,
- Verify and evaluate the constructed model through domain experts' assessment and description logic queries.

The modelling workshops have been used as one of the main techniques in this study. We have conducted two workshops. The first one was intended to acquire relevant knowledge from the domain experts for creation of the model of the ward round at Ryhov hospital in Jönköping. In this workshop, we have engaged multidisciplinary professionals such as modelling experts, medical practitioners and IT experts with their experiences, observations and different expertise to develop a model of the ward round. Before presenting the results in the second modelling workshop, we developed a formal model that can be useful for the improvement of healthcare IS. During the second workshop we presented the results of the modelling to the domain experts to ensure the quality of the model. We also discussed design issues like individual roles and key responsibilities in the ward round, and different resources that were used in ward rounds.

4 Model of the Ward Round Process

This section presents design and briefs implementation of the model of the ward round and explains different activities which are necessary to perform the ward round process. The details of the model are described in [15]. Before considering the ward round for a patient's treatment, it is important to evaluate the patient's condition at the time of admission. The criteria for the admission procedure are also described in [15].

4.1 Design of the Model

The design of the model, which resulted from the first modelling workshop, is depicted in Fig. 1. It shows the structure of the ward round at Ryhov hospital in Jönköping. The ward round is the centre of activities where different hospital professionals gather for treatment of the patient. The overall ward round process is divided into three sub-processes, which consequently support each other. These sub-processes are the pre-ward round process, the ward round process and post-ward round process.

The ward round process includes different tasks that are carried out to treat the patient's illness. Hospital professionals have different roles in the process and the ward round has different goals to achieve quality in the patient's treatment process. These goals lead to different tasks, which include activities in the patient's treatment process. Different resources are utilized for the completion of each task. Resources can be patient treatment documents, medical equipment, IS, competence resources and database systems.

4.2 Ward Round Process

In ward round process modelling, we have divided medical professionals into three designated teams: *designated team 1*, *designated team 2* and *designated team 3*. These teams consist of multidisciplinary professionals who are responsible for initiation of the sub-processes: the pre-ward round process, ward round process, and post-ward round process. The teams perform different tasks and activities and use resources to achieve the set goals. *Designated team 1* includes a junior practitioner, head nurse, additional nurse, occupational therapist, and laboratory personnel that are responsible for update of the patient's medical record in the pre-ward round process. *Designated team 2* consists of a consultant, senior and junior practitioners, nurse, and occupational therapist to perform the

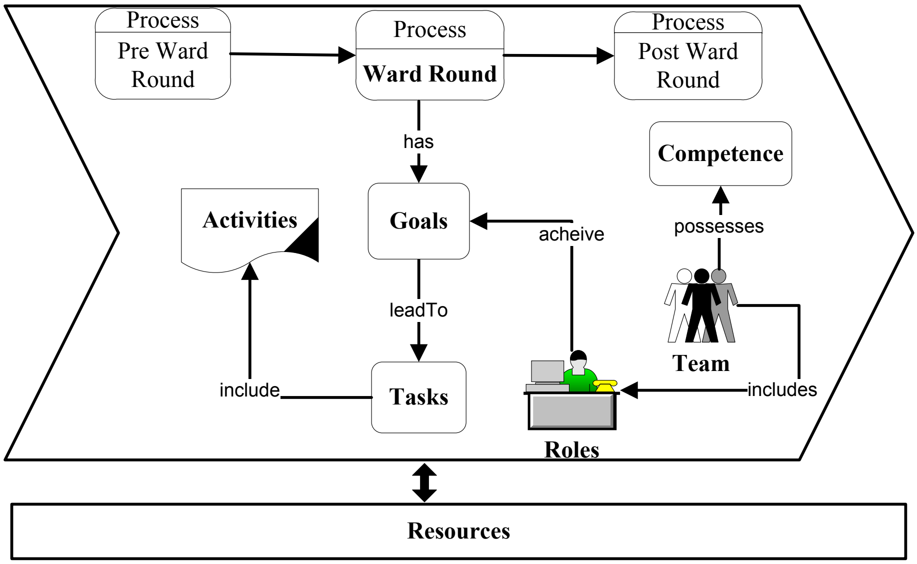


Fig. 1. Overview of the model

ward round process for planning of the patient's treatment. *Designated team 3* is also comprised of senior and junior practitioners, an occupational therapist and medical student for group discussion and evaluation of the current analysis of the patient's illness in the post-ward round process.

We have chosen *designated team 2* for a more detailed description. This team initiates the ward round process with the goal of timely result receipt, the task of patient treatment planning, and the activity of identifying medical problems in the patient's treatment in the ward round session. *Designated team 2* utilizes the resources EBBA and Cosmic to achieve quality of patient treatment in the ward round process. The process of modelling included identification of goals, tasks, activities, roles, and resources, which are listed in the following subsections.

Goals. The following goals are important to achieve quality of the patient treatment planning process in the ward round:

- To analyse the actual causes of the patient's illness,
- To support quality management of planning in the ward round,
- To prepare a pool of additional questions related to the patient's previous life or medical, psychiatric, sexual, family and social aspects,
- To achieve effective decision making in immediate or long-term treatment planning to tackle the patient's illness.

Tasks. The tasks of planning of the patient's treatment.

Activities. The activities that help to carry out the above mentioned task are:

- To consider a differential diagnosis in the patient's treatment planning in the ward round process [16],
- To formulate a differential diagnosis, which involves review of the patient's history and clinical examination of the findings in the patient's treatment planning [16],
- To identify medical problems connected with the patient's illness and to construct a management plan to tackle the disease [16],
- To ask additional questions to clarify different aspects of the patient's history regarding medicine, effects of medicine, sensitive aspects of psychiatric and sexual nature, and evaluate risk factors related to consequences of the illness, family history and social history [16],
- To decide on the patient's treatment plan according to the patient's illness—either an immediate plan or long-term one [16].

Roles. The following roles are extracted from the case during the modelling of the ward round process:

- A consultant is responsible for identification of all possible causes after the review the patient's history during patient treatment planning,
- A consultant is also responsible for identification of medical problems related to the patient's illness during patient treatment planning,
- A senior practitioner formulates a differential diagnosis after the review of the patient's history and examination of the clinical findings,
- A junior practitioners assists a senior one in collection of the examination findings,
- A nurse provides medical reports related to the patient or the patient's portfolio,
- A junior practitioners is also responsible for asking the patient additional questions to support the quality of patient treatment planning,
- A senior practitioner decides on the treatment plan according to the patient's illness—either an immediate plan or long-term plan.

Resources. To perform the activities of the task of patient treatment planning, several resources are utilized during the ward round in Ryhov hospital at Jönköping:

- The information systems Cosmic and EBBA are used for acquiring information related to the patient in the process of patient treatment planning,
- The laboratory database system is used to send reports on the patient medical tests to nurses for patient treatment planning,
- Patient medical reports are used for the assessment of the patient treatment process.

4.3 A Process View of the Ward Round

We have employed EKD modelling technique (see [17]) to design the ward round model shown in Fig. 1. This sections describes key processes, external processes and information sets, which are depicted in Fig. 2. They constitute the process view of the ward round model. An external process is a collection of activities that are located outside the scope of the organizational activity area and communicate with processes or activities of the problem domain area [17]. An information set is a set of information, which is sent from one process to another one to facilitate information flow between the processes [17]. The following processes represent the working flow of the ward round:

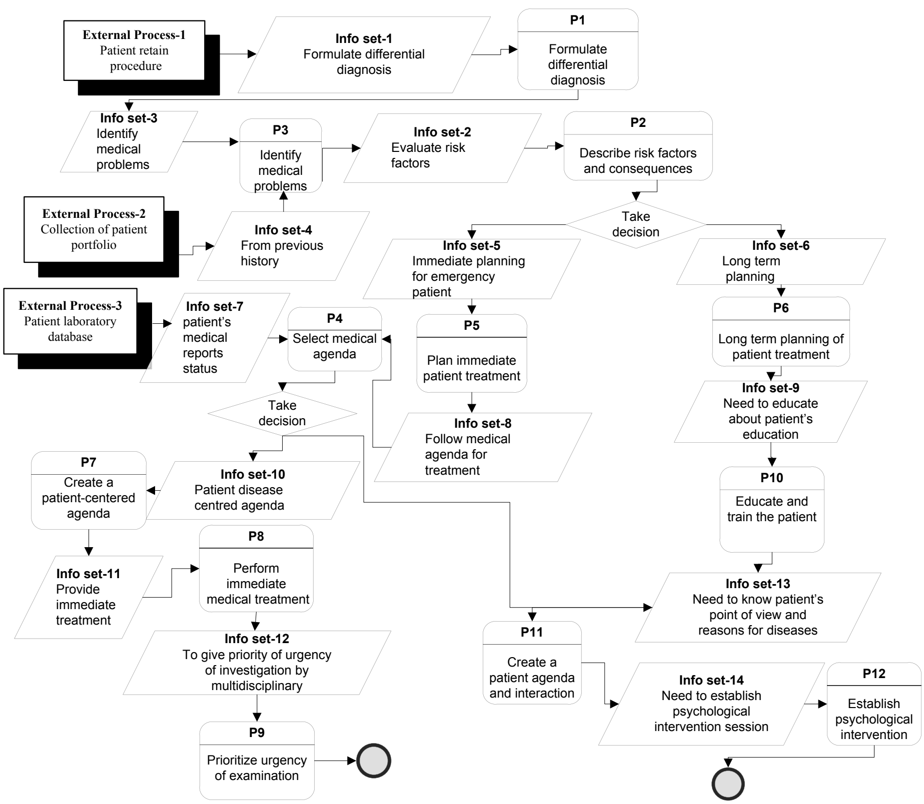


Fig. 2. The process perspective in the ward round model

1. *Formulate differential diagnosis.* This process helps to describe a number of steps to determine the problem at the time of diagnosis.
2. *Describe risk factors and consequences.* This process explains risk factors.
3. *Identify medical problems.* It details medical problems in the diagnosis.

4. *Select medical agenda*. This process chooses a medical treatment agenda: a long-term one or short-term one.
5. *Plan immediate patient treatment*. This describes emergent patient treatment in the ward round.
6. *Long term planning of patient treatment*. Long term treatment is described for a normal patient in the ward round.
7. *Create a patient-centred agenda*. This process determines particular disease treatment.
8. *Perform immediate medical treatment*. Immediate treatment is provided to the patient in the ward round.
9. *Prioritize urgency of examination*. It establishes what data are most necessary to conduct the ward round.
10. *Educate and train the patient*. One needs to know the patient's point of view and explain reasons for disease treatment in the ward round.
11. *Create a patient agenda and interaction*. This process establishes good interaction with the patient in the ward round.
12. *Establish psychological intervention*. A psychological intervention session needs to be established during the ward round.

4.4 Implementation of the Model

This section briefly explains the implementation of the ward round model (detailed description can be found in [15]). A formal approach has been used which resulted in an ontology. We have used the Web Ontology Language (OWL)[2] and the Protégé ontology editor[3] to develop the ontology-based implementation. The constructed ontology contains 146 classes, 76 object properties, 1 data property, and 122 individuals. Fig. 3 depicts an overview of the ontology that contains different entities involved in the ward round and relations between them.

The medical staff members are involved in different processes (pre-ward round, ward round, and post-ward round) according to their competences and assigned roles. The assignment of the roles of a consultant or senior practitioner determines which process is initiated. The selection of a process is followed by the medical agenda [18] in the patient treatment procedure. Different processes in the ward round model utilize diverse resources for receiving related information from the information channels according to the rules set by a healthcare unit. These processes have particular goals. Every goal leads to a number of tasks that include activities. "Component selection criteria" is a component that supports the selection of the right person to perform his/her responsibilities according to the assigned roles. The selection is carried out based on the person's competences and skills.

To illustrate the use of the constructed ontology for modelling of a real situation from the ward round, we give an example scenario describing the role of a junior practitioner and show its ontological representation in Fig. 4.

[2] http://www.w3.org/TR/owl2-overview
[3] http://protege.stanford.edu

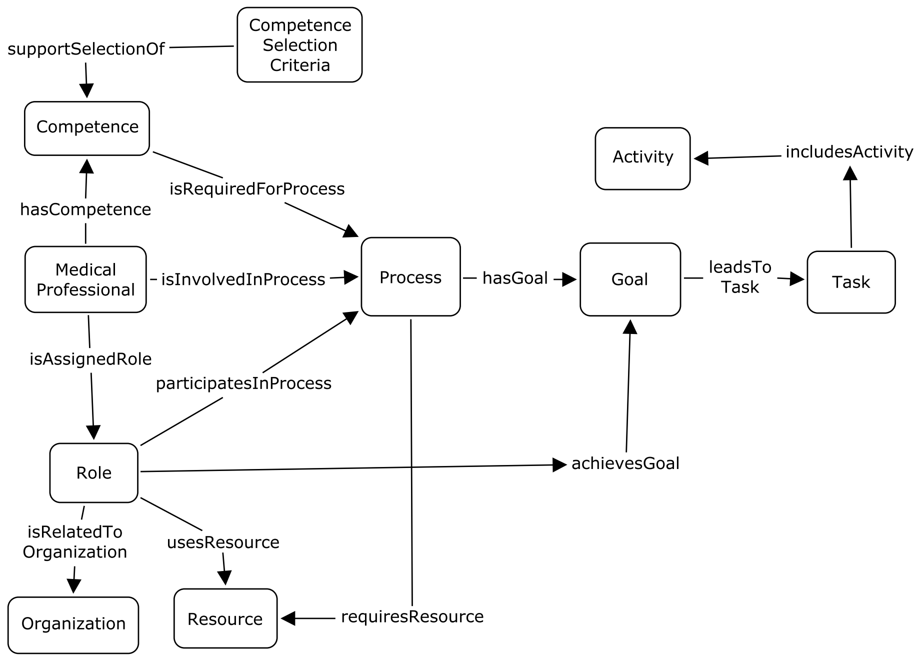


Fig. 3. Overview of the ontology-based implementation of the ward round model

Scenario. The junior practitioner role is assigned to Person A who works at Ryhov Hospital in Jönköping and belongs to designated team 2. Person A participates in the ward round process and performs different tasks like patient treatment planning. Person A carries out several activities such as to complete the physical examination to determine medical problems for patient treatment planning. Person A's main responsibilities are to visit a particular ward daily and check the patient's status from the Cosmic information system to inform the senior staff. Person A has several competences: cultural, occupational and general ones (they are not shown for brevity).

4.5 Evaluation of the Model

The goal of the evaluation was to achieve quality of the work and get assurance that the model represents the ward round in the correct way. The evaluation of the model was done in two ways [15]:

1. *Using description logic (DL) queries.* We have used the DL Query Tab in Protégé 4 to verify competency questions. They help to confirm that the ontology has enough information to answer these questions, which are related

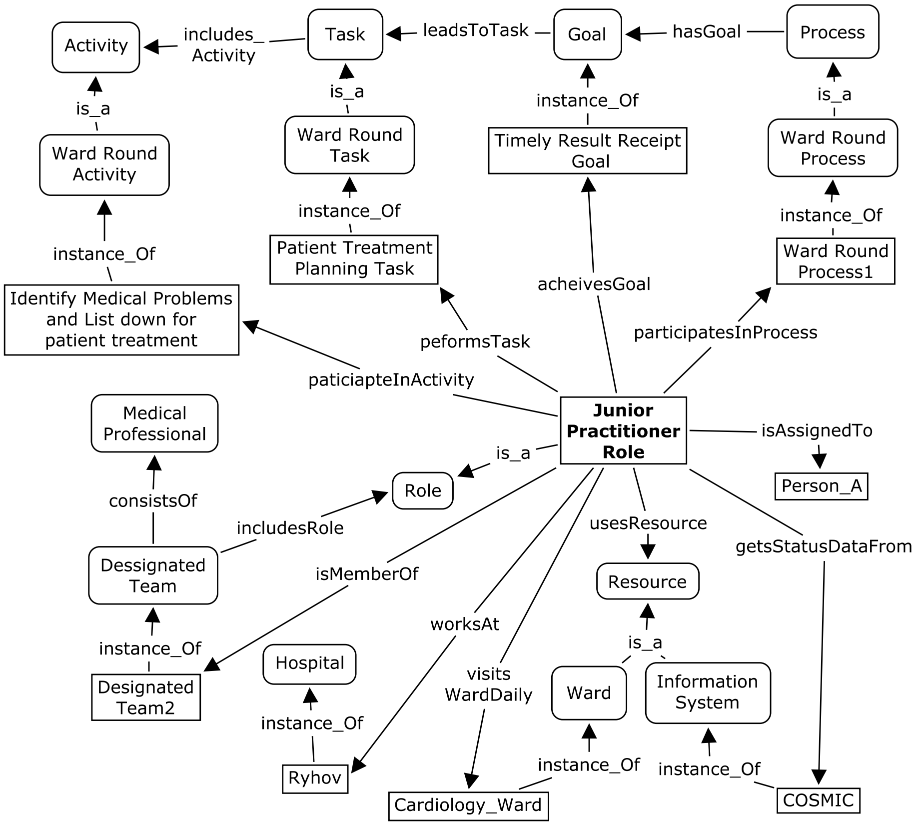


Fig. 4. The junior practitioner role represented with the constructed ontology

to the domain (the ward round). Table 1 shows an example of one competency question with a DL query and Fig. 5 presents the results of executing the DL query in the Protégé editor.

2. 2^{nd} *modelling workshop.* During this workshop, we have presented holistic view of the model design that illustrates how medical professionals with different roles and competences initiate different processes and perform different activities to achieve the ward round goals. We have also exemplified the model of the Ryhov hospital case with a simple scenario that shows representation of a practical situation. These formal modelling results reflect detailed representation of the ward round, which can be used in healthcare IS to improve information flow in the ward round context. Thus, during the 2^{nd} modelling workshop we have presented the final modelling results to the domain experts to verify the model, get feedback and suggestions for future improvement.

Table 1. An example of a competency question and a DL query

Competency question	DL query	Query results
Who are the members involved in ward round team?	`Role and initiateProcess value WardroundProcess1`	Medical Individuals: *OccupationalTherapist*, *AdditionalNurse*, *SeniorPractitioner*, *HeadNurse*, *JuniorPractitioner*

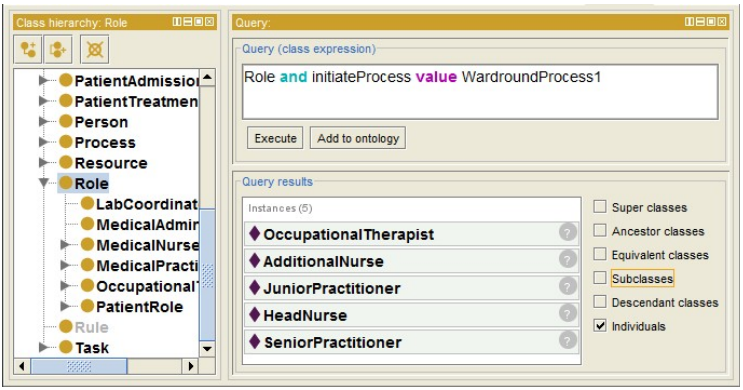


Fig. 5. The results of executing the DL query in the DL query tab in Protégé

5 Lessons Learned

The main objective of the modelling workshops was to develop a model that can be formalised as an ontology in a machine-readable form to support improvement and development of IS in healthcare. We learned that using modelling workshops was very productive to acquire relevant knowledge about the Ward Round to develop an ontological model. By using modelling workshops and other modelling techniques like EKD [17], we were able to develop a model including tacit knowledge acquired from the domain experts and then implement the model in a form that allows for improvement of IS use in different domains like healthcare. Meanwhile the formal part of the modelling was necessary to carry out without involvement of the domain experts as soon as formal techniques, which require very special expertise, would confuse the domain experts.

The steps, which we described in Sect. 3, are well suited to develop any model in the healthcare sector in the systematic way. However, we have observed that participation of only multidisciplinary professionals with different expertise like modelling or use of IT in healthcare during the modelling workshops is not enough. We should also invite healthcare professionals to obtain better knowledge of intrinsic details from the healthcare domain. This may help to develop better ontological models in healthcare.

It is also important to include healthcare professionals in the modelling workshops from the beginning because it is quite helpful for modellers to learn and get experience from different domain professionals and to train to work with participants with mixed culture and background. This work pattern helps the modellers to present the modelling results to the domain experts because they already familiar with the experts. It also creates synergy effects among the participants and improves the quality of the work. According to our experience, we can conclude that modelling workshops are a good opportunity specifically for new-comers in modelling. Novice modellers can quickly learn how to perceive the domain and choose a suitable methodology for modelling from the other multidisciplinary participants.

We have arranged two modelling workshops in our research work but we think that it is not sufficient. During the 1^{st} workshop, we familiarised ourselves with the participants and got "know-how" about the domain. In the 2^{nd} workshop, we presented the results in the form of models to the domain experts to get feedback. We suggest to conduct more workshops with healthcare professionals for improvement and assessment of the model.

We have presented our results to the domain experts during the modelling workshops in diagrammatic form that provided overview of the ward round model in the holistic way. This approach facilitates broad understanding of how the objects in the domain are connected with each other and what the information flow is. We have also used tabular form to present our results to the domain experts. The tabular form allowed them to get additional details about the elements in the ontology-based model. We have learned that both forms are useful for presentation of modelling results.

Different modelling tools have been considered, especially data and software modelling tools. However, finally we chose specialized tools for ontology-based model construction. We think that ontology editors like Protégé 4[4] and Top Braid Composer[5] are appropriate for model construction. During the development we have found that these tools are good because they allow for both expressiveness in models and evaluation of the modelling results.

6 Conclusions

This paper proposed a model representing the ward round process at Ryhov hospital in Jönköping. The model contains individual roles to be taken on by medical professionals while carrying out tasks to achieve effective patient treatment in the ward round process. During this process multidisciplinary professionals form different teams to perform tasks to achieve goals of the ward round. The model also shows different processes at different stages of the ward round. The hospital personnel have roles according to their competences required to perform activities in these processes.

[4] http://protege.stanford.edu

[5] http://www.topquadrant.com/products/TB_Composer.html

The work was carried out based on two modelling workshops and literature study. We have implemented the constructed model in the form of a formal ontology. The ontological implementation has been developed using the Web Ontology Language (OWL)[6] and the Protégé[7] ontology editor. For the evaluation of the results, we have utilized two ways: description logic (DL) queries and a 2nd modelling workshop to verify the constructed model.

This work suggests that formal modelling techniques are important for the healthcare sector. The constructed ontology-based implementation of the model can be used to represent information needs of the healthcare professionals in the ward round patient treatment process in a healthcare unit. The formal ontology can be utilised directly in IS, which allows for the use of semantic techniques to improve information flow in the ward round process. For example, the ontology can be employed to improve the Cambio Cosmic IS, which is used at Ryhov hospital, in order to provide the doctors and nurses with the needed patient records, lab test results and so on according to their way of working during the ward round.

During the work we have learned that more modelling workshops are needed to create models that represent the domain in a more detailed and accurate way. Healthcare professionals are encouraged to be included in the modelling workshops from the very beginning. Diagrammatic and tabular forms are appropriate to communicate results of formal modelling methods to the healthcare professionals.

Our first priority for future work is to further validate the constructed model by testing it in IS in order to provide better healthcare services. We will also continue development of formal conceptual models in healthcare that can support improvement of information flow and development of better healthcare IS. The same modelling technique can be used for construction of ontological models in other areas like inter-professional interaction during the ward round process and doctor-patient communication, unmet patient needs after the ward round session, patient and student perspectives of teaching optimization, role and place of relatives, consultant vs. junior doctor rounds, quality of notes and patient communication, and alternatives to the bed-to-bed round [12]. These areas were out of the scope of this paper but they can be used for further research in healthcare modelling.

Acknowledgements. The authors would like to thank Dr Klas Gäre and MSc Eva Lindholm for their active participation in the modelling workshops and valuable contribution to the discussion of the model.

Some parts of the presented research were carried out within the project "Bridging the Gaps" supported by the Vinvård research program, which is financed by VINNOVA and the Vårdal Foundation in coordination with the Swedish Association of Local Authorities and Regions.

References

1. Littlejohns, P., Wyatt, J.C., Garvican, L.: Evaluating computerised health information systems: Hard lessons still to be learnt. BMJ 326(7394), 860–863 (2003)

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

[7] http://protege.stanford.edu

2. Loucopoulos, P., Kavakli, V.: Enterprise knowledge management and conceptual modelling. In: Goos, G., Hartmanis, J., van Leeuwen, J., Chen, P., Akoka, J., Kangassalu, H., Thalheim, B. (eds.) Conceptual Modeling. LNCS, vol. 1565, pp. 123–143. Springer, Heidelberg (1999)
3. Dang, J., Hedayati, A., Hampel, K., Toklu, C.: An ontological knowledge framework for adaptive medical workflow. Journal of Biomedical Informatics 41(5), 829–836 (2008)
4. Raghupathi, W., Umar, A.: An archetype pattern driven approach to ontology development in health care information systems. In: Kendall, J.E. (ed.) Proceedings of the 39th Annual Meeting of the Decision Sciences Institute, pp. 4151–4156. Decision Sciences Institute, Atlanta (2008)
5. Ganendran, G., Tran, Q.N., Ganguly, P., Ray, P., Low, G.: An ontology-driven multi-agent approach for healthcare. In: Webb, R., Ribbons, R.M., Dall, V. (eds.) HIC 2002: Proceedings: Improving Quality by Lowering Barriers, pp. 464–469. Brunswick East, Vic., Health Informatics Society of Australia (2002)
6. Staccini, P., Joubert, M., Quaranta, J.F., Fieschi, D., Fieschi, M.: 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(2-3), 129–142 (2001)
7. Shahzad, K., Zdravkovic, J.: A goal–oriented approach for business process improvement using process warehouse data. In: Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 84–98. Springer, Heidelberg (2009)
8. Persson, A., Stirna, J., Aggestam, L.: How to disseminate professional knowledge in healthcare: The case of skaraborg hospital. Journal of Cases on Information Technology (JCIT) 10(4), 41–64 (2008)
9. Petersen, S.A., Bach, G., Svarlein, A.B.: Patient care across health care institutions: An enterprise modelling approach. In: Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 91–105. Springer, Heidelberg (2010)
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
11. Kunzmann, C., Schmidt, A.: Ontology-based competence management for healthcare training planning: A case study. In: 6th International Conference on Knowledge Management, IKNOW 2006 (2006)
12. O'Hare, J.A.: Anatomy of the ward round. European Journal of Internal Medicine 19(5), 309–313 (2008)
13. Arasaratnam, R.: Medical ward rounds. Student BMJ 17, 98–99 (2009)
14. Stanley, P.: Structuring ward rounds for learning: Can opportunities be created? Medical Education 32(3), 239–243 (1998)
15. Abid, A.: Ontology-based model for the "ward-round" process in healthcare (OMWRP). Master's thesis, School of Engineering of Jönköping University (2010)
16. Cherry, J., Drury, A., Rigby, S.: Management plans. Student BMJ 16(10), 364–365 (2008)
17. Bubenko, Jr., J., Persson, A., Stirna, J.: D3: Appendix B. EKD User Guide. HyperKnowledge IST–2000-28401. Royal Institute of Technology (KTH), Stockholm, Sweden (2001)
18. Price, J., Leaver, L.: ABC of psychological medicine: Beginning treatment. BMJ 325(7354), 33–35 (2002)

Modeling an Agile Enterprise: Reconciling Systems and Process Thinking

Ilia Bider[1,2], Gene Bellinger[3], and Erik Perjons[1]

[1] DSV, Stockholm University, Stockholm, Forum 100, SE-16440 Kista, Sweden
[2] IbisSoft AB, Stockholm, Box 19567, SE-10432 Stockholm, Sweden
[3] SystemsWiki.org
ilia@ibissoft.se, systemswiki@gmail.com, perjons@dsv.su.se

Abstract. Selection of an enterprise modeling paradigm depends on the practical task the modeling project is trying to achieve. For example, modeling in the frame of Enterprise Architecture paradigm is, usually, aimed at alignment of the enterprise components, such as mission, vision, business processes, services and IT systems. Modeling in the frame of Business Process Management is aimed at process improvement/optimization, and modeling in the Systems Thinking paradigm is aimed at getting a holistic view on the dynamic behavior of the enterprise. This paper suggests a new approach to enterprise modeling that combine these approaches in order to reveal and improve enterprise agility. It is based on the systemic view of business processes, and it presents an enterprise as a three-layered model consisting of assets, sensors and business process instances. Elements of this model can be recursively decomposed, which allows for different levels of details when modeling an enterprise.

Keywords: Business Process Management, Systems Thinking, Enterprise Agility.

1 Introduction

We start with two sayings that concern modeling. The first one runs as "All models are wrong, but some are useful" and is attributed to George E.P. Box [1]. The second one runs as "A method that is good for everything is good for nothing". The meaning of these two sayings in the context of the PoEM conference can be expressed as each enterprise modeling technique can be useful to solve some practical problems in certain context, but might be useless, or even counterproductive for solving other problems, or problems in another context.

Consider the three paradigms most frequently used for modeling enterprises: Business Process Management [2], [3], Enterprise Architecture [4], [5], and Systems Thinking [6], [7], each of them having countless techniques and methods. Each of these paradigms is connected to a special problem area. Business Process Management is associated with ***optimization*** of the usage of resources inside an enterprise through specialization, standardization, and automation [8]. Enterprise

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 238–252, 2011.

Architecture is associated with ***alignment*** of different parts of the enterprise, like mission, vision, structure, processes and IT systems [5]. Systems Thinking is associated with a ***holistic*** view on the dynamic behavior of the enterprise and its interaction with its environment [6]. It is directed to avoiding situations in which the dynamic behaviors existing in different parts of the enterprise, e.g., sales and production, are disconnected. For example, sales is increasing its efforts, while production is pursuing cost savings by diminishing current production levels, which after some time, results in increasing time from order to delivery.

The scope in which the above paradigms are applicable may also differ. For example, a Business Process Management project can have an enterprise-wide scope and be aimed at identifying and modeling all processes in the organization. On the other hand, the project can concern a particular process, e.g., sales, in order to optimize it. The same is true for Systems Thinking. The project can be started to overhaul the whole enterprise, or just to understand and improve a particular situation. Enterprise Architecture paradigm differs from the previous two in this aspect, it is aimed to align all components of the organization and thus, usually require a large scale project that concerns the whole enterprise To complete an Enterprise Architecture project requires time and considerable resources which leads to this paradigm normally being used by large enterprises in sectors considered to be stable, energy, large industrial enterprises, financial sector, large governmental organizations.

None of the above paradigms focuses specifically on the issue of enterprise/business ***agility*** - property of an enterprise to function in the highly dynamic world [9]. The agility concerns both being able to adjust the enterprise to changes in the surrounding environment, and discovering new opportunities constantly appearing in the dynamic world for launching completely new products/services. Becoming agile requires a structure that allows discovering changes and opportunities as soon as possible and react on them appropriately. A model that reveals the current level of agility, and can show ways of improving it would be beneficial for a new generation of highly dynamic team based enterprises.

This paper is aimed at presenting an idea of how a enterprise model explicitly related to issues of enterprise agility could look like. The model can be the base for a new modeling technique managing agility. While creating this model, we do not reject the concepts and features existing in the other three paradigms. Actually, we borrow all we can and adjust and integrate the borrowed concepts in a manner that is appropriate for the goal our model is aimed to serve. For example, a reaction on a discovered change is defined as starting a business process instance/case. However, we do not insist that this process should be optimal, or need to be optimized in the future. It is enough if the process is very loosely defined so that an instance of such a process can be completed in a relatively short time. Optimization may not be appropriate in the dynamic environment as it will hinder the creativity needed for handling a particular process instance.

This work has its roots in the heated discussions in the LinkedIn group Systems Thinking World [10] on whether terms *system* and *process* represent the same or different concepts. These discussions resulted in our creation of the model managing enterprise agility. The model is based on applying a systemic view to business processes. Standard for Enterprise Architecture concepts as mission, vision, policies, structures, are taken into consideration, but are not set in the focus. The resulting enterprise model consists of three layers:

– Enterprise assets
– Sensors
– Business process instances

The rest of the paper is structured according to the following plan. In Section 2, we present a systemic perspective on business processes. Based on this perspective, in Section 3, we introduce the three-layered model for managing enterprise agility as mentioned above. In Section 4, we introduced a simplified classification of business processes based on the nature of their goals. In section 5, we explain how elements of different layers of the model interact with each other. In section 6, we discuss how sensors and business process instances can be recursively decomposed. In section 7, we present some informal examples of application of the ideas introduced in the previous sections. Section 8 is devoted to related work. In section 9, we discuss the implications of the three-layered model, and draw plans for the future that concern the transition from the idea presented in the paper to the practical methodology of enterprise analysis and modeling.

2 Systemic View on Business Processes

According to Systems Thinking [6,7] an enterprise is regarded as a whole, i.e., a system. Such a system maintains its existence through constant interaction between its parts, i.e., people, departments, teams, etc. The system interacts also with its environment, a bigger whole, manifesting a unique behavior that cannot be derived from the sum of its parts. From a Business Process Management perspective [3], an enterprise is regarded as a number of repeatable business processes. These two perspective looks on the surface completely different, and to integrate both of them in a new modeling technique we need, first, to reconcile them. The reconciliation is done via considering processes as a special type of systems.

The term business process encompasses two concepts (which often confuse outsiders):

– *business process instance* (BPI) or case, for example, delivering a service to a particular customer after receiving a call for service
– *business process type* (BPT) or template which refers to all possible instances of a particular kind, e.g., all service delivery instances.

A business process instance (BPI) is a system with a short lifespan. It can be minutes, hours, days, months or years, but its lifespan is always (considerably) shorter than the lifetime of the whole enterprise. Such a system is created to achieve some (operational) goal, e.g., to deliver service, or goods ordered by a given customer and get paid. This system is disbanded after the goal has been reached. A BPI system includes the same components as the whole enterprise system (e.g., people, departments) and some elements of the environment (e.g., customers, investors). At any moment of time, an enterprise has numerous BPIs in progress providing the majority of the interactions between the system's components and the system, and the environment.

A business process type (BPT) can be considered as a set of rules that describes the dynamic behavior of BPIs aimed at reaching a certain class of operational goals. A BPT consists of two parts:

- *Start conditions* that defines when a new BPI system of the given type should be created
- *Execution rules* that define what should be the goal of the process, whom should be included in the process, how the job should be done in the process, how components interact, etc.

The execution rules can be prescriptive (e.g., what should be done), constraint-based restrictive (e.g., what should never be done), recommended (e.g., how normally things are done but it is allowed to do it differently), or a combination of the above. BPT rules are "imprinted" in manuals, process maps, employees' handbooks, computer systems, heads of employees (oral tradition), or a combination of the above. In other words, the knowledge on the rules (and the processes themselves) can range from being completely tacit (e.g., resides in the heads of the process participants), to partly or totally explicit (e.g., depicted in detailed process maps).

BPTs work as business DNA creating BPIs based on the needs, e.g., impulses or changes in the environment or inside the organizational system itself. BPTs plus tools used in BPIs, e.g., telephone lines, computers, production lines, constitute an organizational system's infrastructure that allows it to effectively function inside the given environment. BPTs constitute some kind of hierarchy. The lower levels are occupied by BPTs that produce BPIs as a reaction to simple impulses, like incoming order from a customer. The higher levels are occupied by more strategic BPTs that react on more substantial changes in the environment by reconfiguring the system itself, which may include changing BPTs (a kind of genetic engineering), introducing new BPTs, or deleting the obsolete ones.

3 The Three-Layered Enterprise Model

Based on the elaboration in Section 2, we can view an organization as a three-layered model as depicted in Fig. 1, and explained below:

1. An assets layer consists of:

- People - with their knowledge and practical experiences, beliefs, culture, sets of values, etc.
- Physical artifacts – such as computers, telephone lines, production lines, etc.
- Organizational artifacts, formal as well as informal – such as departments, teams, networks, roles, etc.

– Information artifacts – such as policy documents, manuals, process descriptions (BPTs), etc. To the information artifacts belong both written (documented) artifacts, and tacit artifacts - the ones that are imprinted in people's heads (e.g., culture.)

The assets layer is relatively static, which means that this layer by itself cannot change anything. The components of this layer are activated when they are included in the other two layers. This layer itself can be changed through other layers when the assets are set in motion for achieving some useful goals. Note that assets here are not regarded in pure mechanical terms. All "soft" assets, like sense of common goals, degree of collaboration, shared vision etc., belong to the organizational assets. Note also that having organizational artifacts does not imply a traditional function oriented structure. Any kind of informal network, or resource oriented structural unit, is considered as organizational artifact.

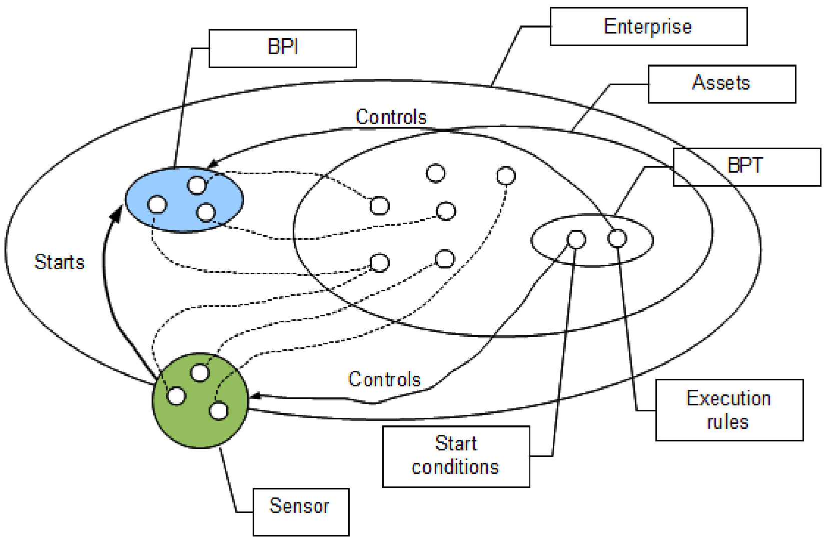


Fig. 1. Three-layered model of an enterprise as a system

2. Sensor layer consists of a set of (sub)systems, the goal of which is to watch the state of the enterprise itself and its environment in order to catch impulses and slower changes that require firing of business process instances of certain types. We need a sensor (which might be a distributed one) for each business process type. The work of a sensor is governed by the Start Conditions of the BPT description (which is an informational artifact). A sensor can be fully automatic for some processes (an order placed by a customer in a web-based shop), or require human participation to detect changes in the system or its surroundings.

3. BPIs layer - a set of systems initiated by sensors for reaching certain goals and disbanded when these goals are achieved. The behavior of a BPI system is governed by the Execution Rules of the corresponding BPT. Dependent on the type, BPIs can lead to changes being made in the assets layer. New people are

hired or fired, departments are reorganized, roles are changed, new policies are adopted, BPT descriptions are changed, new BPTs are introduced, and obsolete ones are removed.

4 Classification of Business Processes

We can roughly differentiate three categories of business processes dependent on the complexity of the sensor, and the nature of the process itself:

1. The first category encompasses operational processes, like sales, production, HR (e.g., hiring), etc. A sensor discovers the need (e.g., customer needs - sales, or internal needs - HR), and initiates a relatively structured process instance to attain the operational goal (e.g., making a deal, or hiring a new employee).
2. The second category encompasses process improvement (optimization) processes. A sensor here is based on the performance indicators established to measure efficiency, productivity, or other parameters of the given BPT. If the performance is not according to the expectations, an improvement (re-engineering) BPI starts with the goal to change process definition(s) used by operational processes. The improvement BPI here can follow some known methods (like Six Sigma, or lean). As a rule, the improvement processes are less structured than the operational ones.
3. The third category encompasses strategic processes. A sensor here is based on the macro view on the whole organizations. If the overall performance is below expectation, a strategic BPI is fired with the goal of considerably changing the assets layer. This can include radical changes in process definitions, removing obsolete processes, introducing new ones, rearranging departments, substituting key-managers, introducing new technology, etc. These are the processes where Systems Thinking is (though maybe too seldom) applied as guidelines for finding the best places to make changes (leverage points). A process here may be completely ad hoc, or use some loose structure, e.g., a series of brainstorming sessions.

5 Interplay between the Layers

Due to the interplay between the three layers, an enterprise behaves as an adaptive system. It constantly interacts with the environment based on the BPTs of operational processes, optimizes itself to the current environment through the improvement processes, and can reconfigure itself when the environment changes based on the strategic processes (after which it can start optimization to the new environment).

Basic interconnections between the elements of different layers are depicted in Fig. 1. These, and the additional ones that are not explicitly shown in Fig. 1, are explained below.

A BPI (layer 3) is started by the corresponding sensor (layer 2), and it uses organizational assets (layer 1), such as people, machines, etc. to produce the result, i.e., reach some operational goal. In particular, the execution rules of the BPT (layer 1) controls the behavior of each BPI that belongs to this process type.

As was already mentioned, a BPI can be started to intentionally change the organizational assets (layer 1), fire or hire people, change policies, change BPTs, reorganize departments, etc. But even when a BPI does none of the above, it does make a change just because it was running for some period of time. During its run, a BPI creates a trace either on paper, or inside the organizational database, or just in the heads of people participating in this process instance. Depending on the type of organizational memory, the trace may stay in memory a very short period of time, or last "forever". All remembered traces of BPIs constitute the experience based knowledge of the organization, which, as an information artifact, adds to the assets layer (layer 1).

A sensor (layer 2) starts a BPI (layer 3) each time it detects that its start conditions are satisfied, and it uses assets (layer 1), such as people, machines, etc., to detect this situation. In particular, the start conditions of the BPT (layer 1) controls the behavior of the sensor (see Fig. 1).

For a sensor to identify when a BPI is to be started, it needs access to the relevant information in order to test the start conditions. This information is usually delivered by some already finished BPIs (layer 3) and can be found in their traces (layer 1). The information that sensors need can be provided by normal BPIs aimed at practical goals (as their side effects). However, in many cases, just having standard operational business process is not enough to provide all information needed for the sensors. Special "information gathering" business processes need to be designed with the only goal of obtaining relevant information for the sensors. The start condition for such a business process could be very simple, a BPI should run each year, month, week, year, our hour. Information gathering processes are especially needed for the sensors belonging to the categories 2 and 3 of the process classification from Section 4. A periodical survey of the customers to determine their level of satisfaction is a typical example of an information gathering business processes.

Another way of obtaining information needed for sensors is to enhance the standard operational processes in order to gather this information during the BPIs runs. This can be done by adding to them steps (operations) that are not important for these processes as such, but can provide information for sensors belonging to other business processes. An example of such enhancement could be a set of questions to the potential customer who has chosen not to buy a product or service, just to find out the reason for his/her decision (wrong price range, wrong service, etc.).

6 Decomposition

Both a sensor and a BPI are systems, and thus they can be, if necessary, decomposed. Consider an example of a compound sensor. Let us have a fast growing enterprise that wants to keep the pace of its expansion for a number of years ahead. Let this enterprise be a consulting business, the growth of which depends on the number of employees. The management decides to run a strategic overview (BPI) according to some template (BPT) as soon as there is a danger for growth or decline (start condition of the BPT). One of the parameters that reflect the pace of growth is the rate of consultants hiring minus rate of losing them to competitors. A sensor needs to evaluate this data against some threshold value and start a strategic overview when the value is below the threshold. There could be two possibilities to create such a sensor:

– ensure that all needed information is available in real-time, then the sensor just needs to do some math and issue a start signal if necessary
– make a periodic, say once a month, information gathering, and only then do some math, and issue a start signal if necessary

The second case represents a compound sensor as in Fig. 2, which consists of a simple Sensor1 that "watches" the clock and starts an information gathering BPI on the 1st of each month. The BPI gathers information and produces a report (information artifact). Another sensor, Sensor2 in Fig. 2, reacts to a new report, makes comparisons and starts a new strategic overview if necessary.

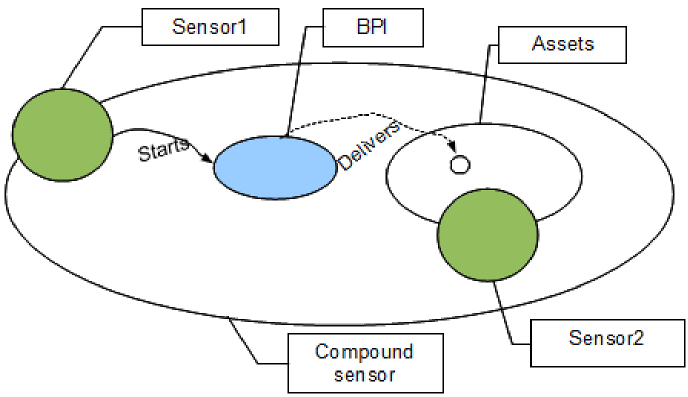


Fig. 2. A compound censor

A similar decomposition can be done for a BPI, for example, when the BPI represents a complex BPI, e.g., a project. Complex sensors/BPIs are more typical for the category 2 and 3 of the process classification in Section 4 than for the category 1 (operational processes).

7 Applying the Model

7.1 "Analysis" of the Internet Bubble

Consider a simplified, and a bit exaggerated, example of an IT consulting company with sales department, consulting department, and HR department. The company uses a "usual" business model of charging per hour based on the expert level of its consultants.

Sales department conducts sales process instances according to the sales BPT (definition), part of which is hourly pricing of consulting services according to the level of the consultants expertise. HR department is hiring consultants according to the hiring BPT (definition), part of which is salary ranges according to the experience and education of the consultants. Hiring is done to ensure growth or just compensate natural lost.

Suppose management discovers that HR hires fewer consultants than expected. An investigation shows that the reason is the company offering less competitive salaries than their competitors. The hiring BPT is revised and salary offers become higher. The rate of hiring returns to normal (expected).

Suppose that at the same time new sales started to decline. An investigation shows that the company chargers more per hour than their competitors. The sales BPT is revised and the company starts charging less per hour. The rate of getting new consulting assignment returns to normal (expected).

The above adjustment can go through several cycles until the "strategic sensor" catches a new trend: the revenues from new assignments do not cover costs for their completion. A strategic business process instance is fired to find the best leverage point(s) to solve the problem using some System Thinking technique. What would it be?

What happens if the strategic sensor discovers the above situation too late, or there is no strategic BPT at all, or the fired BPI is unsuccessful. Well, bad luck, the company goes out of business.

Does everything above sounds too simplified? It might be so, but in the late 1990s a lot of start-up Swedish IT consultancies operated in this manner until most of them went out of business when the IT-bubble burst. We are not stating here that the management of these companies did not know what they were doing, some of them knew. Their actions might have had a more rational behavior, like dumping prices in hope to get rid of the competitors while having enough of risk capital. This, however, did not matter much in the end.

The artificial example above shows that failing to have a proper sensor may result in a complete demise of an enterprise working in a highly dynamic environment. This example reveals the weakness of the traditional enterprise structure, in which each process type engages a separate set of people. Sales BPIs are manned with sales staff, hiring BPIs are manned with HR staff, process improvement BPIs are manned with the Process Office staff, strategic BPIs are manned with high-level management. An example of such structure is schematically shown in Fig. 3. This picture is less detailed than the one in Fig. 1. Here, a process circle encompasses both BPI and sensors.

There are two weaknesses in the traditional organization that are revealed by the example above:

- Parts of the systems are separated from each other and thus may easily drift apart destroying the system as was shown in the example in the previous section
- The traditionally built enterprise is vulnerable if it operates in a highly dynamic competitive environment. The whole structure will work fine provided that sensors discover emerging situations fast enough so that the organization have time to adjust. In the highly dynamic environment, the costs of creating such sensitive sensors might be too high to make the whole idea sustainable.

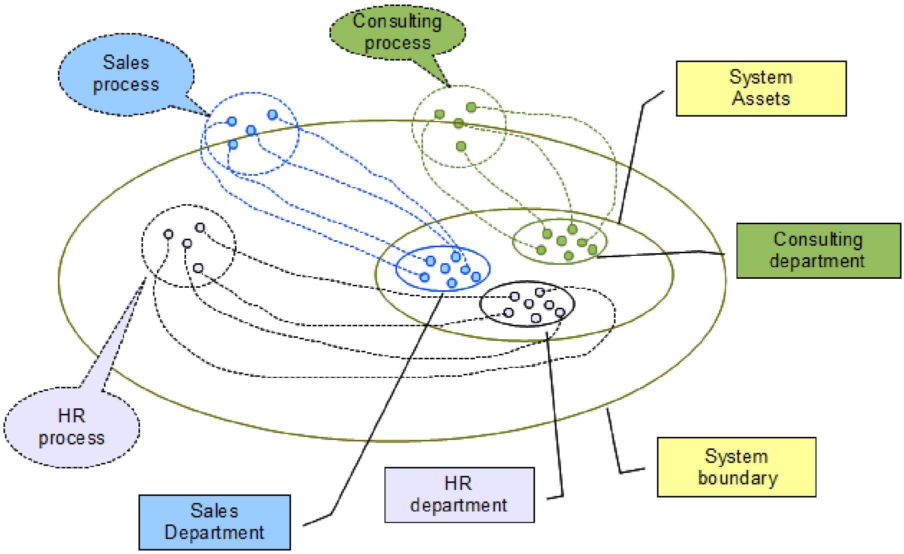


Fig. 3. Traditional enterprise structure from the process perspective

7.2 Selling IT-consultants and Hire a Friend Principles

As a reaction to what happened during the IT bubble (see section 9), the Swedish IT consulting industry adopted two new principles: “selling IT-consultants”, and “hire a friend”. On the conceptual level both principles mean removing the rule that different operational processes, e.g., sales, service delivery, hiring, are manned by different categories of people.

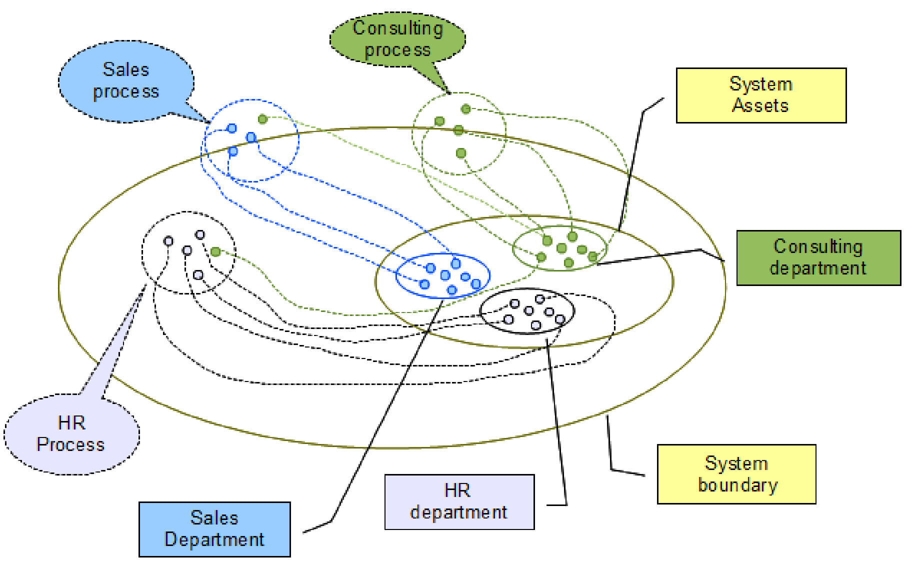


Fig. 4. Cross-manning of business processes

It is not clear whether every IT consultant can be a good sales person, or a good HR person, but in the most cases, due to his/her positioning in the outer world, a consultant can greatly contribute to the sales and HR processes. To make use of his/her position, for example, in sales, the consultant's role in this process can include him/her serving as a part of the process "sensor" by creating leads:

– as a provider of information during the process, and
– as a promoter of the company's line of products and services.

The above does not mean that the sales-staff should disappear and all sales should be conducted by consultants. It means active engagement of other categories of professionals in the sales process on a regular (and not on an ad hoc) basis. Expected results are more sensitive, and less expensive sensors which will make the processes more efficient on the whole.

Selling consultants and hiring a friend represent examples of so-called cross-manning of business processes. A schematic representation of this concept is given in Fig. 4, which is modification of Fig. 3.

7.3 Discussion

An analysis of the examples above shows that the three-layer enterprise model can be applied on the conceptual level to discover and explain weaknesses and strong sides of different ways of organizing business. For example, it helps to understand and explain the essence of cross-manning. Cross-manning can be applied not only to the category 1 operational processes but also to category 2 – improvement processes. Instead of letting specially assigned process re-engineers make detailed process designs, why not let people engaged in these processes do, at least, part of the job themselves? Let process specialists and management devise basic guidelines, and let professionals on the floor fill in the details. The advantages are:

– people on the floor will know sooner when the old process definition stops to satisfy the internal or external environment (more sensitive sensor for the improvement process)
– as they know better not only the business, but also each other's capabilities, they are in a better position to adjust the definitions not only to the abstract goals but also to particular people that man the operational processes

8 Related Research

As was stated in the introduction, we derive our model from reconciliation of Systems and Process Thinking. Therefore, there are many features in our model that can be found elsewhere. We do not claim that all elements of our model are new; only a combination of them is to the best of our knowledge original. Not having enough space to review all related works, we will focus our attention only on two models that we consider most related to ours. One is the Viable Systems Model [11], [12] and the other is System-Coupling Diagrams [13].

The Viable Systems Model (VSM) is presented in Fig. 5. It was developed by Stafford Beer in the 1970s [11] and is modeled after the manner in which a human being functions while interacting with its environment. A viable system has five interacting subsystems that may be mapped onto aspects of organizational structure. Systems 1-3 are concerned with the daily operations of an organization. System 4 is concerned with the strategical responses to the effects of external, environmental and future demands on the organization. System 5 is responsible for policy decisions within the organization as a whole to balance demands from different parts of the organization and steer the organization as a whole. System 5 is responsible to maintain the identity of the organization in order to promote a balance between stability and change. Any system of these 5 is also a viable system, which allows recursive decomposition of the model.

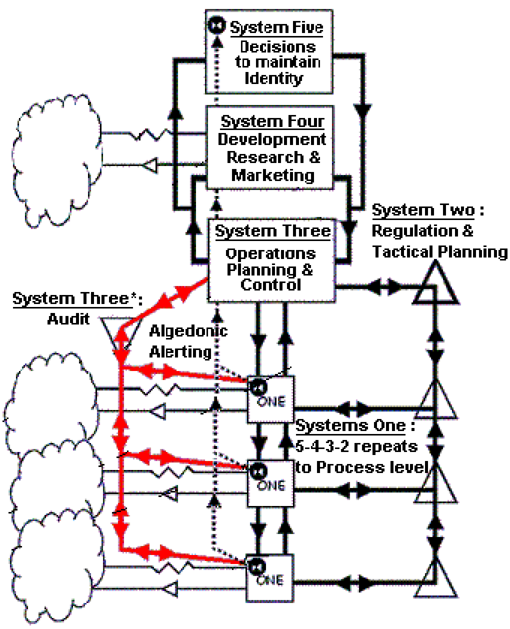


Fig. 5. Viable Systems Model from Wikipedia [12]

Besides systems 1-5, VSM includes Algedonic Alerts (from the Greek αλγος, pain and ηδος, pleasure) – e.g., alarms and rewards that escalate through the levels of recursion when actual performance fails or exceeds capability, typically after a timeout. It is the inclusion of the Algedonic Alerts within VSM which provides a strong relation to the three-layered enterprise model presented in this paper. There would exist different sets of BPTs and sensors for each system within the viable systems model resulting in BPI initiation when the sensors fire. It should also be realized that the BPTs within each system will actually relate to the operational, process improvement and strategic, dimensions as described in Section 4. While this is the case, based on the functions assigned to the different systems within VSM, some systems will have more of one class of BPTs than another. For example, system 5, probably, will have more strategic BPTs than other systems.

The idea of System-Coupling Diagrams comes from [13] and is represented in Fig. 6. It describes a general case when a particular situation in the system's environment causes a larger system, e.g., an enterprise, to create a respondent system, e.g., a project, to handle the situation. The respondent system is built from the assets the larger system already has. We used this idea in our work by interpreting business process instances as respondent systems. In addition to that, we have introduced sensors to discover situation that require building respondent systems. Execution rules, and start conditions of business process diagrams can be considered as control elements of respondent systems and sensors. The notation of Fig.1-4 was inspired by System-Coupling Diagrams.

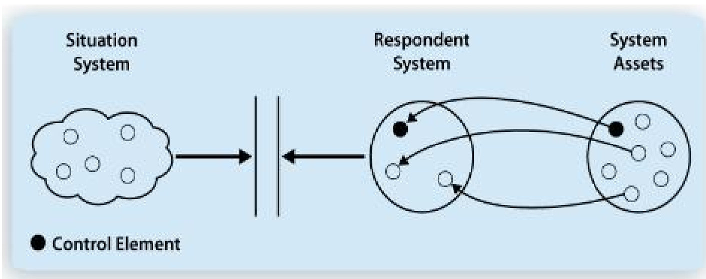


Fig. 6. System Coupling Diagrams from [13]

Business Process Management itself has a movement towards more flexible (agile) processes, which includes, but is not limited to the Adaptive Case Management, and flexible workflows. We do not overview the works related to these issues here, as they concern flexibility/agility in the frame of a particular process, not in the frame of the whole enterprise.

9 Conclusion

As follows from the title, our three-layered enterprise model is being built based on the reconciliation of Systems and Process Thinking. From Systems Thinking comes the view of the business process as a system, and the idea of cross-manning of business processes which creates a tighter cooperation between the parts of the enterprise system. From Process Thinking comes the need of having standardized ways of handling typical situations. Not having them for an organization means that any simple impulse from the outside needs to be processed in an ad hoc manner. An organization without standardized processes can be compared to a person that needs to think how to make each next step when walking along the street. Lack of standard processes is one of the weaknesses of small enterprises that make it difficult for them to compete with the big ones.

This paper begins with the saying “All models are wrong, ...”, which is true for our three-layered model as for any other. However, we hope that this simplified model might actually reveal the level of agility of an enterprise and help in finding ways for its improvement. In other words, we believe the model might be useful for both analysis of the current situation in an enterprise, and for re-designing the enterprise.

When considering business processes, more attention needs to be paid to having good sensors than having optimal processes. There is no need to have an optimal process for handling a customer order if we do not have a sensitive sensor that identifies who might need our products. In the same way, it does not make much sense to have a perfect product development process if we do not have a good sensor that can discover the needs for a new product before our competitors. The concept of cross-manning discussed in Section 7, seems to be one of the principles that could be useful in creating sensitive centers.

The model we suggested may seem to be totally event-driven, the system only reacts on external events. This may be so only if sensors are constructed in the way that they react to what becomes visible for every-one, e.g., economic crises of 2008. Having sensitive sensors makes the system proactive. The sensors can early on catch the trends that have not yet produced visible results. Applying Systems Thinking [6,7] can help in creating such sensors. Returning to the example of financial crises of 2008, having a sensor based on the H. Minsky financial instability hypothesis [14], could have helped to avoid the severity of the crises.

As far as process optimization is concerned, in the highly dynamic environment, this concept should be taken with caution. The more optimized a process is, the more difficult it will be to change it. The less optimal process that is easier to change can be much more "optimal" in the long run. In the practical plan, we need to move from the totally prescriptive definitions of the process execution rules to the constraint-based definitions - a combination of guidelines and restrictions, which allow and require creativity from the process participants handling the instance/case. In the scientific plan, we need to abandon the idea of a process as a flow of operation or events, and start considering it as a trajectory in a multidimensional state-space [15].

As far as computerized systems to support execution of business processes are concerned, the cross-manning concept requires moving the focus to facilitating collaboration/communication between the members of heterogeneous teams. Translating this requirement into the architecture of business processes support systems, there is a need to reconsider current fixation on the conveyor belt principle (workflow engines) in favor of the construction site metaphor of the shared spaces (such as different social software uses) as suggested in [16,17]. In addition, there is a need to have tools that allow the process participants themselves to design and/or change process definitions and adjust the support system to those changes. An example of such a tool is presented in [18].

For now, our three-layered model is just an idea of creating a new modeling technique, and we cannot present its empirical validation at this moment. Quite a lot of research and practical work is needed to convert it into practical modeling notation, and methodology. The promising thing here is that this can be done stepwise. As was shown in section 7, the underlying thinking can readily be applied for understanding some situations. The first step in creating a methodology can be quite simple - design a technique that helps list all enterprise processes, classify them according to the scheme in Section 4, and describe what kind of a sensor each of them has. Based on this list one can start debating the presence, efficiency and sensitivity of existing sensors before raising the issue of optimization of the processes themselves.

Acknowledgment. The first draft of the material presented in this paper was discussed in the LinkedIn group Systems Thinking World [10]. The authors are grateful to all members of the group who participated in the discussions, as their comments helped us to considerably improve the draft.

References

[1] Box, G.E.P.: Wikiquote, http://en.wikiquote.org/wiki/George_E._P._Box
[2] Weske, M.: Business Process Management. Concepts, Languages, Architecture. Springer, Heidelberg (2010)
[3] Harmon, P.: Business Process Change. A Guide for Business Managers and BPM and Six Sigma Professionals. Morgan Kaufmann Publishers, San Francisco (2007)
[4] Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 12–21. Springer, Heidelberg (2010)
[5] Proper, E., Lankhorst, M.M., Schönherr, M., Barjis, J., Overbeek, S. (eds.): TEAR 2010. LNBIP, vol. 70. Springer, Heidelberg (2010)
[6] Checkland, P.: Systems Thinking, Systems Practice. Wiley, Chichester (1999)
[7] Jackson, M.C.: Systems Thinking: Creative Holism for Managers. Wiley, Chichester (2003)
[8] Malone, T., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., O'Donnell, E.: Towards a Handbook of Organisational Processes. Management Science 45(3) (1999)
[9] Sherehiy, B., Karwowski, W., Layer, J.K.: A review of enterprise agility: Concepts, frameworks, and attributes. International Journal of Industrial Economics 37, 445–460 (2007)
[10] Systems Thinking World group, http://www.linkedin.com/groups/Systems-Thinking-World-2639211
[11] Beer, S.: Brain of the Firm; Allen Lane. The Penguin Press, London (1972)
[12] Viable Systems Model, Wikipedia, http://en.wikipedia.org/wiki/Viable_System_Model
[13] Lawson, H.: A Journey Through the Systems Landscape. College Publications (2010)
[14] Minsky, H.P.: The Financial Instability Hypothesis. The Jerome Levy Economics Institute Working Paper No. 74. SSRN (May 1992), http://ssrn.com/abstract=161024 doi:10.2139/ssrn.161024
[15] Khomyakov, M., Bider, I.: Achieving Workflow Flexibility through Taming the Chaos. In: Patel, D., Choudhury, I., Patel, S., de Cesare, S. (eds.) OOIS 2000 - 6th International Conference on Object Oriented Information Systems, pp. 85–92. Springer, Heidelberg (2000)
[16] Bider, I., Perjons, E., Johannesson, P.: A strategy for merging social software with business process support. In: Muehlen, M.z., Su, J. (eds.) J.1, H.4, D.2. Part IV. LNBIP, vol. 66, pp. 372–383. Springer, Heidelberg (2011)
[17] Bider, I., Johannesson, P., Schmidt, R.: Experiences of Using Different Communication Styles in Business Process Support Systems with the Shared Spaces Architecture. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 299–313. Springer, Heidelberg (2011)
[18] iPB Reference Manual (on-line documentation). IbisSoft (2009), http://docs.ibissoft.se/node/3

Towards a CMMI-Compliant Goal-Oriented Software Process through Model-Driven Development

Alexandre Marcos Lins de Vasconcelos[1], Giovanni Giachetti[2], Beatriz Marín[2], and Oscar Pastor[2]

[1] Centro de Informática - Universidade Federal de Pernambuco
Av. Jornalista Anibal Fernandes s/n, 50740-560, Cidade Universitária, Recife-PE, Brazil
amlv@cin.ufpe.br
[2] Centro de Inv. Métodos de Producción de Software - Universidad Politécnica de Valencia
Camino de Vera s/n, 46022, Valencia, Spain
{ggiachetti,bmarin,opastor}@pros.upv.es

Abstract. The *i** framework is a Goal-Oriented Requirement Engineering (GORE) approach that is widely applied at academic level. However, its application to industrial scenarios is limited. For the application of *i** in concrete software development process, an alternative is to transform the defined requirements models into initial input models to be used by Model-Driven Development (MDD) approaches. However, this does not assure that the resultant development process will be sound enough to motivate real development companies to adopt this GORE solution. To tackle this issue, we propose the alignment of GORE and MDD solutions with software process maturity models, which are strongly adopted and applied by industry. In particular, we have considered an approach that integrates the *i** framework into an industrially-applied MDD solution to obtain a development process (that goes from requirements to the final software code), which is compliant with the CMMI-DEV maturity model.

Keywords: Goal-Oriented Requirements Engineering, *i** framework, Model-Driven Development, Software Process Quality, CMMI.

1 Introduction

Requirement modeling plays a relevant role in software development, since the quality of the requirements has a direct impact on the success of software development projects [10]. Among several approaches for defining requirements, the *Goal-Oriented Requirement Engineering* (GORE) [28][31][32] is one that has a wide application spectrum. In general terms, GORE focuses on obtaining the "why" of the intended systems through the analysis of organizational scenarios. It is concerned with the use of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements.

However, as stated in [5], a Requirements Engineering (RE) approach is not useful per se, it must be appropriate for the software process into which it is integrated.

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 253–267, 2011.

Thus, a GORE approach must be properly integrated into a full software process (from requirements to the final code). An alternative to achieve this integration is to automatically transform the requirements models into an initial model [1][8] to be used as input in the context of a Model-Driven Development (MDD) approach [26][33]. Then, this initial model can be refined to automatically generate code through a model compilation (transformation) process. An additional advantage of integrating GORE with MDD is that, through the automatic generation of code from models, MDD allows lower development costs, higher productivity, portability, interoperability, ease of software evolution, and software quality improvement [16].

Among the existing GORE approaches, the *i** framework [35] is one of the most widespread and used at research level [36]. However, there is a gap between the vast application of *i** in academy in relation to its application to real (industrial) development scenarios [34]. An alternative to obtain a suitable support for the application of the *i** framework into real scenarios is to align *i**-based development processes with a software process maturity model [22][30], such as the CMMI-DEV (Capability Maturity Model Integration for Development) [26]. In this way, this kind of GORE solutions become more attractive for the companies that are using those maturity models as the basis to improve their development processes in order to become more competitive in terms of quality and maturity of their processes.

As GORE, MDD, and process maturity models are focused on achieving/increasing the quality of the software product, we believe they are rather complementary. In this context, the following research question is proposed: *How can be designed a GORE-based MDD process to fulfill the requirements of a software process maturity model?* Since this challenge has not been properly addressed by any software process yet, research into this area is relevant and necessary.

Towards answering the research question, in this paper we propose a Goal-Oriented software process (hereafter called GO-MDD) based on the *i** framework and on OO-Method (an industrially applied MDD approach) [20], which is compliant with the requirements development (RD) process area (PA[1]) of CMMI[2]. We have focused on the compliance with RD because, similarly to *i**, the main objectives of this process area are the elicitation and/or specification of system requirements.

The contribution of this paper is twofold. First, practitioners that follow or plan to follow a maturity model and at the same time want to combine GORE and MDD can adapt this proposal instantiating it to their specific needs. Second, the paper can be useful in academia as reference for further research on combining different instances of GORE, MDD, and software process maturity models.

The rest of this paper is organized as follows: Section 2 presents relevant background. Section 3 describes the GO-MDD process. Section 4 presents the analysis of GO-MDD in regard to the RD process area. Section 5 discusses relevant related works. Finally, Section 6 shows the conclusions and proposes future work.

---

[1] A cluster of related practices that, when implemented collectively, satisfies a set of goals for making improvements in an area.

[2] In this paper, the terms CMMI and CMMI-DEV are used as synonyms.

2 Background

The reasons for choosing *i**, OO-Method, and CMMI as the basis for the GO-MDD proposal are the following: *i** is currently one of the most widespread GORE modeling and reasoning frameworks [28][32]; OO-Method is a MDD approach that has been successfully applied in the software industry [19] and; finally, CMMI is the most frequently adopted software process maturity model [22][30]. This section provides a brief explanation of these software development approaches.

2.1 The *i** Goal-Oriented Requirements Framework

The *i** framework [35] emphasizes the analysis of strategic relationships among organizational actors to capture intentional requirements. An actor generically refers to any unit for which intentional dependencies can be ascribed. Actors are intentional in the sense that they do not simply carry out activities and produce entities, but also have desires and needs. The framework offers two types of models: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

The SD model focuses on external relationships among actors. It includes a set of nodes and connecting links, where nodes represent actors (*depender* and *dependee*) and each link indicates a dependency (*dependum*) between two actors. There are four possible *dependum* elements: goal, resource, task, and *softgoal*. A goal is a condition or state of concerns that an actor would like to obtain. A resource is a physical or informational entity that must be available for an actor. A task specifies a particular way of doing something and can be decomposed into small sub-tasks. Finally, a *softgoal* is associated to non-functional requirements.

The SR model is a detailed view of the SD model that shows the internal actor relationships. In addition to the dependencies that are present in the SD model, the SR model incorporates three new types of relationships: (i) task-decomposition links, which describe what should be done to perform a certain task; (ii) means-end links, which suggest that a task is a means to achieve a goal; (iii) contribution links, which suggest how a model element can contribute to satisfy a *softgoal*.

2.2 The OO-Method MDD Approach

OO-Method [20] is an object-oriented method that allows the automatic generation of the final application code from a conceptual model. It is supported by the industrial tool *OlivaNova* [19] and provides a precise UML-like notation, which is used to specify a Conceptual Schema that describes a system at the problem space level. The development process suggested by OO-Method has two phases (Fig. 1): *Development of a Conceptual Schema* and *Generation of a Software Product*.

The first phase consists of eliciting and representing the essential properties of the information system under study, thereby creating the corresponding conceptual schema. In the second phase, a precise execution model, conformed by a set of compilation patterns, indicates the correspondences between the conceptual schema and pieces of code in a target implementation platform. Thus, the application code is automatically generated for an input conceptual schema.

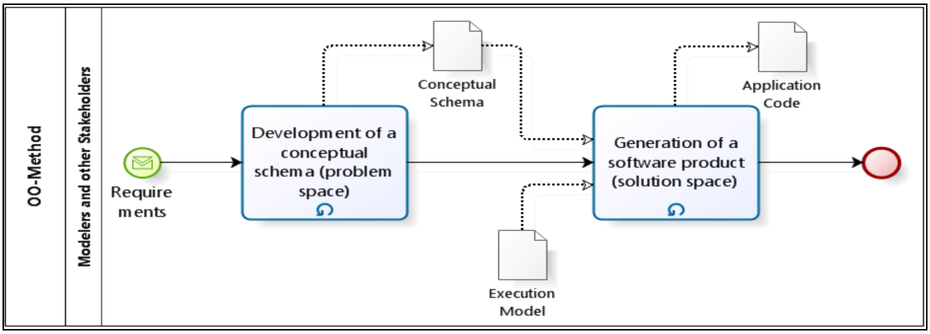


Fig. 1. Phases and artifacts of the OO-Method MDD approach

2.3 CMMI-DEV

CMMI-DEV [26] is a guide to implement a continuous process improvement for developing products and services. For accomplishing this task, it provides two representations: ***Staged***, which assesses the maturity level of a whole development process from an organization; and ***Continuous***, which assesses the capability level of individual process areas (PAs), selected based on the organization's business goals. The process framework described in this paper (see Section 3) is related to the continuous representation, since it is focused on only one process area (PA): requirements development (RD). It complies with the capability level 1 of RD, which is considered, according to CMMI, the basis for improvement initiatives in a specific PA. A meta-model for the continuous representation is presented in Fig. 2.

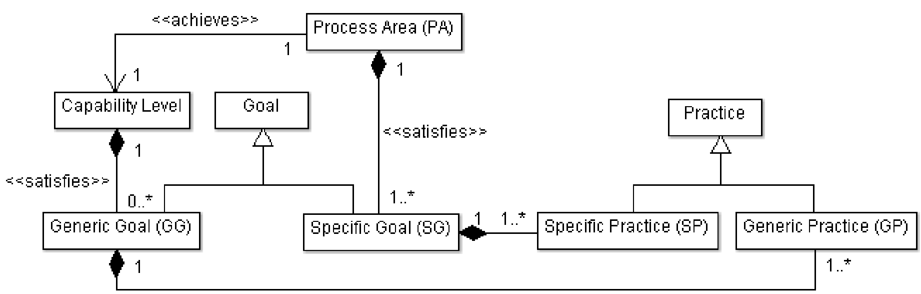


Fig. 2. CMMI Continuous Meta-model, adapted from [17][26]

In the continuous representation, the achievement of a capability level depends on goals and practices (decomposition of goals) of two types: **1)** specific goals (SGs) and specific practices (SPs), which are applied only to a particular PA; and **2)** generic goals (GGs) and generic practices (GPs), which are applied equally to all PAs that achieve a specific capability level. From the assessment of practices and goals, which is performed on a bottom-up way (from the practices up to the goals), it is possible to classify the capability level of a PA on a scale from 0 to 3 (for details see [26]).

3 The Proposed Process Framework: GO-MDD

The process framework, GO-MDD, extends the works published in [1][2][8][9][21]. It is composed of six stages that are performed through an iterative and incremental development cycle (Fig. 3). Thus, after performing all the stages, the cycle can re-start for a new iteration if the product being developed is not finished yet.

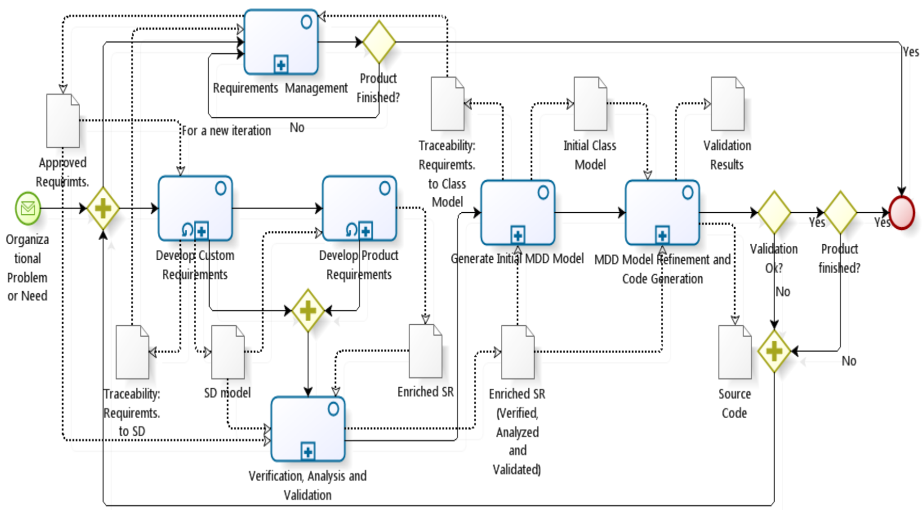


Fig. 3. Stages and artifacts of GO-MDD

We call GO-MDD a process framework and not simply a process because it has a character most descriptive rather than prescriptive, meaning that it focuses on the "what" rather than on the "how" (a detailed discussion on the differences between descriptive and prescriptive processes is presented in [23]). Hence, GO-MDD can be instantiated for each organization prior to its use (e.g., the way of doing the requirements elicitation or their validation).

In particular we apply the CMMI perspective to create a process framework that automatically integrates the *i** framework into a concrete MDD processes (OO-Method). This process framework presents specific transformation guidelines, *i** extensions, and verification mechanisms to assure the correct generation of initial MDD models from *i** models. Then, by means of refinement of this MDD model, a fully executable application that is aligned with the stakeholder requirements is generated. Since the proposed integration of *i** and MDD is based on the class model generation, the results presented in this paper can be used as reference for other object-oriented MDD processes, such as UML-based proposals.

An example, related to the management of work requests in a Photography agency, extracted from the experiment presented in [9], is used to explain the process framework GO-MDD. The Photography agency is dedicated to the management of photo reports and their distribution to publishing houses. This agency operates with freelance photographers, which must present a work request to its production department.

Due to space constraints, only those aspects that are not part of *i** and are more relevant to the paper's objectives are illustrated.

First Stage: Develop Custom Requirements. Requirements from various stakeholders are consolidated, prioritized (according to stakeholder needs and constraints) and detailed to be implemented in the current iteration. These requirements can be elicited in the current iteration or come from a backlog (i.e., a list) of previously approved requirements (see the description of the *Second Stage*). The requirements backlog and the iterative development cycle were inspired from *Scrum* [25]. As the result of this stage, an SD model is produced and the traceability (i.e., a mapping) from requirements to the SD model is created/updated.

Second Stage: Requirements Management. This stage is responsible for monitoring the requirement requests from several stakeholders and performing initial requirements elicitation and analysis (of adequacy and impact) in order to decide whether the requests will be approved (or not) to be developed in some iteration. It is executed in parallel with the first stage, until the product is satisfactorily produced. As the result of this stage, a backlog of approved requirements is created/updated.

Third Stage: Develop Product Requirements. In this stage, detailed in [2][21], an initial SR model is produced from the refinement of the SD model. The goals defined in the SR model are analyzed to decide the intentional elements that must be considered as requirements of the system to be. These elements are highlighted by means of specific stereotypes, which introduce information to automatically perform the corresponding MDD model generation. Thus, an enriched SR model is produced.

Fig. 4 shows an example of an *i** SR model, related to the Photography agency, extended with the stereotypes defined for the integration with OO-Method. This SR model shows that the *production department* depends on the reception of *work requests* (i.e., job applications), which are produced by *photographers* that want a *work opportunity*. The *work requests* are comprised by the photographer's *personal data*. The *production department* is responsible for *refusing* or *accepting* the *received work requests* by indicating the final *work request status*. For the accepted requests, a *photographer level* is assigned according to the information provided by the *Commercial Department*. The stereotypes that extend the *i** SR model introducing specific information to generate the corresponding MDD (class) model according to the OO-Method approach, and their main application to the transformation process are briefly described as follows (further information can be found in [2]):

- **SActor:** Indicates that an *i** actor must be maintained by the corresponding system. This actor will be represented by means of a class in the generated MDD model.
- **SPhysicalR:** Indicates that an *i** resource is considered as a physical resource that must be maintained in the system as a class in the generated MDD model.
- **SInfoR:** Indicates that an *i** resource is considered as an informational resource that must be maintained in the system as class attribute in the generated MDD model.
- **STask:** Indicates that an *i** task must be considered for the system behavior. This will be represented as a class service in the generated MDD model.

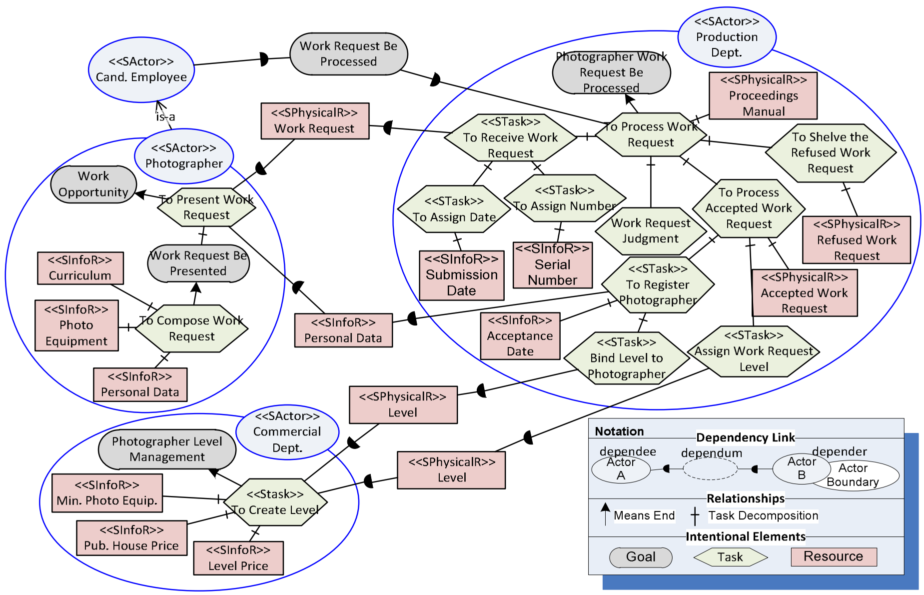


Fig. 4. Example of extended *i** SR model

Fourth Stage: Verification, Analysis, and Validation: The models defined in the previous stages are used as input for this stage. An analysis is performed to guarantee that the requirements defined in the SR model are necessary and sufficient to meet the organizational goals and to balance stakeholder's needs and constraints. Then, the requirements are validated with the stakeholders to guarantee their correctness and completeness and, if any problem is detected in the analysis or in validation, it must be fixed. Later, the resultant SR model is verified by means of a set of measures (detailed in [9]), which evaluate the elements extended. These verification measures are formally specified by means of OCL rules [18] and guarantee the completeness of the MDD model generation in relation to the requirements indicated in the *i** model. This has been demonstrated by means of the controlled experiment presented in [9]. The measures not only identify the modeling issues, but also provide fixing guidelines to improve the SR model and the MDD model generation. Hence, if some problem is detected during the model verification, it must be fixed before getting to the next stage. For instance, the measure *Wrong Attribute Generation* (Table 1) specifies that an *i** resource stereotyped as an informational resource (SInfoR) must be related to a system actor (SActor) or to a physical resource (SPhysicalR), which are transformed

Table 1. Some characteristics of the measure *Wrong Attribute Generation*

Characteristic	Definition
Measurement Scale	Ratio scale
Attribute to be measured	Informational resources not related to a physical resource or to an actor.
Measurement principle	This kind of informational resource corresponds to a wrong attribute generation in the MDD model.
Measurement procedure	The attributes to be measured must be counted to obtain the number of informational resources that cannot be transformed into attributes.

into a class. Otherwise, the informational resource cannot be transformed into a class attribute due to the lack of a class that contains it.

Fifth Stage: Generate Initial MDD Model: Once the *i** SR model has been verified and improved according to the corresponding verification measures, it is transformed into an initial MDD model (class model) by means of a set of model-to-model transformations (detailed in [2] and [21]). We refer to an initial MDD model and not a complete one because there are aspects related to specific system functionality that cannot be obtained from requirements models.

Traceability from requirements to the MDD model is also produced in this stage. It is important to point out that the class model is the central model in the OO-Method approach. The rest of the models that are necessary to completely specify the OO-Method conceptual model (such as the presentation or the functional model) are derived from this central model. Thus, the traceability from requirements to the other OO-Method models can be obtained from the association of requirement elements to the corresponding class model elements. Fig. 5 shows the class model obtained from the extended *i** model presented as example in Fig. 4.

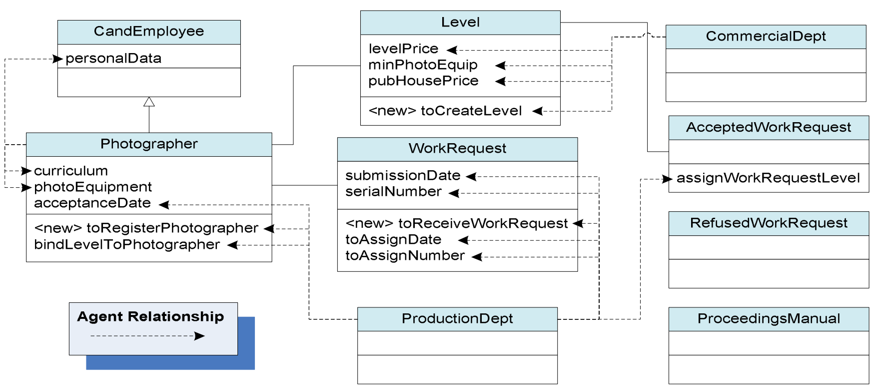


Fig. 5. Initial class model generated from the extended *i** SR model

A specific OO-Method construct that differs from the traditional (UML-like) class model notation can be observed in the generated class model. This is the agent relationship, which indicates the visibility that a class has over attributes or services of other classes of the model. Agent relationships are defined between a class generated from an *i** actor and the elements generated from *i** elements that are inside the boundary of the actor transformed (e.g., the agent relationship that is defined from the class *ProductionDept* to the service *toReceiveWorkRequest*). These relationships provide relevant information for defining presentation models related to the specification of users' interactions with the final system.

Sixth Stage: MDD Model Refinement and Code Generation. The initial MDD model generated is refined to introduce those design aspects that cannot be obtained from the transformation of the enriched *i** SR model. Some of the refinements that must be performed are the specification of additional class services, association of cardinalities, or specific system constraints. The refined model is verified by means of a facility provided by the OO-Method modeling tool [19], which guarantees the correct specification of the MDD model for the automatic code generation. The system source code is generated by applying the OO-Method model compilation technology. Finally, the generated source code must be compiled and executed in order to be validated against the corresponding requirements.

3.1 Research Method for the Design of GO-MDD

The design of GO-MDD was done according the following research method. First, based on previous works of our research group, regarding the integration of *i** and OO-Method [1][2][8][9][21] (that propose the use of stereotypes, transformation guidelines and verification measures), an initial version of the process framework was specified. The compliance of this version was analyzed against the RD process area and, based on the gaps identified in the analysis, new characteristics, activities and artifacts were included into GO-MDD to make it fully compliant with this process area. Finally the compliance mapping, described in the next section, was produced. No exclusions were done from the original process, and the inclusions were mainly related to the consolidation, prioritization and traceability of requirements (*First Stage*); the whole Requirements Management (*Second Stage*); the analysis and validation of requirements (*Fourth Stage*); and the definition of an interactive/incremental cycle (performed along the whole process framework).

4 Compliance Mapping from the RD Process Area and GO-MDD

According to CMMI-DEV, the purpose of RD is to elicit, analyze and establish customer, product, and product component requirements. A compliance mapping between the capability level 1 of RD and the proposed process framework (GO-MDD) is presented in this section. For each SG, its purpose is described, and for all the corresponding SPs, a mapping relating stages, activities, and artifacts of GO-MDD to each SP is produced. To comply with the capability level 1, a process must satisfy the generic goal (GG) associated to this level (GG 1), which has only one generic practice (GP 1.1) that requests all the SGs associated to the PA to be satisfied (if at least one of the SGs is not satisfied, the PA is considered to have capability level 0). For evaluating capability levels higher than 1, a PA must satisfy the GG associated to the specific level, which imposes other requirements, and all the GGs associated to the lower levels. Fig. 6 instantiates the meta-model presented in Fig. 2 to represent the elements involved in the compliance mapping for level 1.

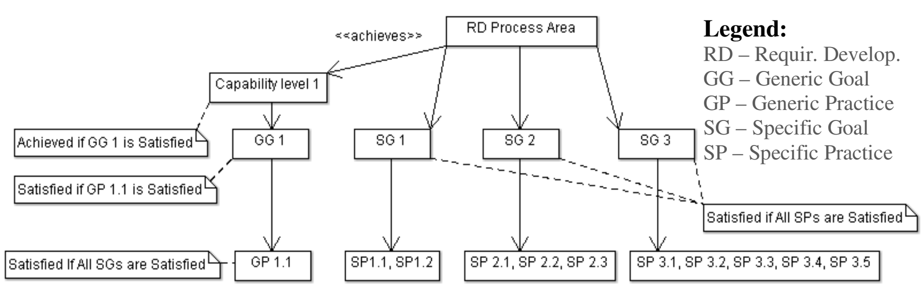


Fig. 6. The capability level 1 for the RD process area

SG 1 Develop Customer Requirements: This goal addresses the collection of stakeholder needs, expectations, constraints and interfaces, and their translation into customer requirements.

SP 1.1 Elicit Needs: requirements should be elicited and a requirements specification (e.g., a textual document or a model) should be produced. The requirements specified (functional and non-functional) express stakeholder needs, expectations, constraints, and interfaces for all the phases of the product lifecycle.

Compliance mapping: A preliminary requirements elicitation is performed in the *Requirements Management* stage and a list of approved requirements is produced. Then, in the *Develop Custom Requirements* stage, the requirements to be implemented in the current iteration are detailed, and an initial *i** model (SD) is produced with the definition of different organizational actors (stakeholders) and their dependencies.

SP 1.2 Transform Stakeholder Needs into Customer Requirements: Requirements elicited from various stakeholders (including business and technical functions) should be consolidated, analyzed regarding missing information and presence of conflicts, and prioritized according to some criteria. Requirements specific to verification and validation (V&V) for the system to be can also be elicited.

Compliance mapping: The SD model produced in the *Develop Custom Requirements* stage is a consolidated and prioritized specification of the needs from various stakeholders.

SG 2 Develop Product Requirements: This goal addresses the refinement and elaboration of customer requirements in order to develop product and product component requirements. Some of the practices associated to this goal can be performed during or in conjunction with a design stage.

SP 2.1 Establish Product and Product Component Requirements: product and product component requirements should be derived (identified) from customer requirements. Product requirements are functional and non-functional requirements expressed in technical terms that can be used for design decisions. Modifications on customer requirements due to approved requirements changes must be reflected in the derived requirements. Derived requirements also address the needs of other lifecycle phases (e.g., production, operations and disposal).

Compliance mapping: The enriched SR model produced in the *Develop Product Requirements* stage is a refinement of the SD model and specifies which requirements (functional and non-functional) are allocated to the products and product components to be developed. Modifications on customer requirements are captured in the *Requirements Management* stage, and their impacts on the derived requirements are analyzed based on the traceability from requirements to the SD model (produced in the *Develop Custom Requirements* stage), on the refinement relationship between the SD and the SR models, and on the traceability from requirements to class model (produced in the *Generate Initial MDD Model* stage).

SP 2.2 Allocate Product Component Requirements: The product components requirements (functional and non-functional) should be allocated to product components of the defined solution.

Compliance mapping: In the *Generate Initial MDD Model* stage, the initial class diagram generated is a first approximation for the allocation of requirements to product components. Later, this allocation can be updated when the class diagram is refined in the *MDD Model Refinement and Code Generation* stage.

SP 2.3 Identify Interface Requirements: Interface requirements between functions, objects or other logical entities should be identified.

Compliance mapping: The initial class diagram, produced in the *Generate Initial MDD Model* stage, includes classes, methods, attributes, and associations. It defines the interfaces among entities that are identified from the requirements. Later, these interfaces can be updated when the class diagram is refined for code generation.

SG 3 Analyze and Validate Requirements: This goal addresses requirements analysis and validation. Its specific practices support the development of the requirements in SG 1 and SG 2. Some of the practices associated to this goal can be performed during or in conjunction with a design stage.

SP 3.1 Establish Operational Concepts and Scenarios: Operational concepts and scenarios should be identified and maintained (i.e., updated when necessary). Operational concepts are general descriptions of the ways in which entities are used or operate. Scenarios are detailed sequences of events that make explicit some of the functional or quality attribute (non-functional) needs of the stakeholders.

Compliance mapping: The SD and SR models illustrate the tasks and subtasks (i.e., the way of doing something) related to the satisfaction of goals that the actors would like to achieve. In particular, the enriched SR highlights the tasks and subtasks related to processes that will be automated.

SP 3.2 Establish a Definition of Required Functionality and Quality Attributes: a definition of the required functionality and quality attributes should be established and maintained.

Compliance mapping: The enriched SR in conjunction with the initial class diagram specify the quality attributes (*softgoals*) and required functionalities (tasks and class services) that are related to the requirements elicited. Later, when the class diagram is refined, these quality attributes and functionality can be updated.

SP 3.3 Analyze Requirements: The requirements for one level of the product hierarchy should be analyzed to determine if they are necessary and sufficient to meet the objectives of higher levels (i.e., the practice analyses the consistence between requirements in different levels of hierarchy).

Compliance mapping: Satisfaction of this SP is discussed together with the next one.

SP 3.4 Analyze Requirements to Achieve Balance: Requirements should be analyzed to balance stakeholder's needs and constraints, such as cost, schedule, product or project performance, functionality, priorities, reusable components, maintainability, and risks.

Compliance mapping: The SP 3.3 and SP 3.4 are satisfied in the following way. Preliminary requirements analysis is performed during the *Requirements Management* stage and also in the *Develop Custom Requirements* stage, when requirements are consolidated and prioritized according to stakeholder needs and constraints. In the *Verification, Analysis, and Validation* stage, the consistence among requirements from different levels of hierarchy is analyzed. Additionally, the defined measures automatically verify the *i** models in the context of the MDD process to assure the transformation completeness of the necessary requirement elements. Thus, the MDD model generated provides a complete representation (at design time) of all the artifacts defined at requirements level.

SP 3.5 Validate Requirements: Requirements should be validated to ensure that the resulting product will perform as intended in the end-user environment.

Compliance mapping: Prior to the generation of the initial class model, in the *Verification, Analysis, and Validation* stage, the requirements are validated with the stakeholders in order to guarantee their correctness and completeness. Also, the iterative and incremental refinement cycle that is present in our proposal allows automatically generated class models to be used to generate prototypes, which are validated by the stakeholders to assure the correct implementation of their needs. Hence, at each iteration, an initial MDD (class) model can be automatically generated from the requirements and later refined to obtain a complete and precise description of the generated MDD elements in order to perform the model compilation into an increment towards the final software product. Thus, necessary changes in the requirements are introduced in the defined *i** models to generate a new version of the corresponding class models, and to perform a new model compilation and validation.

5 Related Work

In the literature, we have not found any work that proposes a software process based on the automatic integration of GORE with MDD and compliant with a maturity model. However, there are works which treat the pair-wise association of these software development approaches. Some of these works are described as follows.

The integration between GORE and MDD has been discussed in several works. However, most of these works (such as [12][14][24]) are not based on standards or well-defined processes, nor do they introduce automation possibilities. Therefore, the application of these proposals must be manually performed [15], which is not a suitable option since the manual translation of models is a time consuming and error prone task [13].

In relation to the integration of GORE with a software process maturity model, in [3] it is proposed an approach for requirements development and management in the context of system family engineering, which, according to the authors, complies with the RD and RM[3] process areas of CMMI. In this approach, high abstraction level *goals* (related to functional aspects) and *softgoals* (related to quality aspects) are the first means used to elicit requirements. However, as opposed to our work, an explicit mapping identifying which are all the evidences to attest its compliance with CMMI is not presented; neither the approach is integrated into a complete software process from requirements to code.

[3] The Requirements Management process area is responsible for tracking requirements changes, analyzing the impacts of the changes and maintaining the requirements traceability.

Although there are some specific works related to the compliance of MDD approaches with CMMI or its ancestor (CMM) [4][6][7][29], they fail to deal with this issue properly. These works do not explain in detail how an approach complies with the maturity model, where the approach should be adjusted for compliance, and whether/where the approach conflicts with the maturity model requirements.

Hence, unlike those previous works, we proposed a complete software process framework, which integrates GORE and MDD through automatic transformation from requirements to initial design models and compliant with a software process maturity model. To demonstrate the adherence of the GO-MDD process framework with the maturity model, a detailed compliance mapping was presented.

6 Conclusions and Future Work

In this work, based on the research question *How can be designed a GORE-based MDD process to fulfill the requirements of a software process maturity model?*, we advocated that GORE and MDD can be put together to comply with the requirements of a software process maturity model, thus supporting the application of a GORE approach into real (industry) scenarios. In order to demonstrate the soundness of this idea, we have proposed a software process framework based on *i** and OO-Method, and have described how it complies with the RD process area of CMMI-DEV. Even though the work has been based on specific instances of GORE, MDD, and a software process maturity model, we believe that this idea can be generalized to other instances, but further research need to be done in this direction.

This proposal is part of a wider work that is related to the use of GORE and MDD to define a full software process, compliant with a software process maturity model, covering the long path that goes from goal-oriented requirements modeling to a final high-quality software product. In this context, several future works can be developed: **1)** Extending the proposed process framework to be compliant with other process areas and capability levels of CMMI is necessary. Despite the framework proposed already presents some characteristics that partially meet other PAs, such as "requirements management" and "project monitoring and control", additional characteristics must be considered to be compliant with these and further PAs; **2)** We are aware that the evaluation of the proposal in real development scenarios is necessary. Hence, we consider as future work the development of empirical studies to validate the feasibility and the effectiveness of the proposed process framework, and to analyze the practical implications of its use in an industrial context; **3)** The investigation of the research question in the scope of other software development approaches (i.e., different GORE, MDD and/or maturity models approaches) needs to be done; **4)** Finally, a systematic literature review [11] should be conducted to verify in deep the existence of other related works.

Acknowledgments. This work has been developed with the support of the Brazilian Research Agency CAPES, under the grant #BEX3229/10-6; and the Spanish Government, under the projects ORCA (PROMETEO/2009/15), PROS-REQ (TIN2010-19130-C02-02) and PTA2008-1443-P (co-financed by ERDF).

References

1. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., Pimentel, J.H.: From i* Requirements Models to Conceptual Models of a Model Driven Development Process. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 99–114. Springer, Heidelberg (2009)
2. Alencar, F.M.R., Pastor, O., Marín, B., Giachetti, G., Castro, J.: Aligning Goal-Oriented Requirements Engineering and Model-Driven Development. In: Proc. of 11th International Conference on Enterprise Information Systems (ICEIS), pp. 347–350 (2009)
3. Cerón, R., Dueñas, J.C., Serrano, E., Capilla, R.: A Meta-model for Requirements Engineering in System Family Context for Software Process Improvement Using CMMI. In: Bomarius, F., Komi-Sirviö, S. (eds.) PROFES 2005. LNCS, vol. 3547, pp. 291–313. Springer, Heidelberg (2005)
4. Crag Systems: The Model-Driven Development Process (2008), `http://www.cragsystems.co.uk/development_process` (last access June 30, 2011)
5. de la Vara, J.L., Sánchez, J.: System Modeling from Extended Task Descriptions. In: Proc. of SEKE 2010, pp. 425–429 (2010)
6. ESI. Center: Model-driven Architecture inSTrumentation, Enhancement and Refinement. IST-2001-34600, MASTER-2003-D3.2-V1.0-PUBLIC (2003)
7. Fricker, S.: Introducing Model-Driven Development for CMMI Engineering Process Areas. In: 18th Software Engineering Process Group Conference (SEPG 2006), `http://www.secc.org.eg/sepg%202006/ingredients/Indexes/authorindex.html#f`, (last access June 30, 2011)
8. Giachetti, G., Alencar, F., Marín, B., Pastor, O., Castro, J.: Beyond Requirements: An Approach to Integrate *i** and Model-Driven Development. In: Proc. of XIII Conferencia Iberoamericana en Software Engineering, CibSE 2010 (2010)
9. Giachetti, G., Alencar, F., Franch, X., Marín, B., Pastor, O.: Technical Report ProS-TR-2011-07: Automatic Verification of Requirement Models for Their Interoperability in Model-Driven Development Processes. Universidad Politécnica de Valencia (2011)
10. Kamata, M.I., Tamai, T.: How Does Requirements Quality Relate to Project Success or Failure? In: Proc. of RE 2007, pp. 69–78. IEEE, Los Alamitos (2007)
11. Kitchenham, B., Charters, S.: Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3. Keele University and Durham University Joint Technical Report EBSE-2007-01 (2007)
12. Liu, L., Yu, E.: Designing Information Systems in Social Context: A Goal and Scenario Modeling Approach. Info Systems 29(2), 187–203 (2004)
13. Maiden, N., Jones, S., 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. 368–383. Springer, Heidelberg (2004)
14. Martínez, A., Castro, J., Pastor, O., Estrada, H.: Closing the Gap between Organizational Modeling and Information System Modeling. In: Proc. of WER 2003, pp. 93–108 (2003)
15. Martínez, A.: Conceptual Schemas Generation from Organizational Models in an Automatic Software Production Process. PhD Thesis. Univ. Politéc. de Valencia (2008)
16. Mohagheghi, P., Dehlen, V.: Where Is the Proof? - A Review of Experiences from Applying MDE in Industry. In: Schieferdecker, I., Hartman, A. (eds.) ECMDA-FA 2008. LNCS, vol. 5095, pp. 432–443. Springer, Heidelberg (2008)

17. Monzón, A.: AQAP-160 vs. CMMI. Slides Presentation. Mil. Transport Aircraft Division (2006), http://www.calidaddelsoftware.com/documentos/II%20Semana%20CMMI/03-%20EADS-CASA.pdf (last access June 30, 2011)
18. OMG: Object Constraint Language 2.2 Specification (2010)
19. Pastor, O., Molina, J.C., Iborra, E.: Automated Production of Fully Functional Applications with OlivaNova Model Execution. ERCIM News, nº 57 (2004)
20. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling, 1st edn. Springer, New York (2007)
21. Pastor, O., Giachetti, G.: Linking Goal-Oriented Requirements and Model-Driven Development. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 257–276. Springer, Heidelberg (2010)
22. Pino, F.J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Soft. Quality Journal 16(2), 237–261 (2008)
23. Ruhe, G.: Learning Software Organizations. Fraunhofer Institute for Experimental Software Engineering, ftp://cs.pitt.edu/chang/handbook/09.pdf (last access June 30, 2011)
24. Santander, V., Castro, J.: Deriving Use Cases from Organizational Modeling. In: Proc. of RE 2002, pp. 32–42 (2002)
25. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Microsoft Press, Redmond (2004)
26. SEI: CMMI for Development, Version 1.3, CMU/SEI-2010-TR-033 (2010), http://www.sei.cmu.edu (Last access June 30, 2011)
27. Selic, B.: The Pragmatics of Model-Driven Development. IEEE Software 20(5), 19–25 (2003)
28. Shuichiro, Y., Haruhiko, K., Karl, C., Steven, B.: Goal Oriented Requirements Engineering: Trends and Issues. IEICE – Trans. Inf. Syst. E89-D, 2701–2711 (2006)
29. Steinhau, R., et al.: Guidelines for the Application of MDA and the Technologies covered by it. Deliverable 3.2, MODA-TEL Consortium, IST-2001-37785, Interactive Objects Software GmbH (2003)
30. Unterkalmsteiner, M., et al.: Evaluation and Measurement of Software Process Improvement – A Systematic Literature Review. IEEE TSE (2011)
31. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: a Guided Tour. In: Proc. of RE 2001, Springer, Heidelberg (2001)
32. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: a Roundtrip from Research to Practice. In: Proc. of 12th IEEE Joint International Requirements Engineering Conference, pp. 4–8. IEEE Computer Science, Los Alamitos (2004)
33. Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S., Czarnecki, K.: Model-Driven Software Development: Technology, Engineering, Management. Wiley, Chichester (2007)
34. Xavier, F.: Fostering the Adoption of *i** by Practitioners: Some Challenges and Research Directions. In: Intentional Perspectives on Information Systems Engineering, pp. 177–193. Springer, Heidelberg (2010)
35. Yu, E.: Modelling Strategic Relationships for Process Reengineering, PhD Thesis, University of Toronto, Toronto, Canada (1995)
36. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering (2011)

Bringing Enterprise Modeling Closer to Model-Driven Development

Iyad Zikra, Janis Stirna, and Jelena Zdravkovic

Department of Computer and Systems Sciences, Stockholm University
Forum 100, SE-164 40 Kista, Sweden
{iyad,js,jelenaz}@dsv.su.se

Abstract. Enterprise Modeling (EM) provides the means for using models to represent organizational knowledge from different perspectives. When information systems (IS) are involved, Model-Driven Development (MDD) is an approach that focuses on the use of models as primary development artifacts. By observing that EM provides the context for high level requirements, which in turn are the input to MDD, we propose a meta-model that integrates enterprise models and requirements with design models in MDD. The meta-model defines six models that cover both organizational and IS development knowledge. Inter-model relationships ensure an integrated view of the enterprise and the supporting IS by allowing model components to be used across different models. The integrated meta-model is demonstrated through an example case study.

Keywords: Enterprise Modeling, Enterprise Models, Requirements, MDD, Model-Driven Development, MDE, Model-Driven Engineering.

1 Introduction

Enterprise Modeling (EM) aims to capture and represent organizational design in terms of business goals, processes, concepts, actors, as well as high level information system (IS) requirements by using conceptual models. Many EM techniques have emerged throughout the years, presenting different views of the enterprise and offering a wide variety of possibilities for designing, improving, re-structuring, and automating all or parts of the business in question.

Model-Driven Development (MDD) is a software development approach where models replace programming code as the primary development artifact. Models in MDD are used for describing the IS design to a level of detail that allows automatic generation of a running system. When used in the context of developing enterprise-level software, MDD has the potential of streamlining the development process. However, the input to the MDD process has to come from a higher level of abstraction, such as organizational designs and requirements, which are often represented by Enterprise Models. Most current MDD approaches implicitly assume that creating the initial MDD model is the responsibility of the modeler and provide little or no guidance for initiating development [22]. The approaches that try to

P. Johannesson, J. Krogstie, and A.L. Opdahl (Eds.): PoEM 2011, LNBIP 92, pp. 268–282, 2011.

establish a connection between preceding modeling efforts and MDD rely on mappings between, e.g. requirements and design models in MDD. They still fall short on achieving full integration and supporting the developers in all development phases.

In this paper, we present an integrated meta-model of MDD and EM, taking Enterprise Knowledge Development (EKD) [2] as an example of a specific EM approach. The aim of this research is to provide a formal connection between the two development activities, thus bridging the gap between designing the organizations and model driven development of information systems. EKD is selected as the candidate EM technique because it includes an overall model composed of inter-related sub-models for integrating different views of the organization. Many concepts in our meta-model are based on the EKD approach. The proposed meta-model is a first step towards a complete and tool-supported MDD process. Combining the high-level business-oriented EKD capabilities with the more concrete and IS-oriented MDD principles can lead to better integration between the business and information systems that support it.

The remainder of the paper is organized as follows: Sections 2 and 3 give a short background to EM and MDD. The proposed meta-model is presented in Section 4 and then demonstrated by an example in Section 5. Section 6 is a discussion of related work, while Section 7 presents concluding remarks and issues for future work.

2 Enterprise Knowledge Development

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. the purpose of the organization, business processes, entities, and structure. For example [12] proposes using the UML notation [14] to model enterprises.

In this research we have chosen EKD as the EM approach to be integrated with MDD. The following six integrated sub-models constitute EKD, each focusing on a different aspect of the enterprise:

Goals Model (GM)–the organization’s vision and strategy; it addresses questions related to what the organization wants to achieve or to avoid and why.

Business Rules Model (BRM)–the business policies and rules; it addresses questions related to business rules and how they support the organization’s goals.

Concepts Model (CM)–the business ontology and vocabulary; it addresses the things and “phenomena” covered in other sub-models.

Business Process Model (BPM)–the procedural aspects of business operations; questions related to business processes and how they handle information and material.

Actors and Resources Model (ARM)–organizational structure; addressing the responsibilities for goals and processes and how the actors are interrelated.

Technical Component & Requirements Model (TCRM)–IS needs; addressing questions about the business requirements for the IS and how they are related to other sub-models of the enterprise model.

The modeling components of the sub-models are related within a sub-model (intra-model relationships) as well as with components of other sub-models (inter-model relationships). Inter-model links are particularly useful for developing a holistic view of the enterprise because they allow tracing decisions and components throughout the model. E.g. goals in the GM provide the reasons for including certain processes in the BPM and IS requirements in the TCRM. EKD strongly advocates the use of a participative approach for model development [2, 20], a technique also applicable in agile development projects [19].

Bubenko et al. [2] identify two purposes for EM: 1) developing the business: shaping the business vision and identifying the strategy and means necessary to attain it, including IS development; and 2) ensuring the quality of the business: creating a unified and shared knowledge culture, and gaining the commitment of different stakeholders. In conjunction with business development, EM often provides input to IS development.

3 Model-Driven Development

The focus of software development shifted from code to models as a response to the increasing complexity of software and the pressure to shorten development cycles [17]. Technology advancements enabled the transformation of models into complete and functional software, making models an essential part of the final product [5]. The emergent model-centric approaches are commonly called MDD. No unified definition exists for this new domain, which is sometimes referred to as Model-Driven Engineering (MDE) in the literature. There are however common characteristics that distinguish approaches that are labeled as model-driven [4, 8, 16, 17]. Models are the central concept in MDD; they present different views of a system and guide the development process. The structure and semantics of models are formalized in MDD as meta-models, enabling automatic creation of models. The knowledge used to create models is also formalized as transformation rules, which are based on the language of the models when the transformed models are in the same language or on the meta-models when the models are in different languages. The process for creating and managing the models, meta-models, and transformations is identified by [8] as part of MDD. Supporting tools are consistently highlighted in the MDD literature as an essential part, even though many MDD approaches have partial or no tool support. [10, 22]

The interest in MDD is growing despite the existence of many unresolved issues and the limited tool support. Creating the initial MDD model has been identified as a problem in [10] and [22]. The problem manifests itself in a gap between IS design models and higher levels of abstraction. Integration properties were proposed in [22] to establish a connection between requirements and MDD. By observing that EM is a practicable instrument for capturing high level requirements needed as the input to MDD, we propose an overarching meta-model that integrates EM and MDD. The next section describes this meta-model.

4 A Unifying Meta-model

The main objective for developing the meta-model is to provide a unified platform for designing enterprise models, which are then used to derive IS models that can subsequently be used to generate a functioning and complete system using an MDD approach. The meta-model defines multiple complimentary models, offering a holistic view of the organization and enabling automatic generation of an IS that is described by the relevant models. The meta-model defines (1) models representing enterprise knowledge (enterprise-level models) based on the EKD approach, namely EKD's GM, CM, BPM, and BRM, as well as (2) system-level models, namely *Requirements Model (RM)* representing IS requirements and *IT Architecture Model (ITAM)* describing the technical components and user interfaces that are involved in the implementation of the IS.

Relationships between the enterprise-level and system-level models are formalized to support evaluation of the system models and improve traceability to their origins in the organizational design. Coarse-grained relationships give a general overview of the interactions between the models (Fig. 1). Fine-grained relationships, called inter-model relationships, relate model components across different models. They depict the use of concepts of one model in other models and present a more complete view of both the enterprise and the IS. Complete traceability is supported by the meta-model without the need to introduce additional explicit traceability links. The fine-grained inter-model relationships are further discussed in the following subsections.

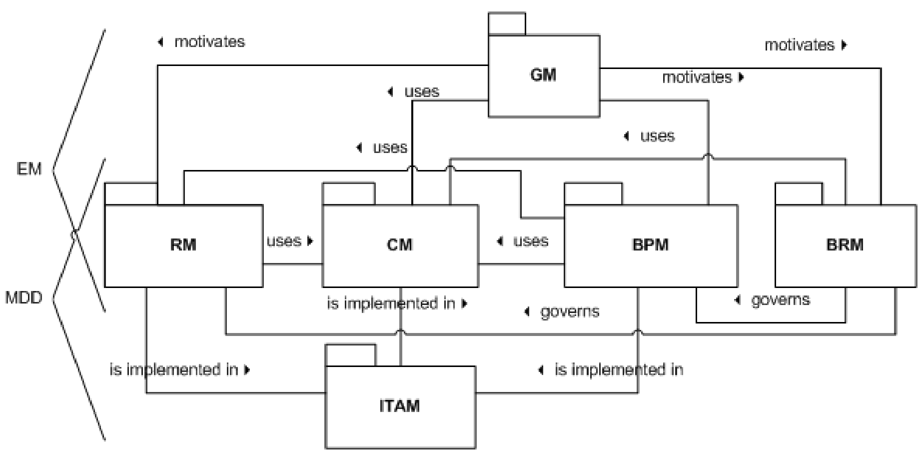


Fig. 1. Relationships between the sub-models in the meta-model

The meta-model in this paper follows a simple UML-like notation. Package symbols are used to denote models in Fig. 1, implying that concepts defined in the meta-model are spread over the six models. Furthermore, generalization links between concepts denote a general-specific relationship, while cardinalities on the relationships express the number of model component instances that can participate in a model relationship instance.

4.1 Common Components Model

All models described by the meta-model are derived from common components that provide the basis for other components. The Common Components Model (CCM) is shown in Fig. 2. It is not included in Fig. 1. because it is a conceptual abstraction that spans all other models. The *model component* is the topmost concept in the meta-model. Each model component has a unique identifier and a text field that allows the component to be labeled with a single name, a sentence, or a long text depending on the modeling needs. A modeling component has a *description*, which is a text that provides additional clarification for the component.

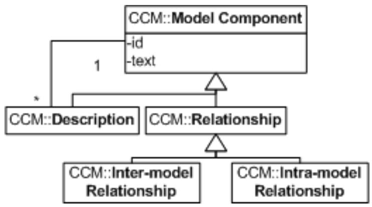


Fig. 2. The Common Components Model

The *relationship* concept is a model component that connects other model components with each other. Two distinct types of relationships are defined in the meta-model: *intra-model relationships* that link components within the same model; and *inter-model relationships* that enable components from different models to be related with each other. The inter-model relationships facilitate traceability among the models and provide mechanisms to design intersecting models.

4.2 Concepts Model

Concepts that are necessary to describe the static aspects of enterprises and information systems are modeled in the CM (Fig 3). They include resources and information objects that are used, processed, exchanged, produced, and stored in the organization, together with their relationships and attributes.

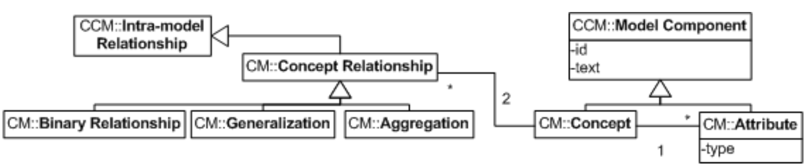


Fig. 3. The Concepts Model

A *concept* represents entities about which the enterprise stores or processes information. Concepts represent resources, information objects, or other things that are of interest to the enterprise. They are described by *attributes* that declare properties for the concepts. Concepts can be related to each other using a *concept relationship*, which can be one of three kinds: the *binary relationship*, which is a general kind of relationship; the *generalization* relationship, which relates a general concept to a more specific one; and the *aggregation* relationship, used to indicate that a concept is composed of other concepts.

4.3 Goal Model

Organizational *business goals* are recorded and represented in the GM (Fig. 4). A business goal is a future state-of-affairs that the enterprise aims to attain, and through which it can grow and generate profit. An enterprise can identify potential desirable situations as *opportunities*, which highlight new possibilities or capabilities that can be transformed into actual business goals. Both opportunities and business goals are defined as types of *intentional components* because they share many properties. Moreover, modeling opportunities as intentional components allows the identification of concepts, roles, processes, and requirements; otherwise associated only with business goals. Intentional components can *support* each other, indicating that achieving one contributes to achieving the other. Intentional components can also *conflict* with each other when the realization of one challenges the realization of the other. The role which *defines* an intentional component is captured in the meta-model to provide traceability to the source of the component.

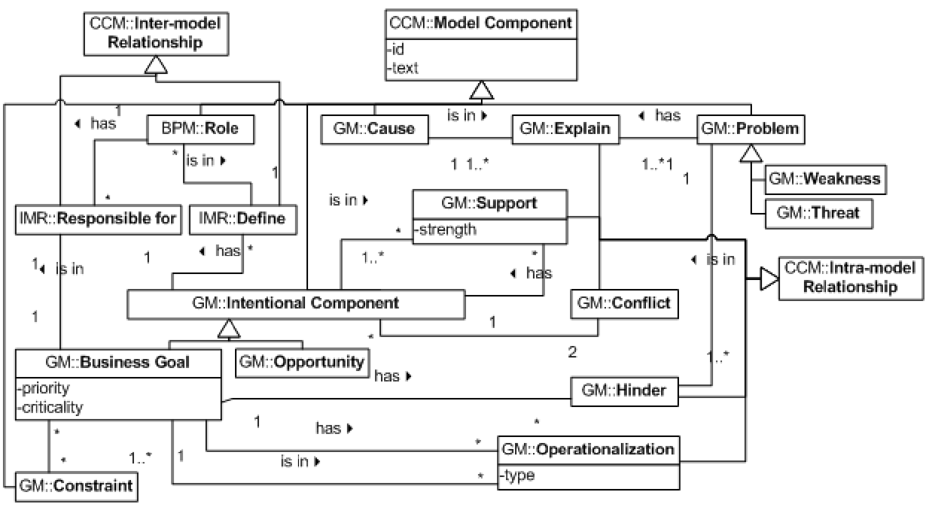


Fig. 4. The Goal Model

The *operationalization* relationship provides additional structure to the GM by allowing business goals to be decomposed into smaller, more concrete sub-goals. Decomposition can occur in one of three types, or modes: AND operationalization, indicating that the fulfillment of all sub-goals is necessary to fulfill the goal; OR operationalization, when the fulfillment of at least one sub-goal is enough to fulfill the goal; and XOR operationalization, when sub-goals are exclusive alternatives for the goal. Operationalization enables organizing the intentional components as a hierarchy.

Goals have roles that are *responsible for* them; tracking the progress of their fulfillment and making sure necessary resources are allocated for that purpose.

Achieving the business goals is usually hindered by various obstacles, and it pays to include those obstacles in the model to provide a clearer view of the organizational landscape. *Problems* that *hinder* business goals can be either internal to the enterprise, in which case they are considered *weaknesses*, or external, in which case they are modeled as *threats*. The *cause* that *explains* a problem is a useful insight when identified explicitly, and can contribute to finding suitable measures and solutions. In addition, a business goal is bound by *constraints*, which represent rules and regulations that affect how the organization operates. Constraints are always external to the organization; internal rules and regulations are described using the BRM.

4.4 Business Process Model

Business goals identified in the GM give rise to, or *motivate*, the design of business processes that describe activities in the enterprise needed to realize the goals. The BPM (Fig. 5) provides a view over the processes and their composition and structure. A *process* model component stands for different sizes of processes at both the business- and IS-levels, thus providing a unified dynamic view of the enterprise and its IS. The relationship between a process and its sub-processes is captured as a *composition* relationship, indicating that the sub-processes work together to accomplish the top process. The meta-model includes no limit to the number of decomposition levels, which is left to the specific needs of projects.

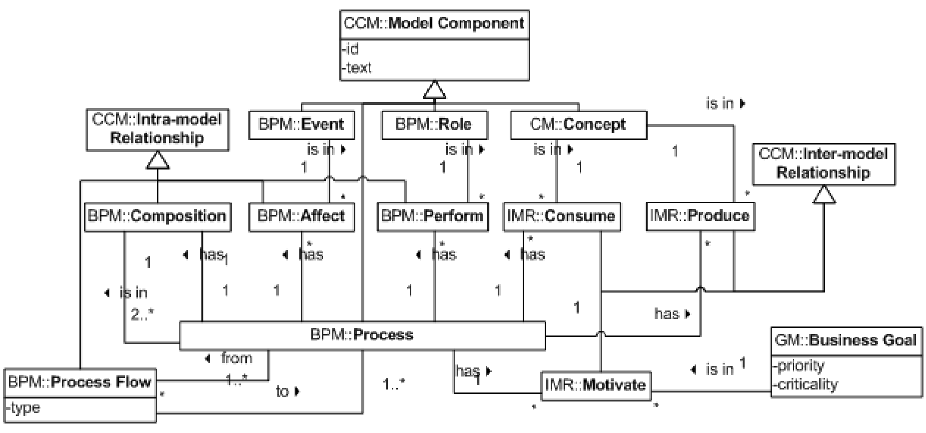


Fig. 5. The Process Model

The flow between processes is described using the *process flow* relationship, which connects processes in one execution flow. The type of the relationship indicates whether the processes are performed in parallel (AND connection), optional (OR connection), or conditional (XOR connection). Processes are affected by *events*, which are external occurrences that influence the execution of the process and cause it to deviate into certain paths, e.g. at the decision points. Concepts that are *consumed* and *produced* are included in the BPM using inter-model relationships, denoting the inputs that guide the process execution and outputs that result from the execution.

Actors that *perform* processes in the enterprise are modeled as *roles,* which can also *provide* and be *responsible for* goals, and can be related to requirements. A role stands for a general position that is independent of the actual persons filling it, and it can represent physical persons, virtual persons, or automated systems.

4.5 Requirements Model

High-level requirements, also called business requirements, express the stakeholders' desires for a future IS. Business requirements are refined into more concrete IS requirements that are better understood by system designers. The line separating business requirements and system requirements is vague and hard to identify, but the decomposition is always necessary. Therefore, the meta-model for the RM (Fig. 6) includes a single *requirement* component that serves as a high-level as well as a concrete system requirement. *Decomposition* of a requirement occurs on any level, and continues depending on the judgment of the modeler and the specific needs of the project. The decomposition relationship is used to connect a parent requirement with its child requirements, which can be all necessary (AND decomposition), alternatives (OR decomposition), or exclusive alternatives (XOR decomposition). Requirements that negatively affect the realization of each other are connected using the *conflict* relationship. Also, requirements that positively affect the realization of each other are connected using the *dependency* relationship.

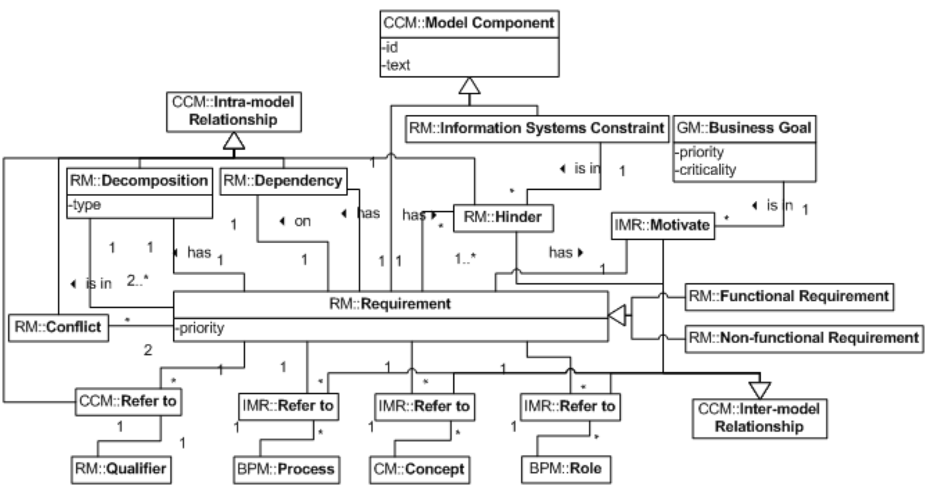


Fig. 6. The Requirements Model

Requirements at different levels of decomposition are *motivated* by business goals. However, some requirements address system related issues that are not relevant to the high-level organizational goals. When those requirements are elicited, *information systems constraints* that motivate or *hinder* them must be identified and assessed.

Requirements in our meta-model provide a central connection point between the models. In the modeling process, actors, things, and activities that are involved in a requirement are identified and the corresponding roles, concepts, and processes are linked with the requirement, respectively. Requirements have qualifiers that express additional information about certain qualities a requirement can have, e.g. performing the functionality at certain times or periodically, or the location for storing data.

4.6 Business Rules Model

Internal rules and regulations that govern the enterprise provide boundaries for the concepts and business processes. Business rules are often formulated together with the business goals to specify how the goals will be achieved. Concepts and business processes that are motivated by the business goals are governed by the defined business rule, and this is captured in the BRM (Fig. 7).

Business rules are represented in the meta-model as model components that are *motivated* by business goals. In other words, the business rules affect the fulfillment of business goals. A business rule *refers to* one or more concepts that are constrained by it. When a rule defines a necessity that needs to be guaranteed at all times by the involved concepts, it is called a *structural business rule*. When a rule addresses derived or dynamic properties that must be checked at certain points in time or when certain events occur, it is called a *behavioral business rule*. This type of business rules constraints the enterprise in terms of the change of its state, and can lead to different results depending on the triggering events.

Breaking a structural business rule produces an invalid state for the enterprise, and hence must be prevented. However, it is possible to break behavioral rules, and such breaches entail corrective action that returns the enterprise to a valid state. While structural rules *constrain* concepts, behavioral rules *constrain* processes, and can *motivate* the design of additional processes that are needed to enforce the rules.

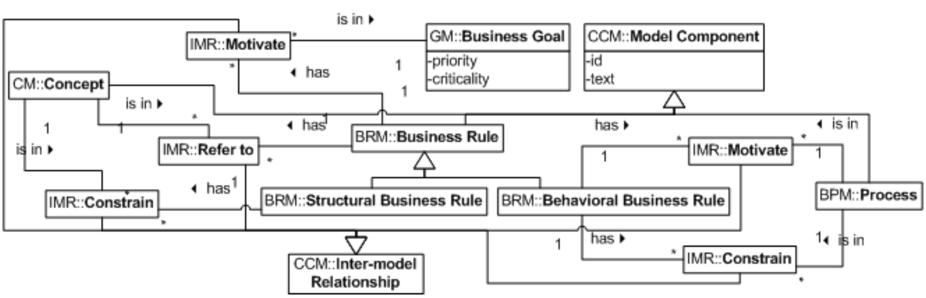


Fig. 7. The Business Rules Model

4.7 IT Architecture Model

Creating a complete IS model involves describing the design architecture that the final system will be based on, and how its different parts will operate conjointly. It also involves describing how eventual users will interact with the system, and other technical details related to the MDD platform used for implementation. This information is captured by the ITAM, which is technology dependent and can exist in various forms for the same GM, CM, BPM, and RM. The ITAM must support inter-model relationships between components of the mentioned models and specific architectural components, depending on the selected implementation architecture. Those relationships highlight the motivation behind architectural design decisions, and provide additional information that can be exploited when transforming the models into an executable system.

5 An Example Case Study

To demonstrate the use of the proposed meta-model in an MDD setting, we use an example case inspired by the project Energy Efficiency and Risk Management in Public Buildings (EnRiMa)[1]. The overall goal of the project is to develop a decision support system assisting building managers in making smart strategic and operational decisions considering a multitude of factors, e.g. occupant needs, market prices, installed technology, environmental factors, and weather.

The aim of this example is to demonstrate the use of the proposed meta-model starting from high-level enterprise models. First, some business goals are elicited in the context of the EnRiMa project. Other models are then developed from the business goals according to the meta-model. Parts of the models are identified as candidates for IS support and hence realized as a Web services implementation.

The business goals used in this example are: 1) satisfy comfort requirements of occupants, 2) reduce CO_2 emissions, 3) increase energy efficiency of the building, 4) enable daily energy adjustments, and 5) balance long- and short-term plans. Fig. 8 shows the GM that describes the goals and their relationships.

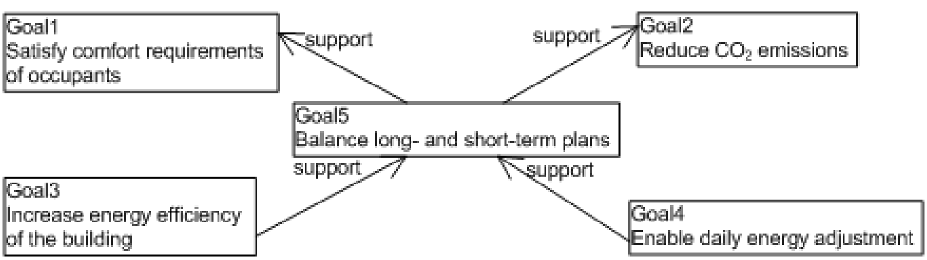


Fig. 8. Example Goal Model

Goal 4 motivates a process whereby hourly readings of wind speed, temperature, and humidity are collected and compared with historical data to predict a short-term weather forecast. The forecast is then projected on current and prospective energy prices in the market, a step motivated by Goal 3. This process is described using a BPM (Fig. 9) that includes top-level processes and shows the different inputs and outputs involved. Modeling of processes can continue in a real situation by breaking

[1] http://www.enrima-project.eu/

down each process into its constituent parts, until a suitable level of granularity is reached. Each part is itself a process with its own inputs and outputs. Inter-model relationships allow the inclusion of business goals in the process model, providing traceability support between processes and their originating goals.

By identifying processes which merit automation, the requirements describing the information systems that support those processes can be elicited. For example, Process 4 can be implemented as Web services that gather the necessary information and calculate the adjustments to the heating system parameters. The requirements that describe the system for executing Process 4 can be formulated as: 1) the system shall retrieve current energy prices; 2) the system shall retrieve prospective energy prices; and 3) the system shall deploy a predefined set of formulae to calculate the adjustments. These requirements can be decomposed into more fine-grained ones, and, as with processes in the BPM, the decomposition can continue until a desirable level of granularity has been reached. The resulting RM is shown in Fig. 10. The business goal motivating the requirements and the process which is related to the requirements are present in the RM using inter-model relationships.

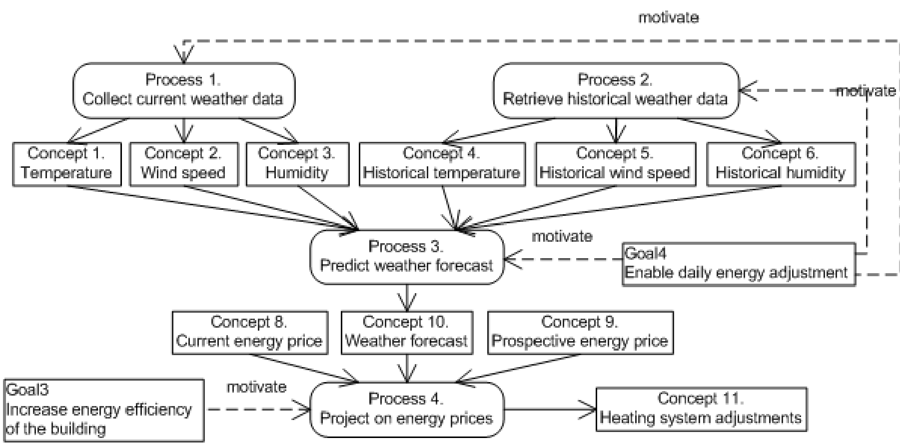


Fig. 9. Example Business Process Model

Concepts defined in the BPM and RM are part of a larger CM that describes all relevant concepts needed for building management together with their relationships.

The information encoded in the models discussed so far is sufficient to enable an automatic generation of a service-based system that supports the identified processes and business goals. Each functional requirement in the RM in Fig. 10 is a candidate for becoming a Web service. In this example, manual intervention is needed to select suitable requirements for becoming Web services. However, mapping requirements and other modeling components will be facilitated using an ITAM which defines the relationships between eventual components of the final system. Inter-model relationships between components in the ITAM and components of other models will help formalize the transformation of different modeling components into databases, Web services, user interfaces, or any other type of IS component.

Functional Requirement 3 (see Fig. 10) is a function suitable for transformation into a Web service. Depending on the chosen granularity for modeling the processes and the requirements, the models can be used to generate simple Web services that include only one operation or more complex services that combine multiple operations. The approach provides flexibility for making such decisions, and eventual supporting tools must enable modelers to choose a suitable mapping to simple operations or complex services (using the ITAM). Moreover, the inputs and outputs of the Web service can be derived from the inter-model relationships between requirements and concepts. While these relationships show only the concepts that are related to requirement, referring to the processes that are related to the same concepts can help in identifying whether a concept is an input or an output for the Web service. An excerpt of the generated WSDL code for the Web service definition is shown in the code snippet below. The XML schema definition referred to in "concepts.xsd" corresponds to the concepts defined in the CM.

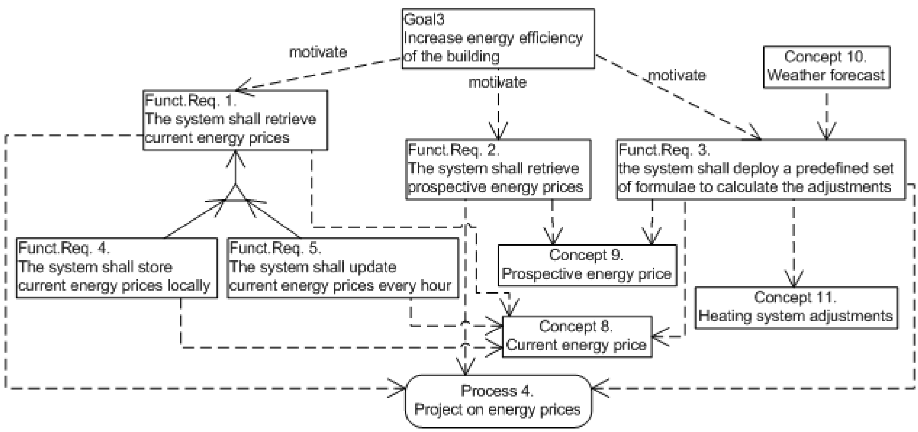


Fig. 10. Example Requirements Model

Code snippet of the generated WSDL code defining the Web service that corresponds to Functional Requirement 3 (see Fig. 10).

```
<definitions name="CalculateHeatingSystemAdjustments"
targetNamespace="http://example.com/calcadjust.wsdl"
xmlns:xsd1=”http://example.com/concepts.xsd”
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  ...
  <wsdl:message name="GetAdjustmentsInput">
    <wsdl:part name="WeatherForecast"
element="xsd1:WeatherForecast"/>
    <wsdl:part name="CurrentEnergyPrice"
element="xsd1:CurrentEnergyPrice"/>
    <wsdl:part name="ProspectiveEnergyPrice"
element="xsd1:ProspectiveEnergyPrice"/>
  </wsdl:message>
```

```
  <wsdl:message name="GetAdjustmentsOutput">
    <wsdl:part name="HeatingSystemAdjustments"
element="xsd1:HeatingSystemAdjustments"/>
  </wsdl:message>

  <wsdl:portType name="AdjustmentsPortType">
    <wsdl:operation name="GetAdjustments">
      <wsdl:input message="GetAdjustmentsInput"/>
      <wsdl:output message="GetAdjustmentsInput"/>
    </wsdl:operation>
  </wsdl:portType>
  ...
  <wsdl:service name="HeatingSystemAdjustmentsService">
    <wsdl:port name="AdjustmentsPort"
binding="someBinding">
      <soap:address
location="http://example.com/stockquote"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

6 Related Work

Lin and Sølvberg [9] present a framework for motivating process models using goal annotations. Ontologies were used to annotate process and goal models on the meta-level, establishing connections between processes and business goals that motivate them. Shahzad et al. [18] propose a generic meta-model for business processes that combines functional, behavioral, organizational, and informational perspectives. The meta-model is used, together with a generic formal process description, to create a process model repository that enables process model reuse.

Attempts to provide an integrated view of IS requirements using meta-models can be traced back to Jordan and Davis [7]. Recent meta-models for requirements management in software product families are found in Arpinen et al. [1] and Cerón et al. [3]. The meta-models include relationships between requirements and other organizational entities, such as actors and contracts. Coarse-grained relationships to other parts of the IS development lifecycle, such as system (design) models and test cases, are also present. The meta-model proposed by Goknil et al. [6] unifies common concepts in existing requirements modeling approaches. Requirements are described using their properties, types, and relationships with other requirement-related concepts, such as test cases and stakeholders. López et al. [11] treat design models as instances of semi-formal requirements models. Use case diagrams, activity diagrams, and workflow diagrams are unified on the meta- level using common modeling units.

While the approaches above include unifying meta-models for IS development, they focus on a single aspect of the final system (processes in [9] and [18], and requirements in the others). In contrast, our approach covers a wider range of organizational and IS concerns.

Pastor and Giachetti [15] present a generic process for linking i* [21] and the OO-Method—as representatives of Goal-Oriented Requirements Engineering (GORE) and MDD, respectively. By basing the linked meta-models on EMOF [13], the authors are able to define an integration meta-model and generate model transformation rules that automatically produce MDD-compliant conceptual models from i* models. However, additional transformations are necessary to produce the final system. Our approach advocates a single-step transformation that eliminates the need for intermediate steps and helps in generating the final system directly from the models.

7 Conclusion

EM aims at creating a structured and unified view of an enterprise, enabling more informed and accurate decisions to be made. MDD is an approach to IS development that focuses on models as drivers of the development process and part of the final product. In this paper, we have proposed a meta-model that spans EM and MDD to give an integrated view of organizational and IS concerns. EKD was chosen as the EM approach to be integrated with MDD because it includes inter-related sub-models that cover different aspects of the enterprise. The meta-model was designed to support the integration properties suggested in [22] for bridging the gap between requirements and MDD design models. Namely, the following properties are supported:

- Static and dynamic aspects of enterprises and information systems are captured using the CM and the BPM;
- Intentional aspects are addressed in the GM, which offers a high-level view of the enterprise aims and lays the context for other models;
- Architectural aspects are captured using the ITAM, which relates components of the different models to specific implementation platform components;
- Change propagation and traceability are supported by inter-model relationships, which enable the components of one model to be included in other models, offering an integrated view of all models.

By integrating enterprise models and IS design models in a single meta-model, the need to transform models between multiple levels of abstraction—a common practice in MDD approaches—is eliminated, leaving only a single transformation step towards the implementation platform. This single-step transformation is further facilitated by the ITAM, which defines the targets that model components will be transformed into.

The meta-model was demonstrated by an example case. However, a larger and more detailed case study is necessary to show the full potential of the meta-model and to uncover the weaknesses in need for improvements. As future work, we will develop an ITAM for a specific implementation architecture, as well as a supporting tool for realizing the case study and for demonstrating the automation that stems from following the MDD principles.

References

1. Arpinen, T., Hämäläinen, T.D., Hännikäinen, M.: Meta-Model and UML Profile for RequirementsManagement of Software and Embedded Systems. EURASIP Journal on Embedded Systems, Article ID 592168, 14 pages (2011)
2. Bubenko jr., J.A., Persson, A., Stirna, J.: D3: User guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns. Royal Institute of Technology (KTH) and Stockholm University, Stockholm, Sweden (2001)

3. Cerón, R., Dueñas, J.C., Serrano, E., Capilla, R.: A Meta-model for Requirements Engineering in System Family Context for Software Process Improvement Using CMMI. In: Bomarius, F., Komi-Sirviö, S. (eds.) PROFES 2005. LNCS, vol. 3547, pp. 291–313. Springer, Heidelberg (2005)
4. France, R., Rumpe, B.: Model-Driven Development Of Complex Software: A Research Roadmap. In: Future of Software Engineering, FOSE 2007, pp. 37–54. IEEE Computer Society, Washington, DC, USA. (2007); Springer, Heidelberg (2002)
5. Gašević, D., Djurić, D., Devedžić, V.: Model Driven Engineering. In: Model Driven Engineering and Ontology Development, pp. 125–155. Springer, Heidelberg (2009)
6. Goknil, A., Kurtev, I., van den Berg, K.: A Metamodeling Approach for Reasoning about Requirements. In: Schieferdecker, I., Hartman, A. (eds.) ECMDA-FA 2008. LNCS, vol. 5095, pp. 310–325. Springer, Heidelberg (2008)
7. Jordan, K.A., Davis, A.M.: Requirements Engineering Metamodel: An Integrated View of Requirements. In: Computer Software and Applications Conference, pp. 472–478. IEEE, Los Alamitos (1991)
8. Caskurlu, B.: Model Driven Engineering. In: Butler, M., Petre, L., Sere, K. (eds.) IFM 2002. LNCS, vol. 2335, pp. 286–298. Springer, Heidelberg (2002)
9. Lin, Y., Sølvberg, A.: Goal Annotation of Process Models for Semantic Enrichment of Process Knowledge. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 355–369. Springer, Heidelberg (2007)
10. Loniewski, G., Insfran, E., Abrahão, S.: A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010 Part II. LNCS, vol. 6395, pp. 213–227. Springer, Heidelberg (2010)
11. López, O., Laguna, M.A., García, F.J.: Metamodeling for Requirements Reuse. In: Workshop em Engenharia de Requisitos, WER 2002, pp. 76–90. Valencia, Spain (2002)
12. Marshall, C.: Enterprise Modeling with UML: Designing Successful Software Through Business Analysis. Addison–Wesley, Essex (2000)
13. Object Management Group (OMG), Meta Object Facility (MOF) Core Specification Version 2.0 (2006)
14. Object Management Group (OMG), Unified Modeling Language (UML) 2.0 (2005)
15. Pastor, O., Giachetti, G.: Linking Goal-Oriented Requirements and Model-Driven Development. In: Intentional Perspectives on Information Systems Engineering, pp. 257–276. Springer, Heidelberg (2010)
16. Schmidt, D.C.: Model-Driven Engineering. Computer 39, 25–31 (2006)
17. Selic, B.: The Pragmatics of Model-Driven Development. IEEE Software 20, 19–25 (2003)
18. Shahzad, K., Elias, M., Johannesson, P.: Towards Cross Language Process Model Reuse – A Language Independent Representation of Process Models. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 176–190. Springer, Heidelberg (2009)
19. Stirna, J., Kirikova, M.: Integrating Agile Modeling with Participative Enterprise Modeling. In: EMMSAD 2008, pp. 171–184. CEUR (2008)
20. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: 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)
21. Yu, E.S.K.: Modelling Strategic Relationships for Process Reengineering, PhD Thesis, University of Toronto, Toronto, Canada (1995)
22. Zikra, I., Stirna, J., Zdravkovic, J.: Analyzing the Integration between Requirements and Models in Model Driven Development. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 342–356. Springer, Heidelberg (2011)

Author Index