Janis Grabis
Marite Kirikova
Jelena Zdravkovic
Janis Stirna (eds.)

PoEM Short Papers 2013

Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013), Riga, Latvia, November 6-7, 2013

CEUR-WS.org/Vol-1023

ISSN 1613-0073

Preface

PoEM 2013 – the 6th IFIP WG 8.1 Conference on the Practice of Enterprise Modeling was held in November 2013 in Riga, Latvia. Since its foundation in 2008, the PoEM conference addresses the use of enterprise modeling in practice by bringing together the academic community and practitioners from industry to contribute to improved and novel enterprise modeling practices.

Enterprise Modeling (EM) is discipline that aims to externalize enterprise knowledge and thus adding to its value. It comprises modelling of the structure, behavior and organization of the enterprise. Using these models, the practitioners aim to better understand the enterprise's functions, as well as to better link the functions to the models of Information Systems.

The PoEM 2013 conference included 39 presentations. The presented practitioner and short papers (on research in progress in enterprise modeling) are included in this volume. Other conference papers are available in PoEM 2013 Proceedings, published by Springer.

These Short Paper Proceedings include 20 practitioner or short papers. The authors of the papers represent ten countries: Colombia, France, Germany, Netherlands, Latvia, Luxembourg, Norway, Poland, Sweden, and Switzerland. Reflecting different topics of EM research, practices and tools, the papers were presented in five sessions: Enterprise Architecture; EM and Business Processes; Conceptualizations, Notations and Ontology; EM and Information Systems; and Compliance in EM.

The organization of PoEM 2013 owes special thanks to the members of the International Program Committee for promoting the conference, as well as for providing valuable reviews for the submitted papers. Furthermore, we are grateful to all authors who submitted the papers to the conference, and all participants who contributed to the Short Paper sessions. Special thanks go to Riga Technical University for an engaging organization of the conference.

November 2013

Janis Grabis
Marite Kirikova
Jelena Zdravkovic
Janis Stirna

PoEM 2013 Organization

Steering Committee

Anne Persson, Skóvde University, Sweden
Janis Stirna, Stockholm University, Sweden

General Chair

Jānis Grabis, Riga Technical University, Latvia

Organizers

Jānis Grabis, Riga Technical University, Latvia
Gundega Lazdane, IIBA Latvia Chapter, Latvia
Lilita Sparane, Latvian IT Cluster, Latvia
Renate Strazdina, Riga Technical University (industrial cooperation chair)
Jānis Kampars, Riga Technical University, Latvia

Program Committee Co-chairs

Marite Kirikova, Riga Technical University, Latvia
Jelena Zdravkovic, Stockholm University, Sweden

Program Committee

Daniel Amyot, University of Ottawa, Canada
Marko Bajec, University of Ljubljana, Slovenia
Janis Barzdins, University of Latvia, Latvia
Guiseppe Berio, University of South Brittany, France
Robert Buchmann, University of Vienna, Austria
Rimantas Butleris, Kaunas University of Technology, Lithuania
Albertas Caplinskas, VU IMI, Lithuania
Steinar Carlsen, Computas, Norway
Wolfgang Deiters, Fraunhofer ISST, Germany
Sergio España, Universitat Politècnica de València, Spain
Xavier Franch, Universitat Politècnica de Catalunya, Spain
Janis Grabis, Riga Technical University, Latvia
Norbert Gronau, University of Potsdam, Germany
Remigijus Gustas, Karlstad University, Sweden
Patrick Heymans, University of Namur, Belgium
Stijn Hoppenbrouwers, Radboud University Nijmegen, the Netherlands
Jennifer Horkoff, University of Trento, Italy
Paul Johannesson, Stockholm University, Sweden
Håvard Jørgensen, Commitment AS, Norway
Marite Kirikova, Riga Technical University, Latvia

John Krogstie, Norwegian University of Science and Technology, Norway
Marc Lankhorst, Novay, The Netherlands
Birger Lantow, University of Rostock, Germany
Ulrike Lechner, Munich University of the Armed Forces, Germany
Michel Leonard, Université de Genève, Switzerland
Peri Loucopoulos, Harokopio University of Athens, Greece /University of Loughborough, UK
Florian Matthes, Munich University of Technology, Germany
Raimundas Matulevicius, University of Tartu, Estonia
Graham McLeod, inspired.org, South Africa
Jan Mendling, Vienna University of Economics, Austria
Christer Nellborn, Nellborn Management Consulting AB, Sweden
Björn Nilsson, Anatés AB, Luxembourg
Andreas Opdahl, University of Bergen, Norway
Sietse Overbeek, University of Duisburg-Essen, Germany
Oscar Pastor, Valencia University of Technology, Spain
Anne Persson, University of Skovde, Sweden
Michael Petit, University of Namur, Belgium
Tomas Pitner, Masaryk University, Czech Republic
Naveen Prakash, GCET, India
Erik Proper, Radboud University Nijmegen, the Netherlands
Jolita Ralyté, Université de Genève, Switzerland
Colette Rolland, Université Paris 1 Panthéon Sorbonne, France
Irina Rychkova, Université Paris 1 Panthéon Sorbonne, France
Kurt Sandkuhl, Jönköping Technical University, Sweden
Ulf Seigerroth, Jönköping International Business School, Sweden
Khurram Shahzad, University of the Punjab, Pakistan
Nikolay Shilov, SPIIRAS, Russia
Pnina Soffer, University of Haifa, Israel
Maarten Steen, Novay, the Netherlands
Janis Stirna, University of Stockholm, Sweden
Darijus Strasunskas, DS Applied Science, Norway
Renate Strazdina, Ernst&Young SIA, Latvia
Olegas Vasilecas, Vilnius Gediminas Technical University, Lithuania
Mathias Weske, University of Potsdam, Germany
Eric Yu, University of Toronto, Canada
Jelena Zdravkovic, Stockholm University, Sweden

Additional Reviewers

Gundars Alksnis, Riga Technical University, Latvia
Solviata Berzisa, Riga Technical University, Latvia
Nelly Condori-Fernandez, University of Twente, The Netherlands
Sybren De Kinderen, Vrije Universiteit Amsterdam, The Netherlands
Arturo Gonzalez, Universitat Politècnica de València, Spain
Matheus Hauder, Munich University of Technology, Germany
Martin Henkel, Stockholm University, Sweden
Amin Jalali, Stockholm Univeristy, Sweden (PhD student)

Oleh Khovalko, University of Potsdam, Germany
Ivan Monahov, Munich University of Technology, Germany
Laila Niedrite, University of Latvia, Latvia
Georgious Plataniotis, CRP Henri Tudor, Luxembourg
Inese Polaka, Riga Technical University, Latvia
Diego Prado-Gesto, Valencia University of Technology, Spain
Luise Pufahl, Munich University of Technology, Germany
Marcela Ruiz, Universitat Politècnica de València, Spain
Agris Sostaks, University of Latvia, Latvia
Inese Supulniece, Riga Technical University, Latvia
Vladimir Tarasov, Jönköping International Business School, Sweden
Justas Trinkunas, Vilnius Gediminas Technical University, Lithuania
André Ullrich, University of Potsdam, Germany
Dirk van der Linden, CRP Henri Tudor, Luxembourg
Marin Zec, Munich University of Technology, Germany

Table of Contents

Embracing Imperfection in Enterprise Architecture Models

Hector Florez, Mario Sánchez, and Jorge Villalobos

Department of Systems and Computing Engineering,
Universidad de los Andes,
Bogotá, Colombia
{ha.florez39,mar-san1,jvillalo}@uniandes.edu.co

Abstract. In Enterprise Architecture (EA) projects, models are built to represent business and Information Technologies (IT) elements, and to abstract the relation between them in one enterprise under study. The construction of EA models depends on information provided by different kinds of sources, but sources could be insufficient or information could be incomplete or incorrect regarding aspects of the enterprise. As a result, modelers must often create models based on low quality information that could not properly represent the enterprise. Since EA models are used as the base for the analysis of the enterprise, using models that do not represent the enterprise correctly, there is a risk that the analysis produces unreliable conclusions. This paper presents a proposal for managing imperfect models in the EA context. An imperfect model is capable of representing information that can be imprecise, inconsistent, vague, uncertain, or incomplete; thus, when analyses are performed, they can use this information to produce more reliable results. In this proposal, imperfect models also include information about modeler decisions.

Keywords: Enterprise Architecture, Model Driven Engineering, Models Imperfection

1 Introduction

Enterprises increasingly depend on Information Technologies (IT) and require support in order to achieve their business goals. Moreover, the enterprise complexity and constant evolution generate difficulties in the relation between the business and IT. EA projects require the construction of one model that we call *enterprise architecture model (m_{EA})*, which is an abstract representation of one simplification [1, 2] of the enterprise that includes enterprise elements and their relations. The m_{EA} is usually big because enterprises have a large number of elements and is complex because they have a large number of typed relations between their elements. One m_{EA} is built by *modelers* through direct and indirect observation. Direct observation is the action in which modelers obtain enterprise information without consulting sources. Indirect observation requires the participation of sources (e.g., persons, documents, and meetings). In an EA project, the m_{EA} is used to analyze the enterprise. These analyses are performed by domain experts called *analysts*, who manipulate the m_{EA} in order to extract useful information for evaluating the current state of the enterprise.

The model construction process is an observation of human process, sources consulting, interpreting, and expression [3], so the construction of one m_{EA} that properly represents the enterprise has a high level of difficulty. The difficulty is based on two main reasons: (1) the enterprise size and complexity and (2) several uncontrolled factors that affect the modeling process [4] such as sources quality and lack of information. Quality refers to the knowledge level of the sources on specific aspects of the enterprise. Lack of information refers to the incapability of the source for providing information on some aspects of the enterprise. Because of this, in some cases it is impossible to build an m_{EA} that correctly represents the enterprise. For instance, consider the case where a model will be used to document the business applications of an enterprise, and analyze their availability. Then, one employee may assert that the availability (that must be a number) of the *BusinessApp_X* is between 92% and 95%, and the availavility of the *BusinessApp_Y* is *High*. The modeler must choose one numeric value for the *BusinessApp_X* between 92% and 95%, and a numeric value for the *BusinessApp_Y* that must be high (up to 100 in this case). In any cases, the values selected by the modeler do not represent the enterprise correctly because he/she cannot know the appropiate value. Later on, the analyst will use the model to perform an analysis of the enterprise; however, since he/she will be using a m_{EA} that might not represent some enterprise elements or relations correctly; in such case, the results should not be considered very reliable.

Model imperfection is inevitable in EA projects. Then, it is better to include information in the model to represent those problems than to ignore them and to assume that the model accurately represents the enterprise. Modeling the imperfection implies creating another model that we call *imperfect enterprise architecture model (m_ζ)*, which is an approximation of the m_{EA} with information that can assess the imperfection. Thus, there is a distance between m_ζ and m_{EA} and the modeler must build the best possible approximation given the quality of the available information. To establish the level of the information validity, the imperfect information should be related with sources that provide the information. In the construction process of the m_ζ, modelers consult sources through observations (e.g., interviews, reviews, meeting acts, etc.) where each observation provides facts. Through these facts, modelers make decisions to assign values for imperfect attributes and relations. Finally, based on the m_ζ, it is possible to create analysis methods while taking into account the imperfection.

The rest of the paper is structured as follows. Section 2 describes the modeling process in the EA context. Section 3 presents our proposed taxonomy of imperfect sources, where we classify sources based on the quality of the information provided. In section 4, we present our proposal for modeling the imperfection in EA projects based on the sources classification. In section 5, we present our tool for modeling imperfection and creating the m_ζ. In section 6 we present modeling imperfection involving traces about decision making by modelers, for imperfect attributes and relations. Finally, section 7 presents the conclusions.

2 Construction of Enterprise Architecture Models

In EA projects, it is necessary to construct one domain metamodel that we call *enterprise architecture metamodel (MM_{EA})* representing the concepts of the enterprise and the relations between them. Those concepts are typed elements which contain structural features. The MM_{EA} is different for each specific project, since it reflects the horizontal and vertical scope for the project. In this paper, we assume that MM_{EA} is correct and never changes during the project.

Normally, the m_{EA} must conform to the MM_{EA}. The m_{EA} is made up of several instances ($\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_p\}$) that conform to the correspondent typed elements of MM_{EA}. Each instance in Γ contains several structural features ($\Phi_{\gamma_i} = \{\varphi_{\gamma_i,1}, \varphi_{\gamma_i,2}, \ldots, \varphi_{\gamma_i,q}\}$), which can be attributes ($A_{\gamma_i} = \{\alpha_{\gamma_i,1}, \alpha_{\gamma_i,2}, \ldots, \alpha_{\gamma_i,r}\}$) or typed relations ($B_{\gamma_i} = \{\beta_{\gamma_i,1}, \beta_{\gamma_i,2}, \ldots, \beta_{\gamma_i,s}\}$). The conformance relation between m_{EA} and MM_{EA} establishes that Φ_{γ_i} must respect the types and structures defined for each metatype in MM_{EA}. However, if modelers build one m_ζ instead of one m_{EA}, this m_ζ does not conform to MM_{EA}.

The m_ζ can include imperfect information, where the modeler can decide which attributes or relations are imperfect. To build the m_ζ, we use the distinction between *ontological conformance* that is based on the relation between the model and metamodel in terms of their meaning, and *linguistic conformance* that is based on the relation between the model and metamodel in terms of their structure [5]. In addition, we achieve the linguistic conformance by the construction of a generic intermediate metamodel that serves to represent any type, attribute and relation; and the ontological conformance by the definition of semantic rules [6]. Then, the m_ζ allows the creation of instances of a type with regular or imperfect structural features. One structural feature is imperfect regarding the MM_{EA}, when its value does not adjust with the characteristics established in the MM_{EA}.

Figure 1 illustrates the proposed strategy. The m_ζ *conforms linguistically* to one generic metamodel called *Extended Metamodel of Imperfection (EiMM)* that serves to represent any type, attribute and relation, where attributes and relations can be imperfect. The m_ζ does not conform ontologically to the MM_{EA} because attributes and relations of instances of the same metatype can have different characteristics. We say that the m_ζ *semi-conforms ontologically* to the MM_{EA}. We introduce the *ontological semi-conformance* concept as the relation between the m_ζ and the MM_{EA} in which the m_ζ can include instances of the metatypes in the MM_{EA}, but the instances' structural features can be imperfect and may change some of their features. Given this, we can say that the m_ζ is an approximation of the m_{EA}. Furthermore, m_ζ also includes decision trace information about the decisions that led to the construction of the imperfect model. Decision trace information is related with sources that provide information about aspects of the enterprise, observations that are the activities in which the modeler can collect enterprise information, and decision details. This decision trace information is structured through the *EiMM* that has types to represent all elements involved in the decision making process. As a result, the m_ζ conforms linguistically to *EiMM* and *semi-conforms ontologically* to the MM_{EA}.

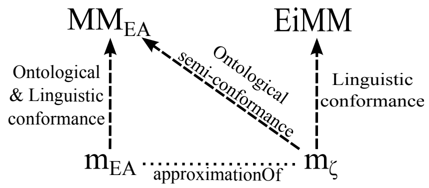


Fig. 1. Strategy for Modeling imperfection.

3 Sources of Enterprise Information

In the construction process of one model, modelers must consult enterprise sources *($\Psi = \{\psi_1, \psi_2, \ldots, \psi_r\}$)*. One source ψ_i provides information to modelers through *Observations ($\Theta_{\psi_i} = \{\theta_{\psi_i,1}, \theta_{\psi_i,2}, \ldots, \theta_{\psi_i,s}\}$)*. One observation $\theta_{\psi_i,j}$ produces some knowledge about enterprise elements that we call *Facts ($\Omega_{\theta_{\psi_i,j}} = \{\omega_{\theta_{\psi_i,j},1}, \omega_{\theta_{\psi_i,j},2}, \ldots, \omega_{\theta_{\psi_i,j},t}\}$)*. Each fact $\omega_{\theta_{\psi_i,j},k}$ provides information about instances, attributes, or relations, for building the model.

Enterprise sources could be imperfect because they can provide information that does not properly represent some elements of the enterprise. We classify sources based on the properties of the information provided as incorrect [4], imprecise [4, 7, 8, 9], inconsistent [10, 11], vague, uncertain [7, 12, 13] or a combination of some of them. In addition, it is possible that one source, which should provide information, cannot provide information about one structural feature. Then, the corresponding observation will not produce any fact. In this case, we classify the observation as an incomplete observation.

Incorrect source. One source ψ_i is incorrect when one observation $\theta_{\psi_i,j}$ over one structural feature $\varphi_{\gamma_k,l}$ establishes a fact $\omega_{\theta_{\psi_i,j},k}$ that provides a value that is not true or is not usable.

Imprecise source. One correct source ψ_i is imprecise when one observation $\theta_{\psi_i,j}$ over one attribute $\alpha_{\gamma_k,l}$ that requires a specific numeric value, provides a range of numeric values with a minimum numeric value *(v_{min})* and a maximum numeric value *(v_{max})* (e.g., the CTO (ψ_i) in one interview ($\theta_{\psi_i,j}$) asserts that the availability ($\varphi_{\gamma_k,l}$) of the *BusinessApp_X* (γ_k) is between 90% and 95%).

Inconsistent source. There are the following cases in which one source ψ_i or several sources $\{\psi_1, \psi_2, \ldots, \psi_r\}$ can be inconsistent.

Case A. Several sources $\{\psi_1, \psi_2, \ldots, \psi_r\}$ are inconsistent when one observation of each source $\{\theta_{\psi_1,i}, \theta_{\psi_2,j}, \ldots, \theta_{\psi_r,k}\}$ about the same aspect of the enterprise determines that one instance γ_l exists; however, some other observations determine that the instance γ_l does not exist.

Case B. Several sources $\{\psi_1, \psi_2, \ldots, \psi_r\}$ are inconsistent when one observation of each source $\{\theta_{\psi_1,i}, \theta_{\psi_2,j}, \ldots, \theta_{\psi_r,k}\}$ provides different values for one structural feature $\varphi_{\gamma_k,l}$ (e.g., the CIO (ψ_1) in one interview ($\theta_{\psi_1,i}$) asserts that the availability ($\varphi_{\gamma_k,l}$) of the *BusinessApp_X* (γ_k) is 92%, and the CTO (ψ_2) in one interview ($\theta_{\psi_2,j}$) asserts that the availability is 95%).

Case C. One source ψ_i is inconsistent when several observations $\{\theta_{\psi_i,1}, \theta_{\psi_i,2}, \ldots, \theta_{\psi_i,s}\}$ provide different values for one structural feature $\varphi_{\gamma_k,l}$. (e.g., the

CTO (ψ_i) in one interview ($\theta_{\psi_i,1}$) asserts that the availability ($\varphi_{\gamma_k,l}$) of the *BusinessApp_X* (γ_k) is 90%, but in other interview ($\theta_{\psi_i,2}$) asserts that is 92%).

Vague source. Vague sources are those that provide one linguistic value among a set of linguistic values $\Xi_{\varphi_{\gamma_k,l}} = \{\xi_{\varphi_{\gamma_k,l},1}, \xi_{\varphi_{\gamma_k,l},2}, \ldots, \xi_{\varphi_{\gamma_k,l},n}\}$ for one specific attribute $\alpha_{\gamma_k,l}$. One correct source ψ_i is vague when one observation $\theta_{\psi_i,j}$ provides a linguistic value to one attribute $\alpha_{\gamma_k,l}$ that requires one specific numeric value (e.g., the CTO (ψ_i) in one interview ($\theta_{\psi_i,1}$) asserts that the availability ($\varphi_{\gamma_k,l}$) of the *BusinessApp_X* (γ_k) is "High" ($\xi_{\varphi_{\gamma_k,l},r}$)).

Uncertain source. One source ψ_i is uncertain when one observation $\theta_{\psi_i,j}$ over one structural feature $\varphi_{\gamma_k,l}$ that requires a specific value, determines that it can have a value with a certainty degree. The value with a certainty degree must be a combination of a value with a probabilistic value *([v, P(v)])*. (e.g., the CTO (ψ_i) in one interview ($\theta_{\psi_i,j}$) asserts that the *BusinessApp_X* (γ_k) supports ($\varphi_{\gamma_k,l}$) the *BusinessProcess_A* with a certainty degree of 80%).

Incomplete observation. One observation $\theta_{\psi_i,j}$ of one source ψ_i that should provide information is incomplete when the observation's facts do not have any value for one structural feature γ_l that requires a specific value. Consequently, the value of the structural feature is indeterminated $\ominus$.

4 Modeling the Imperfection

Modeling the imperfection implies allowing the construction of approximate models instead of exact models regarding MM_{EA}. Then, the modeler does not build a m_{EA}, but builds a m_ζ. Given the facts obtained from one or several sources $\{\Omega_{\Theta_{\psi_1}}, \Omega_{\Theta_{\psi_2}}, \ldots, \Omega_{\Theta_{\psi_r}}\}$, modelers have to make decisions in order to assign values to the model's structural features. There is a set of possible decisions *(Λ=\{λ_1, λ_2, ...λ_u \})* based on the information provided by the sources. Some examples of the decisions that the modeler can make are the following. λ_1: Select all the values; λ_2: Select a subset of the values; λ_3: Select one value; λ_4: Select one value with a certainty degree; λ_5: Select several values with a certainty degree; λ_6: Select a range of numeric values; λ_7: Select one linguistic value; λ_8: Calculate one value; λ_9: Do not select a value.

Decisions λ_3 or λ_8 can be used to remove the imperfection, but this means losing information and potentially creating an incorrect model with respect to the enterprise (e.g., the CTO (ψ_i) in one interview ($\theta_{\psi_i,j}$) asserts that the availability ($\varphi_{\gamma_k,l}$) of the *BusinessApp_X* (γ_k) is between 90% and 94%, but the modeler decides assigning the value 92%). To be able to model the imperfection, it is necessary to model the imprecision, inconsistency, vagueness, uncertainty, and incompleteness.

Imprecise model. One model is imprecise if at least one attribute of one instance has a range of numeric values instead of a single value. The range is given by a minimum value and a maximum value instead of just one value. The modeler can build an imprecise model due to the sources are *imprecise* (decision λ_6), or *inconsistent cases B or C* (decision λ_6).

Inconsistent model. One model is inconsistent if at least one structural feature of one instance has several values instead of just one value. The modeler can build an inconsistent model due to the sources are *imprecise* (decision λ_2), or *inconsistent cases B or C* (decisions λ_1 or λ_2).

Uncertain model. One model is uncertain if at least one structural feature $\varphi_{\gamma_k,l}$ of one instance γ_k has a value, a set of values, or a range of values with a certainty degree. The modeler can build an uncertain model due to the sources are *imprecise* (decisions λ_4 or λ_5), *inconsistent cases B or C* (decisions λ_4 or λ_5), or *uncertain* (decision λ_4).

Vague model. One model is vague if the modeler decides to asign a linguistic value instead of a numeric value to one attribute of an instance. The modeler can build a vague model due to the sources are *imprecise* (decision λ_7), *inconsistent cases B or C* (decision λ_7), *uncertain* (decision λ_7), or *vague* (decision λ_7).

Incomplete model. One model is incomplete if at least one instance's structural feature does not have any value. It might happen because the modeler decided not to assess any value because he/she does not have enough information.

5 A Tool for Modeling Imperfection

In order to achieve modeling imperfection, this proposal includes a tool to build imperfect models *(m_ς)*. This tool is a graphical editor named *iModeler* based on *GraCoT (Graphical Co-Creation Tool)* presented in Gomez et al. [6]. The editor is based on one metamodel named *iMM (Metamodel of Imperfection)*.

5.1 Metamodel of Imperfection iMM

iMM provides a basic linguistic framework for the definition of imperfect models *(m_ς)*. Figure 2 presents *iMM*. This metamodel has the type called `Model`, which serves as container for all other elements. The type `Element` serves to represent

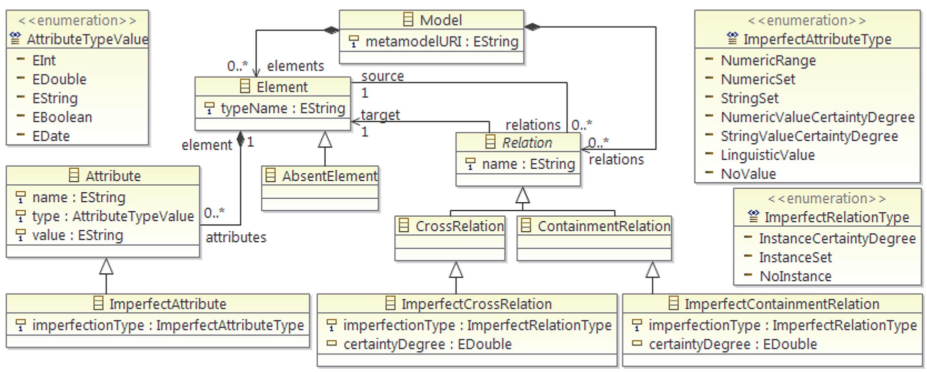


Fig. 2. Metamodel of Imperfection iMM.

the element instances of the imperfect model. Each element in a model has an attribute called `typeName` that serves to relate the element to a type in the MM_{EA}. The types `CrossRelation` and `ContaintmentRelation` serve to represent the relations between elements. The type `Attribute` serves to represent the actual values of attributes contained in elements of the m_ζ. Attribute values may only be integers, doubles, strings, booleans, or dates. The types `ImperfectAttribute`, `ImperfectCrossRelation`, and `ImperfectContainmentRelation` serve to represent, respectively, imperfect attributes and relations. Imperfect attributes may contain a range of numeric values, a set of numeric values, a set of strings, a set of numeric values with a certainty degree, a set of strings with a certainty degree, a linguistic value, or no value. Imperfect cross and containment relations may contain an instance relation with certainty degree or a set of instance relations.

5.2 iModeler Editor

The strategy described in section 2 has been implemented in a graphical editor based on Eclipse Modeling Framework Project (EMF) and Graphical Modeling Project (GMP) named *iModeler*. In addition, for the creation of *iModeler*, the project EuGENia was used in order to create the required GMP's components. *iModeler* serves to create imperfect models *(m_ζ)* that conform to *iMM*. This editor is also capable of validating the *ontological semi-conformance* of the m_ζ with respect to a MM_{EA} providing assistance to the user. The m_ζ has the appearance of an object diagram from UML (see Figure 3b), where each instance displays its metatype from the MM_{EA}, its attributes, and its relations. Imperfect attributes are decorated with a black rounded square. Imperfect containment and cross relations have respectively a filled and unfilled square in the side of the source. Another important characteristic of *iModeler* is the way it handles problems. This is achieved using EMF's validation elements, and our own validation engine based on Epsilon Validation Language (EVL)

For example, the MM_{EA} presented in Figure 3a has the types `Enterprise`, `BusinessApplication`, and `BusinessProcess`. In the construction of the m_ζ,

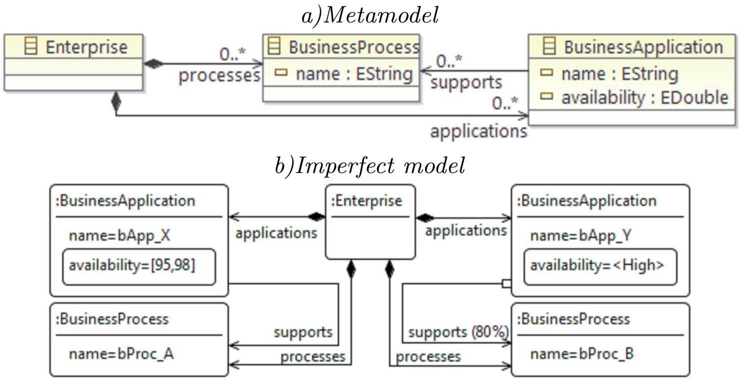


Fig. 3. Enterprise modeling example.

the modeler decides to include imperfection in some attributes and relations. The m_ζ (see Figure 3b) *conforms linguistically* to *iMM* and *semi-conforms ontologically* to MM_{EA}. It has the following instances: 1) `Enterprise` as instance of `Element` that contains all elements in the m_ζ; 2) `BusinessApplication` that represents the application *bApp_X*, created by the containment relation `applications` with the imperfect attribute `availability` with the range of values *[95,98]*; 3) `BusinessApplication` that represents the application *bApp_Y* with the imperfect attribute and relation: a) `availability` with the linguistic value <*High*>, and b) `supports` with the instance that represents the process *bProc_B*; 4) `BusinessProcess` that represents the process *bProc_A*, created by the containment relation `processes`; and 5) `BusinessProcess` that represents the process *bProc_B*. This example demonstrates several kinds of imperfection. In *bApp_X*, the attribute `availability` is imprecise; in *bApp_Y*, the attribute `availability` is vague, and the relation `supports` is uncertain.

6 Decision Trace on Imperfect Models

Modelers can include decision trace information into the m_ζ in order to represent the way in which the information was gathered, and the decisions to construct the model were made. The trace includes the enterprise sources, source consulting through observations, facts produced by observations, and decisions related with imperfect attributes or imperfect relations of the m_ζ.

In order to include all elements described above, we complemented the *iMM* creating one metamodel named *Extended Metamodel of Imperfection (EiMM)* (see Figure 4) for our tool *iModeler*; thus, the m_ζ now *conforms linguistically* to *EiMM*, The *EiMM* includes all elements presented in *iMM* and the following additional types: 1) `ModelExtension` serves as container for all other additional elements; 2) `Source` and its specializations `DirectObservation`, `Document`,

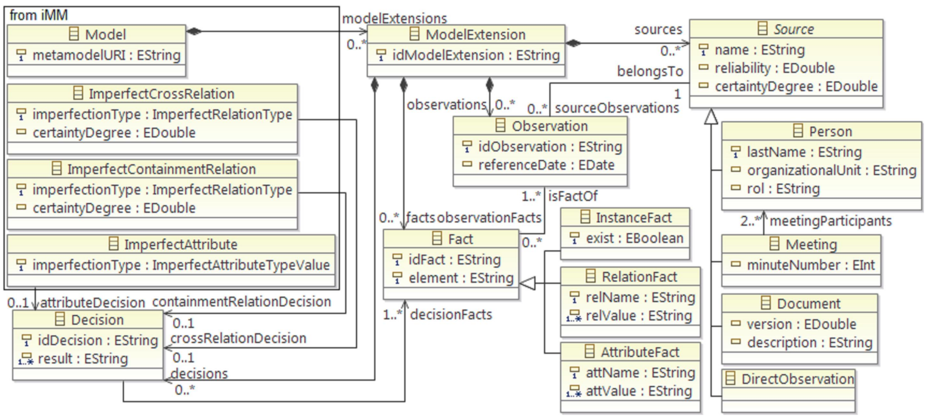


Fig. 4. Extended Metamodel of Imperfection EiMM.

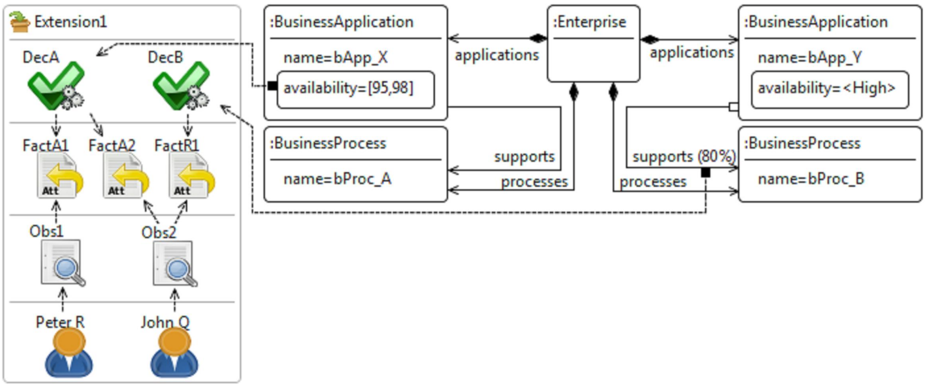


Fig. 5. Extended Imperfect Model Example.

`Meeting`, and `Person` serve to represent sources; 3) `Observation` serves to represent interviews, document revisions, and meeting acts; 4) `Fact` and its specializations `InstanceFact`, `AttributeFact`, and `RelationFact` serve to create registers of information obtained about instances, attributes, or relations of the enterprise; and 5) `Decision` serves to specify the decision made by modelers for imperfect attributes or relations. By means of the *EiMM*, *iModeler* allows modelers to include all necessary decision trace information into the m_ζ.

Continuing the example presented in the previous section, the modeler decides to include decision traces for the attribute `availability` of the *bApp_X* and for the relation `supports` of the *bApp_Y*. Then, the modeler does an interview to the CTO named "Peter R", who asserts that the availability of *bApp_X* is 95%. However, in an interview to the CIO named "John Q", the information obtained is that *bApp_Y* supports *bProc_B* with a certainty degree of 80% and the availability of *bApp_X* is 98%. Consequently, the m_ζ includes the following trace elements: 1) one instance of the type `ModelExtension` to include all decision trace elements; 2) two instances of the type `Person` with information about the CTO and the CIO; 3) two instances of the type `Observation` with the information of the interviews; 4) two instances of the type `AttributeFact` that specifies the availavility of *bApp_X*; 5) one instance of the type `RelationFact` that relates the *bApp_Y* with the *bProc_B*; and 6) two instances of the type `Decision` with the results of the decisions made by the modeler for the imperfect attribute `availability` and the imperfect relation `supports`. Figure 5 represents the imperfect model with decision trace of the example.

7 Conclusions

In the Enterprise Architecture context, models are built using information provided by various and heterogeneous sources. These sources usually do not have perfect information, so enterprise models might not represent the enterprise correctly; thus, analysis results based on the mentioned models can be incorrect.

Imperfect models represent and structure imperfect information while enabling modelers to keep all the information provided by sources about the elements' attributes and relations. Modelers can also include decision trace information. Based on the imperfect information and decision trace information, analysts can perform better analysis processes with a higher level of reliability.

The solution strategy presented in this paper represents the imperfection in EA models based on the creation of models that *conform linguistically* to *EiMM* and *semi-conform ontologically* with the MM_{EA}. The m_ζ is created using the tool *iModeler* that supports imperfect attributes and relations, and also allows the inclusion of decision traces.

In this work, we have focused on providing mechanics to properly manage and keep the information about imperfection. It was necessary to conceptualize these kinds of imperfection, while understanding the impact from the sources in the quality of the models. Based on these results, we will start to study how to analyze the enterprise, taking into account the mentioned imprecision while providing better conclusions.

References

1. Seidewitz, E.: What models mean. Software, IEEE **20**(5) (2003) 26–32
2. Ludewig, J.: Models in software engineering–an introduction. Software and Systems Modeling **2**(1) (2003) 5–14
3. Bézivin, J.: On the unification power of models. Software and Systems Modeling **4**(2) (2005) 171–188
4. Henricksen, K., Indulska, J.: Modelling and using imperfect context information. In: Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, IEEE (2004) 33–37
5. Kuhne, T.: Matters of (meta-) modeling. Software and Systems Modeling **5**(4) (2006) 369–385
6. Gómez, P., Sánchez, M., Florez, H., Villalobos, J.: Co-creation of models and metamodels for enterprise architecture projects. In: Proceedings of the 2012 Extreme Modeling Workshop. XM '12, ACM (2012) 21–26
7. Smets, P.: Imperfect information: Imprecision, and uncertainty. Uncertainty Management in Information Systems **1996** (1996) 225–254
8. Hayashi, T., Wada, R.: Choice with imprecise information: an experimental approach. Theory and Decision **69**(3) (2010) 355–373
9. Li, X., Dai, X., Dezert, J., Smarandache, F.: Fusion of imprecise qualitative information. Applied Intelligence **33**(3) (2010) 340–351
10. Hunter, A., Nuseibeh, B.: Managing inconsistent specifications: reasoning, analysis, and action. ACM Transactions on Software Engineering and Methodology (TOSEM) **7**(4) (1998) 335–367
11. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Inconsistency Tolerance. Springer (2005) 191–236
12. Afshani, J., Harounabadi, A., Dezfouli, M.A.: A new model for designing uncertain enterprise architecture. Management Science **2** (2012)
13. Dai, J., Xu, Q.: Approximations and uncertainty measures in incomplete information systems. Information Sciences (2012)

Challenges of EA Methodologies Facing Progressive Decentralization in Modern Organizations

Irina Rychkova[1], Jelena Zdravkovic[2], Thomas Speckert[2]

[1] Centre de Recherche en Informatique, Université Paris 1 Panthéon - Sorbonne, 90 rue Tolbiac, 75013 Paris, France
irina.rychkova@univ-paris1.fr
[2] Department of Computer and Systems Sciences, Stockholm University, Forum 100, SE-16440, Kista, Sweden
jelenaz@dsv.su.se, thsp7525@dsv.su.se

Abstract. Enterprises with transparent boundaries, decentralized organizational structure, and constantly increasing requirements on IT flexibility, is a novel generation of organizations of 2020s. This paper elaborates on how the Enterprise Architecture (EA) can better support such organizations. We analyze three types of organizational structure: centralized, federated and decentralized. First, we identify the concepts that link organizational structure, IT Governance and EA. Then we use these concepts to identify conceptual problems related to IT decentralization, and to propose solutions. We illustrate our findings with a real case of a Higher Education organization in Sweden.

Keywords: Enterprise Modeling, Enterprise Architecture, IT Governance.

1 Introduction

According to [1], in the coming years enterprise software systems will not be able to continue to evolve along the beaten paths, because there is an urgent need for new directions in the ways enterprise software is conceived, built, deployed and evolved. This contention is becoming materialized even presently, when the boundaries of companies gradually fade away paving the road to *liquid enterprises* having fuzzy boundaries in terms of human resources, markets, products and processes which require adequate Internet-based Enterprise Systems.

Decentralization of organizations and subsequent changes of their management require major changes in organizations' processes and heavily involves the use of IT. Between traditional (highly centralized) and decentralized or "liquid" enterprise, many other organizational structures can be identified [2]. In these work, we analyze three forms of organizational structure: *centralized, federated and decentralized*.

This work studies the conceptual differences between these organizational forms with focusing *on how these differences affect creation, maintenance and evolution of the IT within the corresponding types of organizations*. Our objective is to make an explicit link between the structure of an organization and its EA, ensuring thus better support for federated and decentralized organizations. We envision an architecture-driven corporate and IT governance involving adequately performed communication with a set of policies, multi-level decision making, knowledge management, automation of tasks by taking advantage of IT infrastructure, human management, etc.

The main research question addressed in this paper is: How to integrate the decentralization concepts into EA methodologies? The proposed solution follows Design Science research framework [18], which suggests that an innovative solution is proposed to solve a problem of general interest. Following the framework, in our study (i) we identify a problem from the real world - a need to support the modern types of enterprises characterized by increasing decentralization and demand in flexibility and agility of their IT; (ii) then we define a relevant knowledge base for our research that is grounded on organizational science, and the enterprise architecture discipline; (iii) we build design artifacts: the two constructs to be used for reasoning about organizational structure in general, and IT organizational structure in particular; (iv) to evaluate the created artifacts, we apply them in the environment of a Federated Organization in the Swedish Higher Education sector.

This paper reports on the research in progress and will be organized as follows: in Section 2, we outline the theoretical foundations for this work and discuss the related works. In Section 3, we define a relationship between a structure of organizational IT and an EA: first, we present the concepts of center and steering forces that link organizational structure, IT Governance and EA; than we use these concepts to identify conceptual problems related to IT decentralization and to propose solutions. These findings are illustrated with a real organization case in Section 4, which is followed by our conclusions and the direction of future work in Section 5.

2 Theoretical Foundations and Related Work

In this section brief overviews of the topics and the results related to the research of this paper are presented.

2.1 Centralized, Federated and Decentralized Organizations

The organizational structure defines the rules according to which allocation of responsibilities and resources, coordination and supervision, is made for an organization (and - in case of IT - for the IT). Many popular organizational types are defined in the literature [3]-[6] In this work, we focus on three types of organizational structure: centralized, federated and decentralized organizations [2].

Centralized organizations lean towards a vertical style of coordination [7], characterized by formal authority, standardization, and planning.

Decentralized organizations lean towards lateral coordination, characterized by meetings, task forces, coordinating roles, matrix structures, and networks [7]. An example of decentralized organization is a collaboration of partners working on a concrete set of problems (e.g. research collaborations, virtual labs) or forming in response to a particular customer need or market situation (e.g. virtual organizations, coopetitions [8]). Besides this collaboration, missions and objectives of each partner can be completely different and even concurrent.

Federated organizations combine characteristics of centralized organizations (e.g. centralized planning, standardization, etc.) and decentralized organizations (e.g. local leadership, competitive local objectives, etc.). One example of federal organizations is a research institution that is formed by multiple schools, centers, and labs.

2.2 Enterprise Architecture (EA)

The role of EA discipline is to provide the organizations with a roadmap for creation and evolution of their information systems. EA of an organization changes and grows together with the organization, its structure, vision and operating model [9].

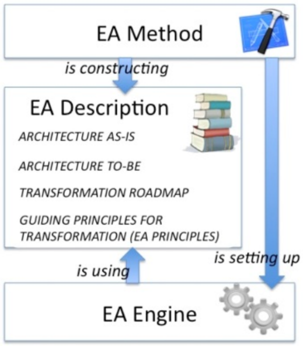


Fig. 1. Enterprise Architecture of an organization contains three interrelated parts: EA Method , EA Description and EA Engine .

EA "defines the underlying principles, standards, and best practices according to which current and future activities of the enterprise should be conducted" [10]. EA methodology and tools produce artifacts to specify the current state of a company's architecture ("as-is"), the target architecture ("to-be"), identify how to best cross the gap between them (architectural roadmap), and set up the standards and rules to follow during this transformation (EA principles). These elements are often addressed in literature as *EA description*; the process that an organization has to execute in order to obtain its EA description is called *EA method* (Fig.1). A traditional EA project consists in implementing an EA method and producing an EA description. To assure that the organization will continuously follow the EA principles and achieve the designated goals (architecture "to-be") a third element has to be defined: *EA engine*. The presence of this element in our model in Fig. 1 reflects the fact that EA is not static: it makes the organization change while changing itself over time. Dedicated structures and procedures have to be defined in an organization in order to continuously steer this organization towards its target architecture.

2.3 IT Governance

According to [17], IT Governance is a subset discipline of corporate governance focused on information systems and their performance and risk management. The discipline describes how people authorized over some domain of business should consider IT in the monitoring, control, and improvement of the business. Architecture governance is a key aspect of IT Governance – it is responsible to create and manage policies for the structure and content of IT in an organization, and to enable their reuse in the form of best practices. Service Oriented Architecture (SOA) governance is a well-known example where the architecture, i.e. SOA and further up an EA that incorporates SOA, drives IT governance to ensure service orientation.

2.4 Peer-to-Peer

In [2], we claim that the structured and disciplined approach to IT evolution not necessarily has to rely upon IT centralization: novel EA concepts are needed to ensure the harmonization of development and evolution of IT with the properties of decentralized and federated organizations. We argue that peer-to-peer is a relevant concept to decentralization in EA for two reasons. First, units in decentralized organization are able to contribute to the enterprise in a manner that is completely up to them. This is similar to peers in a peer-to-peer system, where the peers participate in a voluntary manner. Second, the challenge that peer-to-peer systems overcome is similar to decentralized organizations: "to figure out a mechanism and architecture for organizing the peers in such a way so that they can cooperate to provide a useful service to the community of users" [11]. Therefore, we consider peer-to-peer principles [12], [13] applicable to EA for enhancing their support of decentralization.

3 Organizational Structure and EA

The objective of EA methodologies created in early 1990s was to align the IT capabilities with Biz needs via IT centralization. The main price to pay for IT centralization was the loss of flexibility and the inertia in decision making in IT. By that time, however, this was much less critical than to make the IT "disciplined" and to justify the investments in IT. Today, the flexibility in IT becomes more and more strategic. For modern organizations with transparent boundaries, it is simply impossible to centralize IT for literally independent partners. On the other hand, it is still crucial to maintain "disciplined" approach in IT evolution so that the partners not only remain independent but could also efficiently work together as a "virtual whole".

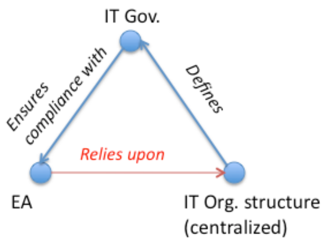


Fig. 2. IT Organizational structure, IT Governance and EA form a triangle where EA relies upon the IT Org structure.

The works presented in [4], [14], [15] and [16] focus on the relation between the structure of an organization and its IT. Following these works, we claim that the notions of Organizational Structure (in IT), IT Governance, and EA are interrelated: IT governance is defined by the IT Org. Structure and has to comply with the vision of Architecture to-be and the EA principles; EA principles, in turn, should reflect the style of IT Organizational Structure. This relations form a triangle as shown in Fig. 2.

The question is: *how EA should reflect the change in the IT Org. Structure in order to support the "disciplined" IT evolution?* Upon which alternative mechanisms should EA rely when centralized strategic and resource planning is getting replaced by local planning; does central management replaced by the management on the operational level and centralized coordination and top-down decision making gives its way to self-organization and ad-hoc partnership?

To answer these questions, we define the concepts of *center* and *steering forces* (Section 3.1), and using these concepts, we represent the three types of organizational structures (Section 3.2). Than we formulate the problem related to mismatch between the organizational structure and the EA in use (Section 3.3).

3.1 Concepts for Reasoning about Decentralization

We consider three generic forms of organizational structures: centralized, federated and decentralized. We focus on the elements of these structures that impact the definition (EA method) and then implementation (EA engine) of the EA principles driving the organization to its target architecture: the *center* and the *steering forces*.

We define **Center** as a part of organization (a person, a group, or a unit), which plays the role of a leader, supervisor or coordinator, and possess some power to steer the other parts of the organization. Center can be implicit or explicit. Organizations with centralized IT (Fig. 3-a), have explicit center (e.g. EA department; EA steering committee etc). This center initiates, supervises and validates the changes in the organizational IT and in the EA itself. It steers all the organizational units by setting rules and checking for compliance. We can also say that there exist *steering forces* between the center and the non-central units.

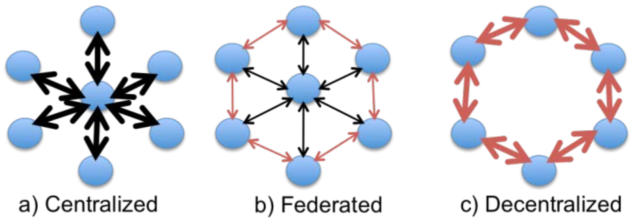


Fig. 3. Three types of organizational structure described with the notion of center and steering forces. Organizational units are depicted with filled circles. A circle in the centre stands for the "Center". The arrows relating the circles depict the steering forces.

Steering forces can be defined as explicit and implicit protocols, policies, rules and procedures regulating the flow of communication and control between organizational units. These forces can be characterized by their direction (top-down, bottom up, sideways) and their strength. In organizations with centralized IT, the strong steering forces connect the center with the other units forming a hierarchy (radial forces). In Fig. 3-a, a simple model with two levels of hierarchy is presented. These forces can be both top-down (supervision, decision making, task/resource planning) and bottom-up (local initiatives leveraged to the center for approval).

In federated organizations (Fig. 3-b), the center remains explicit but the radial steering forces connecting the center with other units are weak since decision-making and prioritization in IT can be also done locally. On the other hand, sideways - steering forces appear in this model since more and more interactions are joint projects emerging locally, between units and without passing by the center.

In decentralized organizations (Fig. 3-c), the center disappears (or becomes implicit) and neither overall commitment to a given set of EA principles nor centralized control over IT evolution can be ensured. The only type of steering forces that makes the organizational IT evolve is strong sideways forces.

3.2 EA for Centralized, Decentralized and Federated Organizations

Based on the conceptual representation of the three organizational types from the previous section, we explore how these characteristics of centralized, federated and decentralized organizations can be reflected by the EA methodologies.

Centralized organization:

EA Method should set up a structure aligned with the structure in Fig. 3-a: to define a project leader or a sponsor (center) that will occupy a high hierarchical position in the organization and will automatically provide the top-down steering forces (decision making, resource allocation); to identify data/process owners in the local units that would provide the bottom-up steering forces and actively participate in the EA creation; to assign responsibilities and define protocols that would help to reach a consensus about the EA description to produce (radial steering forces).

EA Description has to focus on company-wide, long-term master plan for IT development that fits the global vision of the organization. EA principles have to define a single standard to be followed by all organizational units.

EA Engine, similarly to EA method, needs an explicit center (controlling authority) and strong radial steering forces (protocols, instrumentalized processes and resources to ensure compliance with EA) to be defined. The center will steer the organization by promoting initiatives, making decisions and validating results. The organizational units will leverage their initiatives to the organization level (bottom up) for further approval.

Federated Organization:

EA Method should set up a structure aligned with the structure in Fig. 3-b: define a project leader (center) who will ensure the alignment between the EA project and the objectives of the organization. Since the radial steering forces are weak and can only partially ensure communication and coordination of efforts between organizational units, no centralized control or validation of EA description can be achieved. Therefore, sideways steering forces have to be developed in to complement the lack of radial steering forces. Within an EA methodology, new protocols for negotiation, information sharing and cooperative decision making have to be elaborated.

EA Description has to focus on company-wide, short-term master plan for IT development that fits the global vision of the organization. EA principles should support variability in processes and resources instead of a single standard that “fits all”. For example: the central unit decides on generic process and resources, but the units implement their own variants.

EA Engine should rely upon both center and local leadership and define two types of steering forces complementing each other. EA methodology has to specify tools and activities based, for example, on the peer-production principles, and supporting both centralized and user-driven (collaborative) change management.

Decentralized Organization:

EA Method should set up a structure aligned with the structure in Fig. 3-c, where no center is explicitly defined and only cooperation-driven sideways steering forces are enabled. While possibly maintaining their own, local EA, the partners in a decentralized organization has to be able to “connect” their architectures and to achieve interoperability. EA methodology should provide metrics for assessing the interoperability and alignment between local EA and global EA.

EA description has to focus on local short-term master plans for IT development that are aligned with the objectives of an organization (a partnership). Organizational EA principles should support variability in processes and resources allowing the partners to implement their own variants of a given process with respect to their local architectures and local EA principles.

EA Engine should rely upon strong sideways forces, where EA methodology has to specify tools and activities supporting user-driven change management.

3.3 Mismatch Between the IT Organizational Structure and the EA on Place

Based on the theory above, many practical problems related to the EA implementation can be explained by a mismatch between the IT organizational structure and the EA in place. In particular, we identify two types of problems:

- *Problem A:* IT initiatives fail and decisions in IT become inefficient when federated/decentralized organization uses the EA that (still) relies on centralized coordination and control;
- *Problem B:* Poor or no strategic alignment can be guaranteed when centralized/federated organization relies uniquely on local leadership and implements solutions that require purely decentralized management.

We reformulate these problems in terms of misbalanced steering forces in the organization. This leads us to a solution that can be summarized as follows:

Problem A: A misbalance between the organization with *weak or non-existing* radial (top-down and bottom-up) steering forces and its EA that relies upon *strong* radial forces only; The solution is to revise EA Method and EA engine by involving sideways steering forces that would compensate the lack of radial forces. More concretely, the organization has to replace some (al for decentralized organizations) mechanisms of centralized control and coordination by their decentralized equivalents (e.g. cooperative decision making, peer-production etc)

Problem B: A misbalance between the organization with *weak or non-existing* sideways steering forces and its EA that relies upon *strong* sideways forces; The solution is to revise EA Method and EA engine by involving radial steering forces that would compensate the lack of sideways forces. More concretely, the organization has to reinforce the mechanisms of centralized control and coordination.

In the section below we illustrate our theory on the case of an organization for Higher Education reflecting a federated organizational structure.

4 Case Study

We have analyzed a prominent university for higher education in Sweden. As common, the university includes a number of units - faculties, and faculty departments. Nowadays, the units are becoming more independent than before due to several factors:

- Geographical dislocation. Some faculty departments have been moved out of the main university campus. An example is the Computer and Systems Sciences department located in Kista, the leading Swedish IT cluster. This proximity enables cooperation between IT companies and students through mentoring programs, internships, graduate work opportunities, guest lectures, etc.
- Decentralization of management. Coordination and decision-making are through delegation pushed down to the faculties and further to faculty departments. Concretely, the decisions are delegated by the principal to the faculty boards and deans, and to the faculty departments' heads and their administrations.
- Both formal and informal communication patterns. Formal hierarchical communication from the faculty to its departments and informal direct communication between the departments are present. For example, the administrative tasks (e.g. registration for graduate courses, or postgraduate research etc) is primarily formal, whereas the course curriculum can be established between departments cooperatively, using informal communication links.

According to the theory presented in Section 3, the organization above is a *federated organization* with explicit center and both radial and sideways steering forces defined. Below, we present some examples illustrating IT projects conducted by the university and the difficulties encountered. We will explain these difficulties using our theory and demonstrate that their origin is a mismatch between the organizational structure in place (federated) and the EA engine exploited for making decisions/developing solutions.

Example 1: Room reservation (over-centralization). The central (university) IT department has purchased a packaged IS to be used for room booking university wide. Some departments already had their local solutions for room booking, which were better adapted for their needs. As a result of this initiative, the departments ended up paying for the new system (due to centralized budgeting) but kept their own system and refused integration (due to decentralized decision making).
This example illustrates the Problem A from the previous section - A mismatch leading to inefficient and finally abandoned solution. The decision about purchasing the university-wide system relied uniquely on radial forces (centralized), whereas sideways forces (negotiation with departments, collaborative requirements gathering, etc) have not been exploited at all.

Example 2: Publication cataloguing (over-decentralization). In the past, some faculty departments developed local IT solutions for cataloguing research publications. Over time, this multitude of local solutions became problematic due to numerous mappings and data inconsistencies. Recently, the university brought the decision to allow the faculties and their departments to continue to store and assess publications' data in the way that suits best to them, while requiring a workable mapping to a central catalogue structure that follows the required standard regulated on the state level. The coordination and decision-making here exploited the sideways forces only. Since the publications meant to reflect a common face of the university - their central management using radial forces was desirable.

In organizations with the federated structure, the problems above can be avoided if the EA methodology properly integrates the decision-making patterns that rely on both radial and sideways steering forces. In the first example, the centralized EA principles have been implemented (whereas the real organizational structure is federated). The correct solution would be to exploit both radial forces and sideways forces (to involve the departments into cooperative requirements gathering, solution evaluation etc). In the second example, in contrast, the decentralized EA principles have been implemented. The correct solution could be, for example, to centrally define a common standard for publications (radial forces) and to let the faculties implement this standard in the way that fits their local architectures.

5 Conclusion and Future Work

This paper outlined the challenges related to increasing demand in process flexibility and the emergence of novel generation of organizations with transparent boundaries. To meet these challenges, the IT structure of organizations has to change: the centralized organizations characterized by strong top-down coordination and control, now tend to move towards more agile (decentralized) structures, where new communication, coordination and decision making patterns are used. We claim that the structure of organizational IT not only defines the IT Governance style of the organization, but it also has to be explicitly reflected by the Enterprise Architecture of the organization and supported by an EA methodology.

In this work we defined the concepts of center and steering forces and modeled organizations with different degree of centralization in their IT: centralized, federated and decentralized. Using these concepts, we identified the problems that result from mismatch between the organizational structure and the EA in place. As we explained in Introduction, the work follows the Design Science IS research framework [18], in the problem definition, the use of relevant knowledge base, development of main research artifacts - the two constructs (center and steering forces) which allowed us to identify the problems related to a misfit between IT organizational structure and EA in use, and to evaluate in on an real case in the Swedish Higher Education sector.

For the future work, we plan to elaborate on the proposed concepts and to identify metrics that would allow us to assess the centralization/decentralization more precisely (to measure the strength of steering forces, etc.). We also envisage to study the concrete mechanisms and patterns for communication, coordination and decision making in centralized, decentralized and federated organizations, and to see how they can be transformed into concrete EA principles or explicitly integrated into EA methodologies. For example, process variability as a mechanism to handle local differences while complying with global standards in federated organizations.

References

1. Missikoff, M.: The Future of Enterprise Systems in a Fully Networked Society. Proc. of CAiSE 2012, Lecture Notes in Computer Science, 2012, Volume 7328/2012, 1-18, DOI: 10.1007/978-3-642-31095-9_1, (2012)
2. Speckert T., Rychkova I., and Zdravkovic J.: On the Changing Role of Enterprise Architecture in Decentralized Environments: State of the Art. To appear in proceedings of 8th International Workshop on Trends in Enterprise Architecture Research (TEAR). 9-13 September, Vancouver, BC, Canada (2013)
3. Mintzberg, H. The structuring of organizations: A synthesis of the research. University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship (1979)
4. Weill, P. Don‘t just lead, govern: How top-performing firms govern it. MIS Quarterly Executive, vol. 3, no. 1, pp. 1–17 (2004)
5. Thompson, J. D. Organizations in action: Social science bases of administrative theory. Transaction Pub (1967)
6. Robbins, S. P. Organizational behavior: Concepts, controversies and applications: Australia and New Zealand. Royal Victorian Institute for the Blind. Special Request Service (1997)
7. Bolman, L. G. and Deal, T. E. Reframing Organizations, 4th ed. San Francisco, California: John Wiley & Sons (2008)
8. Bengtsson, M. and Kock, S. Coopetition in business networks–to cooperate and compete simultaneously. Industrial marketing management, vol. 29, no. 5 (2000)

9. Weill, Peter, and David Robertson. Enterprise Architecture as Strategy. Harvard Business Press (2006)
10. Schekkerman, J. How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework, 2nd ed. Trafford Publishing (2004)
11. Saroiu, S., Gummadi, P. K. and Gribble, S. D. Measurement study of peer-to-peer file sharing systems, In Electronic Imaging, International Society for Optics and Photonics, pp. 156–170 (2002)
12. Benkler, Y. The wealth of networks: How social production transforms markets and freedom. Yale University Press (2006)
13. Aberer, K. and Despotovic, Z. Managing trust in a peer-2-peer information system. In Proceedings of the tenth international conference on Information and knowledge management. ACM, pp. 310–317 (2001)
14. Rockart, J., Earl, M., and Ross, J. Eight imperatives for the new IT organization," Sloan management review, pp. 43–56 (1996)
15. Fulk, J. and DeSanctis, G. Electronic communication and changing organizational forms. Organization science, vol. 6, no. 4, pp. 337–349 (1995)
16. Osterloh, M. and Frey, B. S. Motivation, knowledge transfer, and organizational forms. Organization science, vol. 11, no. 5, pp. 538–550 (2000)
17. Weill, P. and Ross, J. W. IT Governance: How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, Boston (2004)
18. Hevner, A. R., March, S. T., Park, J., & Ram, S. Design Science in Information Systems Research. *MIS Quarterly , 28* (1), pp. 75-105 (2004)

Connecting the Dots: Examining Visualization Techniques for Enterprise Architecture Model Analysis

David Naranjo, Mario Sánchez, and Jorge Villalobos

Department of Systems and Computing Engineering
Universidad de los Andes, Bogotá, Colombia
{da-naran,mar-san1,jvillalo}@uniandes.edu.co

Abstract. The discourse of Enterprise Architecture is based on modeling and performing 'holistic' (multi-layer) analyses. However, view centered methodologies offer a partial glimpse of the overall architecture, and current tools do not bring an explicit method of navigating the underlying model. Considering that we need a starting point for analysis and explore the whole model in order to drill down on more specific analysis techniques, we compare overview visualizations depending on topological properties of the model and a set of domain-specific requirements. Four techniques are examined, and they visualize five Analytical Scenarios that represent typical questions that could arise on a EA diagnostic.

Keywords: EA Visual Analysis, Visualizations, Enterprise Model Topology

1 Introduction

Nowadays, as we are more capable of capturing all kinds of information, the need to make sense of it is most compelling. Instead of facing the problem of collecting this data, the main issue is to propose methods and models, which can turn it into reliable and provable knowledge [1]. In complex domains we have seen the rise of experts (we usually call them *analysts*), that process this information using certain reasoning process, thus providing insights that generate knowledge and serve as input for decision making.

In this aspect, Enterprise Architecture (EA) has been proven as a valuable tool for aligning business strategies with Information Technologies projects and infrastructure. A key function of EA is the static analysis of current architectures in order to drive IT projects and close gaps. Given its broad scope –e.g. impact analysis vs. business process analysis–, and the use of different techniques, ranging from ad-hoc to quantitative analysis, it is important to point that no common definition of the term *EA Analysis* has emerged [2].

However, we invite the reader to think of EA analysis as more than a static process with a collection of techniques. We think that the analyst, his own experience and reasoning power are 'the spice' that give flavor to analysis, *making sense* of the information of the enterprise, that means, turning it into reliable and comprehensible knowledge [3], with the objective of providing valuable insights that support decision making at the different organizational levels. For this reason, making assessments of architectures is a core competency for an architect.

As all the collected information from the different domains of the organization –the EA model– is more detailed and complex, assessments are more difficult to perform. Given the iterative and reflective nature of this analysis process, EA Frameworks such as DoDAF emphasize on tool-assisted and tool-supported analyses whenever possible [4]. However, in our previous work [5] we have seen that there is still no starting point for analysis both in EA commercial products and in the research community. Most tools offer queries and partial views as the only means to perform analysis, with little support to ad-hoc analyses, and offering no traceability on their outcome and impact on the architecture.

In decision-making, "the useful information is drawn from the overall relationships of the entire [data] set" [6]. For this reason, a major challenge in EA is to find effective techniques to visualize enterprise architectures as a whole. Overview/holistic visualizations of the architecture harness the cognitive power of the visual domain and make more easy to find patterns and explore the architecture. This brings us to our research question: *How effective can be existing visualization methods when we apply them in the context of EA analysis?*

With the purpose of illustrating the different subjects at hand, we will provide the results of our experimentation with four VA techniques, that are widely used in other domains, to visualize the enterprise model of our case study. This experimentation is supported by five Analytical Scenarios, which are typical questions that can be formulated throughout the EA Analysis process. Each visualization is adapted to show the maximum amount of information, taking care of following the Visual and EA Analysis requirements defined in [5].

The structure of this paper is as follows: First, we will characterize enterprise models and their topology (Section 2) and introduce EA Analysis supported by Visual Overview (Section 3). In Section 4, we will explain the Case Study and Analytical Scenarios selected for this research. Finally, in Section 5 we will perform an evaluation of the effectiveness of the selected techniques for overview analysis.

2 The topology of Enterprise Models

Enterprise Architecture has been supported by different paradigms of Enterprise Modeling [7, 8, 9]. For this paper, we will adopt the following notion:

Definition 1. *An Enterprise Model (EM) is a representation of the information in all aggregate artifacts (including documents, diagrams, deliverables, or any structured piece of enterprise knowledge) that are relevant for an enterprise [10], which may come from a variety of sources, and is intended to be used by people different backgrounds.*

Analysis methodologies over these models encourage the identification of critical elements by examining their incoming and outgoing links [11], or in general, by asserting topological properties of these models. Thus, visualizing the global structure of EA models can lead to structural assessments of the model, such as discovering interesting elements and groups, or following paths between them.

In order to support this kind of assessments, we make a bridge between EM analysis and studies in graph analysis and complex networks. This will allow us to take advantage of existing properties of this kind of networks, and to formulate interesting topological properties specific to EMs. A complex network represents a complex system, where the relationships amongst the components of the system are usually more important than the components themselves [12].

Our premise is that EMs are complex, and we can define their complexity in terms of its number of relations, as it surpasses the number of elements of the model. We will describe five properties of EMs inspired on this premise.

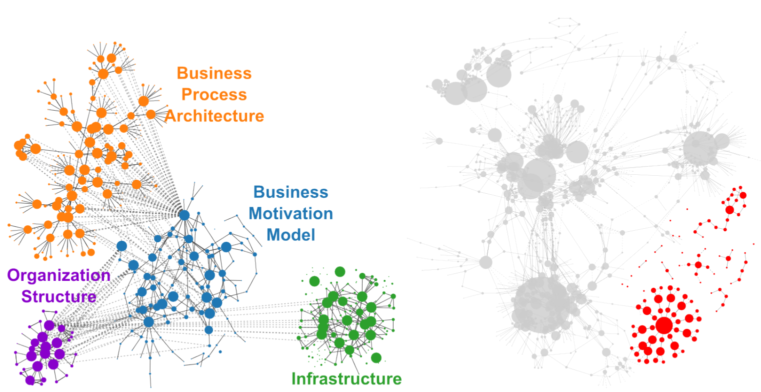
P1 - Enterprise models grow over time

While EMs are abstractions [13] that should be kept simple and small [14], our perception is that these models grow over time, given the incremental nature of EA projects, and taking into account the role of EA as a continuous business function. Also, EMs are an asset that represents enterprise knowledge and therefore should be managed [13].

As semi-automated mechanisms for collecting architectural information are more widely used, enterprise model scalability becomes a critical issue. For instance, Binz et al. point to the complexity of Enterprise Topologies –a snapshot of all services and applications in an enterprise, together with their supporting infrastructure and relations [15]– that may consist of thousands to millions of nodes.

P2 - Enterprise models are structured

One of the goals of EA is to provide alignment of the different business and technological resources, by establishing relations between the elements on different architectural domains.



(a) Detail of some domains on an EM (b) Weak Components of the model

Fig. 1. Force directed graphs consist of nodes whose position is given by a force model, considering the arcs that connect them. Nodes represent model elements, while arcs represent relations.

In EMs, this organizing structure is explicit: they conform to a **metamodel** that integrates the different **domains** of the organization and offers a clear view on the structure of and dependencies between relevant aspects of the organization.

Definition 2. *An **enterprise metamodel** is the common language on which the enterprise model is 'spoken' by the means of concepts, relations and constraints that bring consistency to the architectural description.*

Instead of a "one-size-fits-all" metamodel, recent approaches to enterprise modeling focus on integration [8], adaptability [16] and multi-perspective modeling [9] by the integration of standards and reference models, complemented with domain-specific models, specifically tailored to support the management tasks [17] and analysis needs of the enterprise. Interesting dependencies for overall analysis are the ones that connect the different domains/perspectives such as Goal, Process, Organization Structure, Infrastructure or Application models (see Fig. 1(a)).

Definition 3. *An **enterprise domain** is a subset of the enterprise metamodel that aggregates similar concepts, forming communities that are connected between them by inter-domain relations.*

Taking into account that we are interested in the relations between domains, unaligned/isolated elements are meaningless on their own. Therefore, a first step when analyzing the model is to identify 'islands' or weak components, i.e. disconnected groups of elements (see Fig. 1(b)).

P3 - Enterprise models are semi-hierarchical

Organizations are complex systems that are conformed of a number of parts which are inherently hierarchical, i.e. they are composed of interrelated subsystems, and interact in a non-simple way.

Under this light, EMs are semi-hierarchical, as each domain manages different levels of detail on its concepts by composition (i.e. aggregation) relations. For instance, Process Models are defined under a vertical scope, where each depth level also involves more granular sub-processes and activities (see Fig. 2(a)).

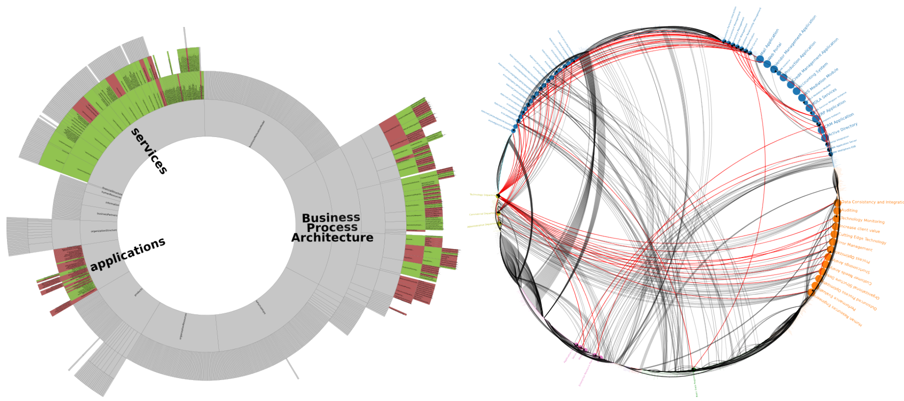
An advantage of this property is that hierarchies are one of the most recurring information structures in computing, and offer a natural way for navigating the model.

P4 - Relation have their own semantic

EA Analysts need to find pathways between the different model elements, which is important when they want to assess the overall impact of a change in the architecture, e.g. the cost of making changes to enterprise-wide software systems [17]. This assessment is made almost intuitively: even without previous knowledge, most pairs of vertices in complex networks seem to be connected by a short path.

Nevertheless, these paths are made by different kinds of relations, e.g. composition and association relations. The latter provide a horizontal scope to the model, i.e. a 'is related to' semantic, which is mostly given by their name, and depends on the metatype of their target. For instance, an Application may be *used by* a Role, which in turn *is responsible* for some Resources, and *belongs to* an Organization Unit.

For this reason, it is common in analysis techniques to assign weights or other attributes to relations, translating this abstract semantic to more quantitative means.



(a) Sunburst (b) Radial Graph

Fig. 2. Sunbursts offer an overview of the whole model, while being constrained by a circumference. Radial graphs position elements in a circumference, while inter-domain relations are displayed inside the circle.

P5 - Not every element has the same importance

Analyzing resilience of enterprises is important to discover points of failure under different domains, such as vulnerabilities in infrastructure security, core processes in business process analysis, or even overly-coarse services that need to be decomposed. For this reason, it is crucial to prioritize which elements have more influence in the topology of the model. This is useful when an analyst wants to assess the connectivity of the network: the removal of vertices or groups may affect, or even dismantle its structure.

In this aspect, several ranking algorithms exist to discover the relative importance of elements (e.g. PageRank, HITS, SALSA) based on their occurrence and connectedness. However, importance may be driven also by other semantic criteria that are based in the metamodel, not in the model itself. For instance, when analyzing which processes are supported by IT, we want to focus on Macro-Processes, Processes, and IT elements such as Applications and Services (see Fig. 2(b), where these elements are given some emphasis in size).

3 Giving a shape to the architecture

We often see the use of the term *Analysis* on different levels of granularity: as the overall process and also as a concrete strategy or technique. On the remaining of this paper, we will use the following definition:

Definition 4. ***EA Analysis:*** *All of the processes that transform architectural data into useful information [4], which serves as a basis for bringing an assessment or concept.*

In concrete, we are interested in the *static analysis* of this information, i.e. a snapshot of the architectural description, to offer a value judgment of a given state of the architecture.

In previous work [5], we evaluated a set of tools that support EA modeling and Analysis, and suggested that Visual Modeling Languages (VMLs) and queries are just a localized/filtered view on the model. As DoDAF [4] points out, "Architectural views are no longer the end goal, but are described solely to facilitate useful access to information.". Visual exploration of the whole model is mostly neglected, even tough it is useful when a person simply does not know what questions to ask or when the person wants to ask better, more meaningful questions [18].

Instead of using queries as the main way to inspect the model, our focus is on total visualizations that provide an overview of the EM and help the analyst on finding new properties of the overall set, thus we encourage top-down analysis for pattern discovery.

Definition 5. *A **visualization technique** can be seen as the combination of **marks** [19, 20] (atomic graphical elements, e.g. circles, squares, or lines), a* **layout** *algorithm, some **visual attributes** (e.g. color, size, shape), a set of supported **interactive operations** and a **mapping** between data and such visual attributes.*

Considering the overwhelming number of visualization techniques available, in our case is certainly complex to assess which ones can be applied for such a specific application domain. For this reason, instead of approaching this problem with an existing taxonomy, we employ a set of Visual Requirements (see [5]) as the criteria for selecting interesting visualization techniques for showing the landscape of the architecture.

In our search for visualizations, we analyzed the different artifacts, views, viewpoints, diagrams, pattern catalogs and tools from our domain (EA), and confronted them with visualization taxonomies and tools/ frameworks. The selected techniques were: 1) Force-directed Graphs, 2) Radial Graphs, 3) Treemaps, and 4) Sunbursts.

4 Analytical Scenarios for EA Visual Analysis

In order to evaluate and interact with each visualization, we visualized the EA model of a fictional company from our EA Laboratory, *Muebles de los Alpes*.

The EA metamodel of this Retail/Manufacturing company incorporates concepts from several metamodels (e.g. Business Motivation Model, ArchiMate, BPMN, TOGAF), as well as specifically tailored representations of standards, frameworks such as Service Oriented Architecture, or even formalizations of domains such as Applications, Information, Organizational Structure, Financial Management, and Human Resources.

The enterprise model was developed by a group of experts, and validated by different architects. It is based on the different architectural deliverables, and reflects the current state of the organization, which is supported by a real set of IT components, such as a CRM, an ESB, an ERP, in-house applications, and several infrastructure deployed in the virtualized platform of our EA laboratory.

For the visualization of typical questions over EMs we designed five analytical scenarios that fall into three main categories that correspond to common concerns of architects, and are inspired on the properties described in Sec. 2 (see Table 1):

It is also noteworthy that the visualizations generated display patterns that are difficult to envision by other means (i.e. queries or views).

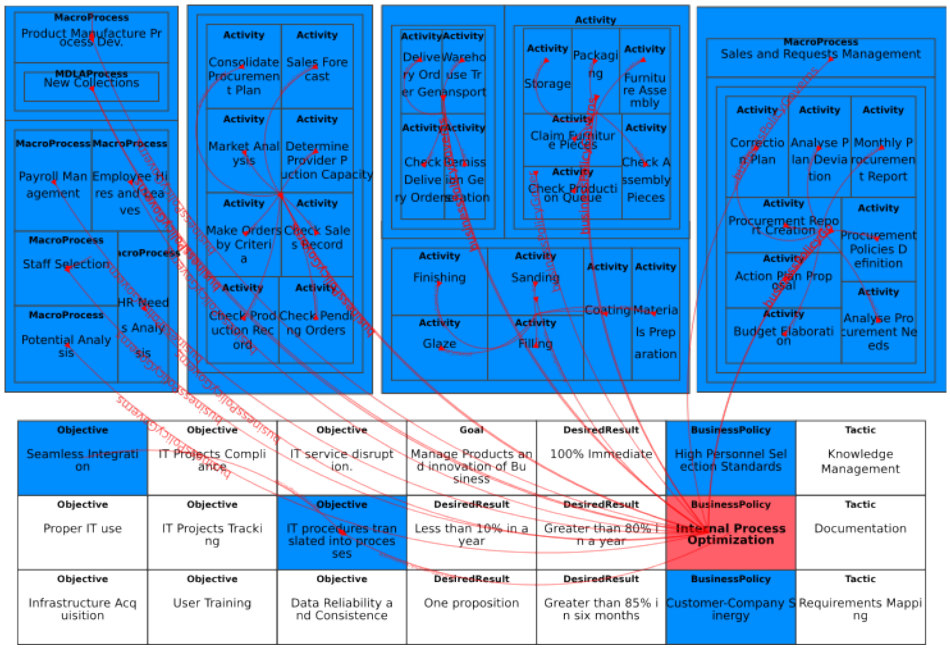


Fig. 3. Very small fragment of a Treemap, describing the impact of the change of a Business Policy on the Motivation Model, affecting several Process Model elements, such as Macro Processes and Activities.

Table 1. Analytical Scenarios for EA.

Category	Topic	Question	Strategy	Technique
Business-IT	Motivation Realization	*Which goals/objectives are realized by IT?*	Find paths between Motivation and IT elements	Sunburst (Fig. 2(a))
Alignment	Process Support	*Which processes are not supported by IT?*	Cross-reference processes and applications	Radial Graph (Fig. 2(b))
EA Management	Domain Connectivity	*What is the dependency between domains/layers?*	Display inter-domain relations	Force Graph (Fig. 1(a))
	Impact Analysis	*What is the impact of a change in the model?*	Find all the shortest paths from a given element	Treemap (Fig. 3)
Anomalies	Structural Holes	*Are there isolated elements or groups?*	Discover all the isolated sub- graphs of the model	Force Graph (Fig. 1(b))

5 Evaluation

Many authors [21, 22, 23] have studied the cognitive power of the different visual attributes and their capacity to display different (e.g. quantitative, ordered, nominal) kinds of information, and how we can combine these attributes for better results.

Based on the characterization of Bertin [6], and in order to measure the expressive potential of a visualization technique, we propose a method for evaluating its global effectiveness:

Table 2. Usage of the visual variables by each technique. Convention: **Available**, **Used to differentiate domains**, **Used by the technique**, **Does not apply**.

Technique	Color	Size	Position	Shape	Transp.	Ang.pos.	Orient.	Texture
Force Graph	Used to differentiate domains	Available	Used by the technique	Used by the technique	Available	Does not apply	Does not apply	Available
Radial Graph	Used to differentiate domains	Available	Used by the technique	Used by the technique	Available	Used to differentiate domains	Does not apply	Available
Sunburst	Available	Used by the technique	Used by the technique	Used by the technique	Available	Used to differentiate domains	Does not apply	Does not apply
Treemap	Available	Used by the technique	Used to differentiate domains	Used by the technique	Used by the technique	Does not apply	Does not apply	Does not apply

$$\varepsilon_v = \Sigma(A_v * w_A) \quad (1)$$

where A_v is the availability of a visual variable for a given visualization (see Table 2), and w_A is the weight -expressive power- of each variable. On the other hand, we defined in previous work [5] a set of evaluation criteria for assessing the support for EA Visual Analysis offered by some EA management tools and general purpose visualization tools, in order to assess the gap between what is offered by the former and what is possible by the latter. We incorporated a metric for each EA requirement, ending with four quantitative and four qualitative criteria (see Table 3).

Table 3. Criteria for the evaluation of each EA Requirement and their associated metrics.

ID	Measure	Type	Units	Scale
C1	Domains differentiated/Total Domains	Quantitative	% differentiation	0-1
C2	Display of inter-domain relations	Qualitative	Scale	0-5
C3	Visual Effectiveness (ε_v) of the visualization	Quantitative	Effectiveness	0-3
C4	Easiness of selecting arbitrary elements	Qualitative	Scale	0-5
C5	Number of levels of detail supported	Qualitative	Level of detail	0-4
C6	Perception of similarity and grouping	Quantitative	Gestalt principles	0-8
C7	Level of differentiation of relations	Qualitative	Scale	0-5
C8	Has the visualization significantly changed when applying different models?	Quantitative	Yes/No	0-1

5.1 Results

Evaluation results are displayed in Fig. 4 by using a Parallel Coordinates visualization, which provides a visual summary of the effectiveness of each technique.

We can see that graph visualizations point to *convergent* analysis, as their strong points are when searching concrete elements and groups, displaying the dependency between domains. At the same time, hierarchical techniques offer an interactive way of exploring the model, facilitating abstraction of elements and offering *divergent* analysis, where it is more important to characterize the architecture as a whole.

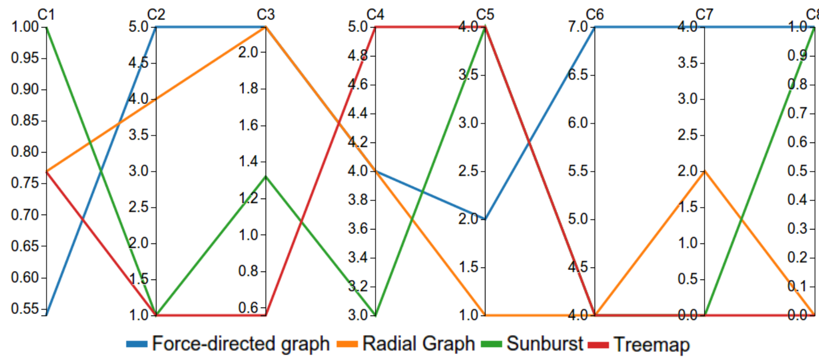


Fig. 4. Evaluation results.

6 DISCUSSION

A previous step before proposing new domain-specific visualization techniques is the assessment of existing ones, in order to derive which aspects of each technique prove useful in our context.

When addressing the visual analysis of complex systems by displaying the overall structure, the analyzed visualizations employ different methods to show the 'big picture' of these complex systems. The visual patterns that they display point to structural assessments of this information. These kind of conclusions are fundamental for the analysis and evolution of the architecture, as well as a way to leverage the complexity of EA Management.

However, this overview analysis is mediated by the concerns of the analyst. This trade-off is reflected on the EA requirements, where some emphasize convergent analyses, i.e. the discovery of interesting elements, while others on Divergent analyses, which are a characterization of the architecture as a whole.

This study opens the door to more specialized research on EA visualization and tooling, as well as the proposal of innovative visualization techniques for the field. Other interesting research opportunities, such as the correspondence between visual and architectural patterns need to be further developed, and a Visual Analysis tool for EM has some requirements in the overall process that we didn't touch, e.g. Provenance (persisting previous hypotheses/analyses and annotations), Traceability of the analysis, or even interactive visualizations specific for EA, as well as an architecture to support all of these features, integrated with current EA modeling tools.

References

1. Keim, D., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melanon, G.: Visual analytics: Definition, process, and challenges. In: Information Visualization. Volume 4950 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2008) 154–175
2. Buckl, S., Matthes, F., Schweda, C.M.: Classifying enterprise architecture analysis approaches. In: Enterprise Interoperability. Volume 38 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2009) 66–79

3. Kohlhammer, J., May, T., Hoffmann, M.: Visual analytics for the strategic decision making process. In: GeoSpatial Visual Analytics. NATO Science for Peace and Security Series C: Environmental Security. Springer Netherlands (2009) 299–310
4. US Department of Defense: DoD Architecture Framework Version 2.0: Volume 1. Technical report, Pentagon, Washington DC (May 2009)
5. Naranjo, D., Sánchez, M., Villalobos, J.: Visual analysis of enterprise models. In: Workshops Proceedings of the 16th IEEE International Enterprise Distributed Object Computing Conference, EDOCW 2012, IEEE Computer Society (September 2012)
6. Bertin, J.: Graphics and graphic information-processing. Walter de Gruyter, Berlin (1981)
7. Stirna, J., Persson, A.: Evolution of an enterprise modeling method next generation improvements of ekd. In: The Practice of Enterprise Modeling. Volume 134 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2012) 1–15
8. Zikra, I.: Implementing the unifying meta-model for enterprise modeling and model-driven development: An experience report. In: The Practice of Enterprise Modeling. Volume 134 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2012) 172–187
9. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. Software & Systems Modeling (2012) 1–22
10. Braun, C., Winter, R.: A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In: Enterprise Modelling and Information Systems Architectures, Proc. of the Workshop in, EMISA (2005) 64–79
11. Horkoff, J., Yu, E., Ghose, A.: Interactive goal model analysis applied systematic procedures versus ad hoc analysis. In: The Practice of Enterprise Modeling. Volume 68 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2010) 130–144
12. Cilliers, P.: Boundaries, hierarchies and networks in complex systems. International Journal of Innovation Management **05**(02) (2001) 135–147
13. Wesenberg, H.: Enterprise modeling in an agile world. In Johannesson, P., Krogstie, J., Opdahl, A., eds.: The Practice of Enterprise Modeling. Volume 92 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2011) 126–130
14. Jorgensen, H.D.: Enterprise modeling what we have learned, and what we have not. In Persson, A., Stirna, J., eds.: The Practice of Enterprise Modeling. Volume 39 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2009) 3–7
15. Binz, T., Leymann, F., Nowak, A., Schumm, D.: Improving the Manageability of Enterprise Topologies Through Segmentation, Graph Transformation, and Analysis Strategies. In: Proceedings of EDOC 2012, IEEE Computer Society (September 2012)
16. Hug, C., Front, A., Rieu, D.: Promise: A process metamodelling method for information systems engineering. In: Evaluation of Novel Approaches to Software Engineering. Volume 230 of Communications in Computer and Information Science. Springer Berlin Heidelberg (2011) 89–105
17. Lagerström, R., Franke, U., Johnson, P., Ullberg, J.: A method for creating entreprise architecture metamodels applied to systems modifiability. IJCSA **6**(5) (2009) 89–120
18. van Wijk, J.: The value of visualization. In: Visualization, 2005. VIS 05. IEEE. (oct. 2005) 79 – 86
19. Bertin, J.: Semiology of graphics. University of Wisconsin Press (1983)
20. Mackinlay, J.: Automating the design of graphical presentations of relational information. ACM Trans. Graph. **5**(2) (April 1986) 110–141
21. Kandinsky, W.: Point and Line to Plane. History of Art Series. Dover Publications (1979)
22. Tufte, E.: Envisioning information. Graphics Press (1990)
23. Ware, C.: Visual Thinking for Design. Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann (2008)

Software Migration Project Cost Estimation using COCOMO II and Enterprise Architecture Modeling

Alexander Hjalmarsson[1], Matus Korman[1]
and Robert Lagerström[1],

[1] Royal Institute of Technology, Osquldas v. 10,
10044 Stockholm, Sweden
alehja@kth.se
{matusk, robertl}@ics.kth.se

Abstract. Large amounts of software are running on what is considered to be legacy platforms. These systems are often business critical and cannot be phased out without a proper replacement. Migration of these legacy applications can be troublesome due to poor documentation and a changing workforce. Estimating the cost of such projects is nontrivial. Expert estimation is the most common method, but the method is heavily relying on the experience, knowledge, and intuition of the estimator. The use of a complementary estimation method can increase the accuracy of the assessment. This paper presents a metamodel that combines enterprise architecture modeling concepts with the COCOMO II estimation model. Our study proposes a method combining expert estimation with the metamodel-based approach to increase the estimation accuracy. The combination was tested with four project samples at a large Nordic manufacturing company, which resulted in a mean magnitude of relative error of 10%.

Keywords: Software migration estimation, Enterprise architecture modeling, Software engineering, Expert estimations.

1 Introduction

When having a software product portfolio spanning over hundreds of legacy systems, maintenance becomes a problem. Expensive hardware as well as lack of experienced developers in the environment drives the cost of maintenance each year. These legacy systems are often crucial to the businesses and cannot be phased out without proper replacement [1].

Even though new computing technologies have emerged on the market, a considerable amount of software still runs on legacy systems. It is estimated that around 200 billion lines of Cobol code are running in live operation and that 75% of the world's business data are processed in Cobol [2,3]. With an estimated shortfall in Cobol developers in the 2015-2020 timeframe, as the older generation leaves the workforce, it is imminent that migration from the legacy mainframes becomes a priority for many organizations [3]. There are many difficulties involved in the migration process. Understanding the design and functionality of the legacy systems may be troublesome due to the fact that many of these systems have poor, if any, documentation. Because of this, interaction from a system expert is often required [4]. These experts need to analyze the old systems to create accurate requirement specifications regarding technical functionality. This documentation is crucial for the developers and architects involved in the migration process.

Because of the importance of these systems the replacement often needs to suit both new business objectives while maintaining functionality for legacy systems that have not yet been migrated. These factors all come into play when estimating the cost of a migration software project. A case study made by [5] showed that as much as 72% of 145 studied maintenance projects used expert opinion as method for estimating software development costs. Another survey showed that out of 26 studied industrial projects 81% were based on expert estimates [6]. One of the problems with expert estimates is that these can be strongly biased and misled by irrelevant information, which can lead to over-optimism and inaccurate estimations. This often cause project over-runs and may be avoided with an unbiased estimation model [5].

There are claims that a combination of estimates from independent sources, preferably applying different approaches, will on average improve the estimation accuracy. Research has shown that a combination of model and expert estimates produces up to 16% better than the best single decision [7].

This paper proposes a metamodel based on the ArchiMate modeling language [8,9] combined with the COnstructive COst MOdel II (COCOMO II) [10]. In our case study we found that the estimation capabilities of the proposed metamodel together with expert estimation is acceptable. Therefore, we suggest that the metamodel should be used as a complement to expert estimations in order to provide more accurate assessment of migration projects.

The remainder of this paper is structured as follows: Section 2 describes COCOMO II; Section 3 presents enterprise architecture modeling; Section 4 describes the proposed estimation metamodel; Section 5 presents the case study; and Section 6 concludes the paper.

2 COCOMO II

COCOMO, COnstructive COst MOdel, was in its first version released in the early 1980's. It became one of the most frequently used and most appreciated software cost estimation models of that time. Since then, development and modifications of COCOMO has been performed several times to keep the model up to date with the continuously evolving software development trends. The latest version of COCOMO, called COCOMO II, had its estimation capabilities calibrated in the year 2000 with the help of information from 161 project data points and eight experts [10].

In the COCOMO II model, the final cost in person-months (*PM*s) is calculated as:

$$PM = A * Size^{E} \prod_{i=1}^{n} EM_i \; ; \qquad (1)$$

Where A is a calibration constant that depends on the organizations practices and the type of software migrated. E is a constant used to scale projects depending on size. E

reflects the fact that cost and size are not perfectly linear. *EMs* are so called Effort Multipliers.

2.1 Scale Factors

The constant E is derived using the following formula:

$$E = B + 0.01 \sum_{j=1}^{n} SF_j \ ; \qquad (2)$$

Where *SF*s are five scale factors. These are precedentedness, development flexibility, architecture/risk resolution, team cohesion, and process maturity. Boehm et al. [10] selected these five factors that describe economies or diseconomies of scale in software projects. This is based on the theory that depending on these variables, the productivity in the project can increase or decrease as it gets larger.

2.2 Effort Multipliers

COCOMO II [10] contains seventeen so called Effort Multipliers (*EM*). These cost drivers affect the software development project in either positive or negative way. The *EM*s are divided into four categories: product factors, platform factors, project factors and personnel factors. They each have a different set of factors within their respective category. The product factors are; required software reliability (RELY), database size (DATA), product complexity (CPLX), developed for reusability (RUSE), and documentation match to life-cycle needs (DOCU). The platform factors are; execution time constraint (TIME), main storage constraint (STOR), and platform volatility (PVOL). The personnel factors are; analyst capability (ACAP), programmer capability (PCAP), personnel continuity (PCON), applications experience (APEX), platform experience (PLEX), and language and tool experience (LTEX). The project factors are; use of software tools (TOOL), multisite development (SITE), and requirement development schedule (SCED).

3 Enterprise Architecture Modeling

Enterprise architecture analysis has emerged during the last decade as an approach to assess different types of non-functional requirements in a company. Migration projects are common projects in an enterprise today, thus including cost estimation for these projects with enterprise architecture could appeal to architects. Research in the area has proposed a framework of enterprise architecture analysis using ArchiMate and a computational model "The Predictive, Probabilistic, Architecture Modeling Framework" (P^2AMF) [11]. P^2AMF can enable calculation on entities in for instance an ArchiMate model. This framework will be the basis of the metamodel used to enable COCOMO II estimations.

3.1 ArchiMate

ArchiMate is a modeling language intentionally resembling the Unified Modeling Language (UML) [8,9]. The reason of using ArchiMate as the basis of graphical notation framework is due to its generality, making it possible to extend existing metamodels with change project estimation as well as providing a solid ground for future adaptions.

The ArchiMate language consists of three core concepts, namely the active structure, passive structure, and behavioral elements. The passive structure elements are elements on which behavior is performed while the active structure is the entity performing the behavior. These concepts are then specialized in each of the three layers specified in ArchiMate [8,9]; the business layer that offers products and services to external customers, the application layer that supports the business layer with application services which are realized by software applications, and the technology layer containing the infrastructure services needed to run applications, realized by computers, communication hardware and system software. The classes found in ArchiMate is for instance; business process, software application, and infrastructure service.

3.2 The Predictive, Probabilistic, Architecture Modeling Framework (P2AMF)

The Predictive, Probabilistic Architecture Modeling Framework (P^2AMF) is a generic framework for system analysis [11] based on OCL and used in order to describe expressions in the Unified Modeling Language (UML). P^2AMF is fully implemented in the Enterprise Architecture Analysis Tool (EAAT) [12,13]. The framework has been utilized to calculate the formulas in the COCOMO II model accordingly.

The end result of this would be that the algorithmic formula used in the model would have a probability distribution indicating the probable cost range of the project rather than a specific mean value. This, in combination with the ArchiMate language, provides a strong basis for using the P^2AMF for cost estimation. However, due to space limitations we have not made use of the probability distributions in this paper.

4 The Proposed Estimation Metamodel

This section presents the metamodel for migration project cost estimation. The metamodel is heavily influenced by COCOMO II [10] and the previously proposed metamodel by [14] and [15]. The most relevant parts of COCOMO II are included in the metamodel proposed while Lagerström's previous work has served as an influence and guideline for the metamodel construction and is thus left out of this description.

ArchiMate is in general used to describe the layers in enterprises' architectures and to for example show what applications are used in what business processes. ArchiMate is tailored for describing as-is and to-be scenarios [8,9]. In this paper we present a specialization of ArchiMate that handles project specific factors. The project specific metamodel elements are then combined with the regular ArchiMate metamodel classes to calculate the migration cost estimate.

The combined metamodel contains the seventeen effort multipliers as well as the five scale factors in a combination. The metamodel differentiates between the three ArchiMate layers as well as the new project specific metamodel classes (see Fig. 1): the business layer (in red) contains the class "*Personnel*;" the application layer (in green) contains the classes "*ApplicationComponent*," "*ApplicationFunction*," and "*ApplicationService*;" the infrastructure layer (in yellow) contains the class "*InfrastructureService*;" and the project entities (in blue) are "*SoftwareDevelopmentProcess*," "*SoftwareDevelopmentProject*," "*Activity*," "*Change*," and "*EffortDivisor*."

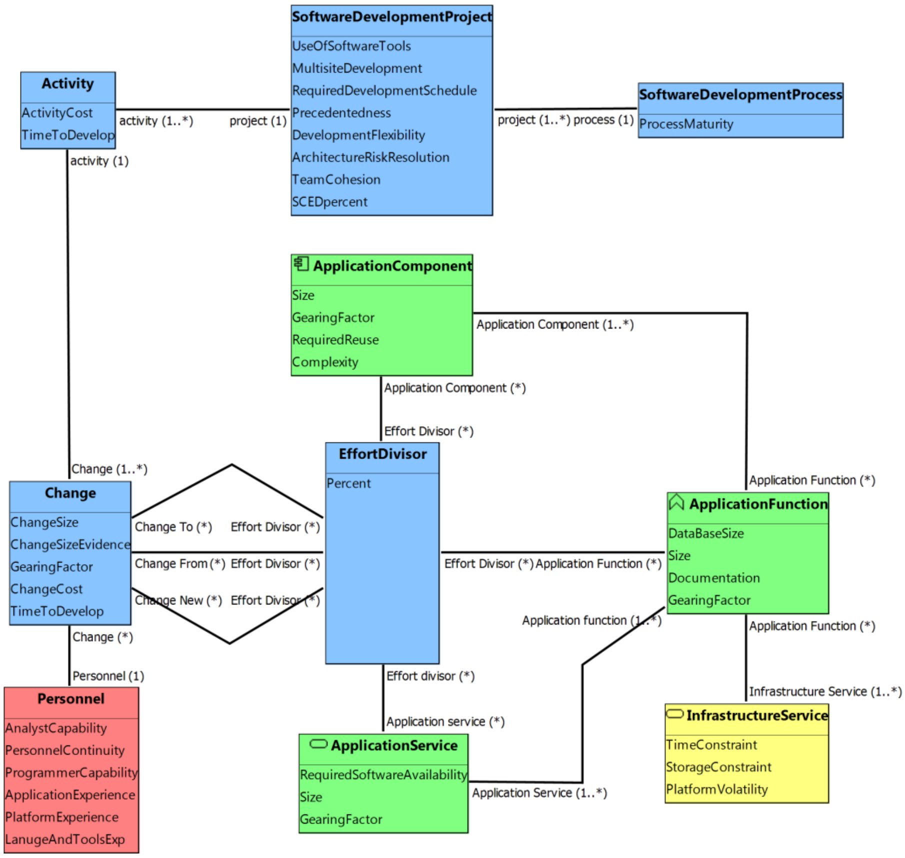


Fig. 1. The proposed metamodel for software migration cost estimation.

5 The Case Study

Our study was conducted at a large Nordic manufacturing company. The data points used in order to validate and calibrate the metamodel are projected as having been closed during the last six months and satisfy the constraint of having > 2000 SLOC produced in the project. The data was collected through interviews with managers, developers, and architects in the projects. Project reports were also used to validate the information elicited and as a source of the project costs (effort in person-hours/man-months). In total we looked at four different migration projects. Due to space limitation we provide some more details regarding Project B (below) before presenting the analysis and results. The complete study can be found described in [16].

5.1 Project B

This project was initiated for the purpose of replacing an old application with a new one running on the company's standardized platform with included support and development agreements. The old application was based on old technology and could not run on modern PC's such as the ones based on the x64 architecture. The software is used to determine variables of the propeller shaft used in vehicles produced by the company. It is only used by the experts in the area and the old application did only run on one PC. Overall, the project was deemed successful. Deviations in the project schedule occurred due to the complexity in the algorithms that were implemented. The project utilized a software development method working iteratively in sprints with demonstrations to customers after each of the sprints. The project had an 18% overrun of the estimated budget due to new requirements added to the migrated version of the software, which increased the scope of the project. The size of Project B was straight forward as it only consisted of migrating one application. The project resulted in 5,500 SLOC developed with the .NET platform. Table 1 presents the data for Project B.

Table 1. Data for Project B.

Scale Factors	**Rating**	**Effort Multipliers**	**Rating**
PREC	NOMINAL	RELY	LOW
FLEX	LOW	DATA	VERY HIGH
RESL	HIGH	CPLX	VERY HIGH
TEAM	VERY HIGH	RUSE	VERY HIGH
PMAT	HIGH	DOCU	VERY HIGH
		TIME	NOMINAL
Factors:		STOR	NOMINAL
	$E = 1.0582$	PVOL	NOMINAL
	$EM_{project} = 0.774$	ACAP	VERY HIGH

$EM_{product} = 2.23$	PCAP	VERY HIGH
$EM_{platform} = 1$	PCON	VERY HIGH
$EM_{personell} = 0.32$ $EM_{tot} = 0.55$	APEX	HIGH
Actual	PLEX	HIGH
$PM_{act} = 17.42$	LTEX	HIGH
$SLOC = 5{,}500$	TOOL	HIGH
	SITE	VERY HIGH
	SCED	NOMINAL

5.2 Validation Method

The validation consists of measuring the accuracy of the model. The accuracy is measured by using the Mean Magnitude of the Relative Error (*MMRE*) and the Magnitude of the Relevant Error (*MRE*) [17].

$$MMRE = \frac{1}{n}\sum_{i=0}^{n} MRE_i \ ; \qquad (3)$$

$$MRE = \frac{|E - \hat{E}|}{E} \ ; \qquad (4)$$

Where E is the actual result and $\hat{E}$ is the estimate.

A model has an acceptable accuracy level if 75% of the projects' estimations are higher or equal to 75% [17]. This is called the prediction quality (*PRED*) and has been used frequently when comparing models and methods within the area of software estimation [14,18]. The prediction quality formula (formula 5) where n is the complete set of projects and k is the amount of projects that have greater or equal accuracy as q.

$$PRED(q) = \frac{k}{n}. \qquad (5)$$

An acceptable accuracy level for a model can be denoted *PRED*(0.25) = 0.75, meaning that 75% of the projects shall be within 25% of the actual result.

5.3 Accuracy

Even before calibration the model conforms rather well to the data gathered. The two largest projects, Project A and C are within the predictive quality margin of 25% (16% and 4%). Project B is not estimated accurately and has a *MRE* of 44%. The model underestimates the effort needed for the project which partly may be because of the additional effort needed due to the problems found in the old application that was migrated.

Compared to the expert estimates the model produces competitive estimates. In the table the mean relevant error has been computed with four different measures. These are the model and expert estimates as well as two combinations of them. The two combinations are the result of the optimal combination between model and expert estimates for the specific purpose. Optimal predictive quality (*Opt. pred*) ensures that all projects are within 25% of the real effort outcome. The optimal mean relevant error (*Opt. MRE*) uses the combination that gives the lowest average *MRE* for the projects.

Table 2. Results before calibration.

Measured in hours				**MRE**			
Project	**Model**	**Estimate**	**Real**	**Model**	**Expert**	**Opt. pred**	**Opt. MRE**
A	14794	9700	12700	16%	24%	15%	0%
B	1494	2140	2648	44%	19%	25%	34%
C	7748	6500	8060	4%	19%	16%	10%
D	1315	918	1209	9%	24%	17%	6%
			Mean MRE	**18%**	**22%**	**18%**	**12%**

Opt. pred is using 24% model and 76% expert. *Opt. MRE* is using 59% model and 41% expert. Table 2 shows that heading for the optimal predictive quality in the model would lower the mean magnitude of relevant error, while the optimal *MRE* achieves a very good mean magnitude of relevant error. From the result it also can be seen that by combining the expert judgments with the model both increases the predictive quality as well as the *MMRE*. This is in line with previous research [7].

5.4 Calibration

Calibrating COCOMO II with organizational specific data typically results in better estimates [10]. One way of calibrating COCOMO II to existing project data is by using the multiplicative constant *A* (see [10,16] for the exact calibration equations). The local calibration usually improves the prediction accuracy due to the use of subjective factors in the model. Further, the lifecycle activities in the projects covered by COCOMO II may differ from the ones in the particular organization [10].

The calibration resulted in an increased value of the multiplicative constant *A* used in the effort estimation from 2.94 to 3.23. As can be seen in

Table 3, the calibration yields a lower *MMRE* for the model estimation. This is because the calibration is minimizing the sum of squared residuals in log space rather than the *MRE*. *Opt. pred* was achieved using 31% model and 69% expert, while *Opt. MRE* was achieved by using 46% model and 54% expert.

Table 3. Results after calibration.

Measured in hours				MRE			
Project	**Model**	**Estimate**	**Real**	**Model**	**Expert**	**Opt. pred**	**Opt. MRE**
A	16250	9700	12700	28%	24%	8%	0%
B	1640	2140	2648	38%	19%	25%	28%
C	8510	6500	8060	6%	19%	12%	8%
D	1445	918	1209	19%	24%	11%	4%
			Mean MRE	**23%**	**22%**	**14%**	**10%**

6 Discussion and Conclusions

The results of the case study validates that the combination of COCOMO II with the ArchiMate modeling language works as predicted and that the model estimates are on par with the managers at the case study company. The combination between model and expert estimates performs far better than single selections of model or expert estimations. Without calibration, optimal *MMRE* strategy achieved a *MMRE* of 12% with *PRED*(.25) = 75%. When adding the constraint of *PRED*(.25) = 100%, the *MMRE* rose to 18% which was slightly better than the expert estimates (22%) and on par with the model (18%).

One question that might arise is: Why combining EA and COCOMO II and not only use COCOMO II? As we see it, there is a strength of using EA models as input together with project specific data. ArchiMate as-is and to-be models that already contain information can easily be re-used for every software migration project and the project specific information is the only part that needs to be up-dated. Also, many companies today struggle with maintaining their EA models since new projects alter the as-is architecture continuously. With this approach one could align the as-is and to-be models with all the on-going projects and automatically update the models once the projects are finished. Also, for architects it provides an instrument to work with when creating to-be models and assessing if future scenarios are appropriate for change projects.

In this paper we have presented a metamodel for software migration project estimation. The metamodel was constructed based on metrics from COCOMO II, modeling elements from ArchiMate, and an analysis engine of P^2AMF. The metamodel was tested in four cases at a large Nordic manufacturing firm. Our results show that the metamodel itself performs rather well but as COCOMO II suggests it performs even better when calibrated with data from the company under analysis. In software cost estimation research it has been shown that model estimates and expert estimates complement each other in a good way and that the combination often outperforms the two approaches. This was also the case in our study. Therefore, we conclude that our proposed metamodel is useful, especially after company specific calibration and in combination with expert estimates.

References

1. Bennet, K.: Legacy Systems - Coping with Stress. IEEE Software 12(1), 19--23 (1995)
2. Datamonitor: COBOL - Continuing to Drive Value in the 21st Century. London: Datamonitor (2008)
3. Barnett, G.: The Future of the Mainframe. London: Ovum (2005)
4. Bisbal, J., et al.:A Survery of Research into Legacy System Migration. Dublin: Trinity College Dublin (1997)
5. Kitchenham, B., Pfleeger, S. L., McColl, B., Eagan, S.: An Empirical Study of Maintenance and Development Estimation Accuracy. Journal of Systems and Software 64(1), 57--77 (2002)
6. Molokken, K., Jørgensen, M.: A Review of Software Surveys on Software Effort Estimation. Empirical Software Engineering, 223—230 (2003)
7. Blattberg, R. C., Hoch, S. J.: Database Models and Managerial Intuition: 50% model + 50% manager. Management Science 36(8), 887—899 (1990)
8. The Open Group: ArchiMate 1.0 Specification. http://pubs.opengroup.org/architecture/archimate-doc/ts_archimate/
9. Lankhorst, M.: Enterprise Architecture at Work - Modelling, Communication and Analysis. Springer. Third Edition (2013)
10. Boehm, B., et al.: Software Cost Estimation With Cocomo II. New Jersey: Prentice Hall (2000)
11. Johnson, P., Ullberg, J., Buschle, M., Franke, U., Shahzad, K.: P^2AMF - Predictive, Probabilistic Architecture Modeling Framework. In: Proc. of International IFIP Working Conference on Enterprise Interoperability Information, Services and Processes for the Interoperable Economy and Society (2013)
12. The Enterprise Architecture Analysis Tool, www.ics.kth.se/eaat
13. Buschle, M., Ullberg, J., Franke, U., Lagerström, R., Sommestad, T.: A Tool for Enterprise Architecture Analysis using the PRM Formalism. Information Systems Evolution, 108--121 (2011)
14. Lagerström, R., Johnson, P., Höök, D.: Architecture Analysis of Enterprise Systems Modifiability: Models, Analysis, and Validation. Journal of Systems and Software 83.8, 1387--1403 (2010)
15. Österlind, M., Lagerström, R., Rosell, P.: Assessing Modifiability in Application Services using Enterprise Architecture Models - A Case Study. In: Proc. of Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation. Springer Berlin Heidelberg (2012)
16. Hjalmarsson, A.:Software Development Cost Estimation using COCOMO II based Meta Model. Master Thesis. The Royal Institute of Technology, Stockholm, Sweden. XR-EE-ICS 2013:005 (2013)
17. Conte, S. D., Dunsmore, H. E., Shen, V. Y.: Software Effort Estimation and Productivity. Advances in Computers. Academic Press, Inc, 1--59 (1985)
18. Kemerer, C. F.: An Empirical Validation of Software Cost Estimation Models. Communications of the ACM 30.5, 416--429 (1987)

Process Model Driven Requirements Engineering

Dag Rojahn Karlsen[1], Helle Frisak Sem[1], Steinar Carlsen[1]

[1] Computas AS, Lysaker Torg 45, N-1327 Lysaker, Norway
{dag.rojahn.karlsen, helle.frisak.sem, steinar.carlsen}@computas.com

Abstract. This paper demonstrates an approach to requirements engineering with process models as the hub for a family of consistent sub-models. The process model taken as a starting point gives focus, and the different requirement perspectives are served through the enrichment of the process model in several dimensions. Different views into the model family give the well-known specification models; information models, exchange models, service models, etc. The family of related sub models are refined iteratively to describe the need for information (NFI), the need for services (NFS), a shared common information model (CIM) and refined domain information models (DIM). We call this family of interrelated sub models of information systems *knowledge models*, since they capture both the structure and the behaviour of the subject matter domain. Working in an integrated way with several models serves to enhance consistency and to test the quality of each particular model.

Keywords: Process models, knowledge models, requirements engineering, enterprise models, information models

1 Background

This paper presents an enterprise modelling approach to requirements engineering discovered in our work in a project for a large Norwegian public sector organisation - LNOP. LNOP manages a substantial part of the national budget in Norway and administers a diverse set of schemes under precise regulations.

LNOP was established as a merger between two existing organisations, each with a long history. The local authorities and central government cooperate through 456 LNOP offices in municipalities and city boroughs. LNOP employs around 19,000 people. Of these around 14,000 are employed by the central government, and around 5,000 are employed by the local authorities. In addition to the local LNOP offices there are more than one hundred special units that perform centralised duties that would not be appropriate for front line local LNOP offices to perform.

LNOP has a large portfolio of legacy systems, some of them originating as far back as 1976. In order to move to an efficient IT situation characterised by automation and self-service, LNOP launched a large, high-profile IT modernisation programme in 2012, with projects involving several contractors, employing up to 300 internal and external staff members.

2 Overall Process

The project utilizes TOGAF [1] as its enterprise architecture framework. To produce the different architectural artefacts in the specification process, the programme has chosen the modelling languages BPMN 2.0 [10] for process modelling and UML 2.0 [11] for information and other modelling. This paper describes how we have combined process modelling and information modelling in a way that has been fruitful in the specification process.

We know that there are enterprise architecture frameworks in the market, such as ArchiMate [13], with the aim to describe the different layers of architecture in an integrated and holistic way. We cannot see, however, that these cover the integration between process and information models with a purpose as described below. We are aware, however, that this may be achieved using other languages and styles, but the aim of this paper is to describe what we did in an actual project.

To maintain consistency and control in a project with several contractors developing a number of applications, and a large portfolio of legacy systems, the project utilizes a service-oriented [2] [3] and model driven approach, specifying systems and their responsibilities and interrelations within a variety of architectural styles and models. For an example of service-oriented modelling methodology, see [4].

An ambitious goal of the project is to develop a Common Information Model, which is a common model for the exchange of information between applications, old and new, creating an information exchange language for LNOP. The motivation behind this is to minimize the dependencies between integrated applications, following an accommodated version of the pattern called Canonical Data Model [5].

2.1 The Overall Process for the Development Project

The goal of the project is to develop a number of new applications based on a common methodology. The agile method scrum [6] [7] is used as the development method. The business needs are specified on two levels, both administered in Atlassian JIRA [8] and documented in Atlassian Confluence [9]. The top level is called epic. Each epic is further specified by a set of user stories. Prioritising and planning is performed by means of the epics and the user stories.

Suitable sets of user stories are identified to belong to the same application, and the specification and modelling necessary for development starts, resulting in a Solution Architecture under the responsibility of the LNOP, and a Solution Specification under the responsibility of the contractor responsible for application development. Contracts for the development of a small set of user stories are then entered into and developed through three-week development sprints.

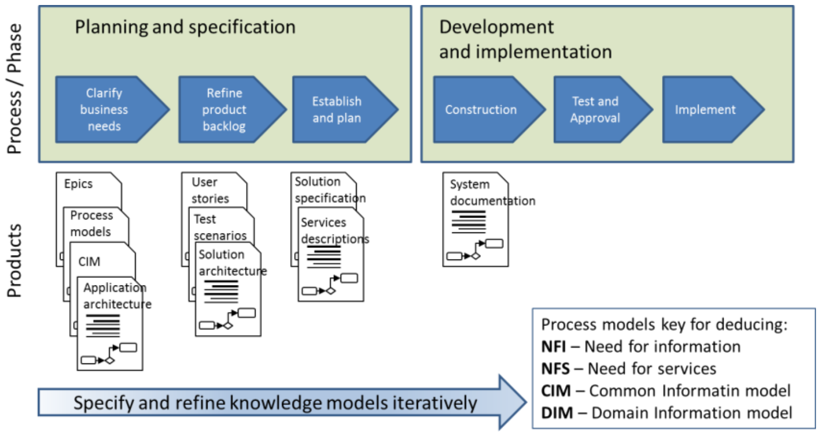


Fig. 1. The overall development process with phases and documentation products.

The paper focuses mainly on the first three steps, the planning and specification phase and how to work with some of the models involved.

2.2 Definitions

NFI – Need for Information
The information needed by business processes at a conceptual level as identified by process modellers.

NFS – Need for Services
The services needed by a process/activity. The service is required to provide information as identified by one or more needs (NFIs).

CIM – Common Information Model
The common information model specifies the common language used in the data exchange between all applications. Service information models are based on the CIM.

DIM – Domain Information Model
The information model is designed to support a given application domain. Parts of the model will cover information to communicate with other application domains – accordingly DIM and CIM are related.

2.3 Planning and Specification – The Initial Way

The project is organized as business requirement teams, process modelling, information modelling and solution specification teams. The requirement teams are responsible for identifying business needs, specified by epics and user stories.

The process modellers identify and model the business process models, and based on business needs they identify corresponding information needs.

The information modelling teams are of two kinds: One to develop the common information model and corresponding service information models, and another to develop the domain information models.

The solution specification teams are also of two kinds: One to develop the solution architectures and another to develop the solution specifications.

LNOP as client is responsible for all teams except for the domain information modelling teams and the solution specifications teams that are under the responsibility of the contractors.

The common information model team is continuously developing the common information model, and provides the solution specification team with a dedicated representative. Architects on the client side and on the contractor side are both engaged in the specification phase, presenting their work to the client's business experts for the field in question for adjustment and approval.

Initially, each model and architecture product was made separately, often with different people and not enough communication between the teams working on different models. Much time has been spent in order to align the different models, not to speak of the time wasted in the establishment and planning step, or even the construction step, because the models were not consistent with each other. Working in an integrated way with more than one model serves both to enhance consistency and to test the quality of each model.

3 The new approach

Getting an overview of all the dependent activities and their relations has been one of the biggest challenges in the project.

The approach so far has been to look at different dimensions of the project more or less independently, resulting in a fragmented world of documents and models. In addition, the size of the project has made it extremely difficult to get the necessary overview of how all designed artefacts are linked and related.

To meet these challenges, we have developed a holistic approach that integrates models from different domains into what we like to call *knowledge models*.

A *knowledge model* in our terminology must include the process dimension – it must describe what is going on (processes and tasks). In addition the processes must link to other dimensions, such as information, organisation and roles, competence and skills, products and services, etc. depending on the purpose of the model.

The main thought is that process models are key – they represent the glue that directly or indirectly ties the different dimensions together, enabling us to build holistic knowledge models.

The approach has been developed in an information systems development project, a project within the larger modernisation programme, in a team covering roles from LNOP and contractor, including technical and business expert members, as well as process and information modellers. The task of the team has been to give necessary specifications for the development of a particular application, already identified in earlier architectural work, provide the necessary models, in particular information models, and identify system context and dependencies to other applications.

The outcome has been a holistic model that includes:

- **Conceptual views** of the processes (business level) identifying the need for information (NFI) with links to a catalogue of concepts and terms
- **Logical views** of the processes identifying the need for services (NFS), and for data exchange between processes/applications as a first step towards aommon Information Model (CIM), and for information structures to support the domain applications
- **Logical models** that combine process and information constructs in the same diagrams to detail the data exchange models and application domain models

3.1 The Conceptual Models

We have developed a way of working to combine business process models (using BPMN) with Need For Information (NFI) and links to the concept catalogue. The technique keeps the process diagram simple by linking to separate diagrams that model the information needs and relate them to the terms. This allows the same information need to be referred several places in the process diagram.

These process models are built by business analysts who know the business processes and have the knowledge to identify the information needed in the processes.

Another important part of the project is to have a concept catalogue where the different terms and concepts are defined, enabling a common language between people from different disciplines.

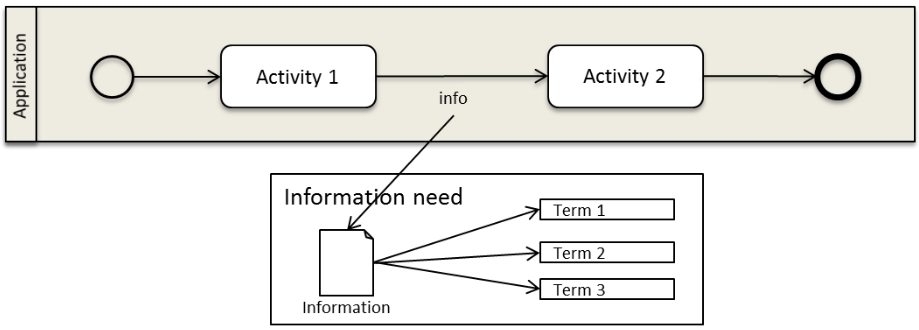


Fig. 2. Conceptual process model identifying Need for Information (NFI) with links to concepts / terms.

3.2 The Logical Models

Below we summarise properties regarding the various logical models involved in the approach.

3.2.1 Need for Services

The project is service-oriented and needs-driven. Needs for Services (NFS) must be identified early in the specification process. This is achieved by building process models that identify the communication between applications and add logical entities that represent the NFS'es to the process models.

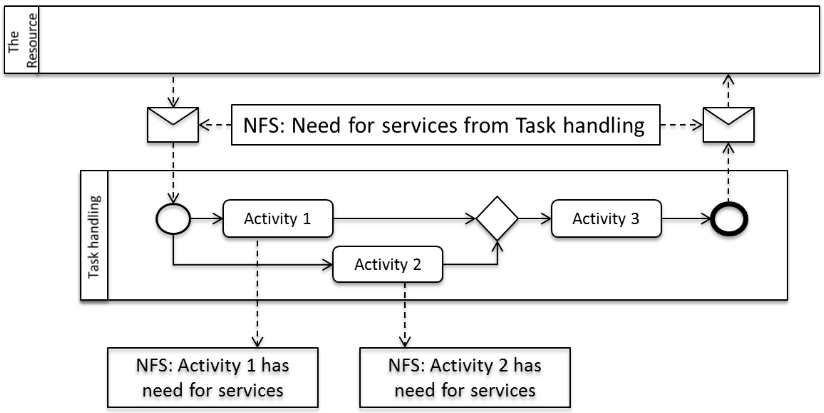


Fig. 3. Need for services illustrated.

The system in focus is the application being specified. The figure illustrates that other LNOP systems may require services from the application being specified, and also that the application itself has a need for services from other service providers at LNOP.

By doing this for all processes in the application being specified, and for other LNOP applications to be developed, the central integration team of the project gets an overview of all the need for services both to be provided by these applications, as well as those needed by the applications themselves.

In this process the central integration team will be able to identify if the same or a similar service may be needed by several of the LNOP applications and then they can harmonize the needs and specify one service realization that supports all similar needs.

3.2.2 Identifying CIM as part of Data Exchange Process Modelling

One of the main targets in the project is to establish a Common Information Model (CIM), requiring a special focus on data exchange, as the modelling language used in the communication between applications is required to be CIM.

The Data exchanged is represented as UML classes, as a first step towards a physical representation. Detailing the attributes is not required at this stage.

In Figure 4 we specify what data are exchanged between the two applications and how they are represented as UML classes. The classes define the protocol used in the communication between the two. As the customer requires that all communication between applications are according to a common language model, the UML classes is supposed to be a part of CIM. Accordingly the definition of the UML classes has to take into account the already existing parts of CIM, if there are any (sub-) models related to the task at hand.

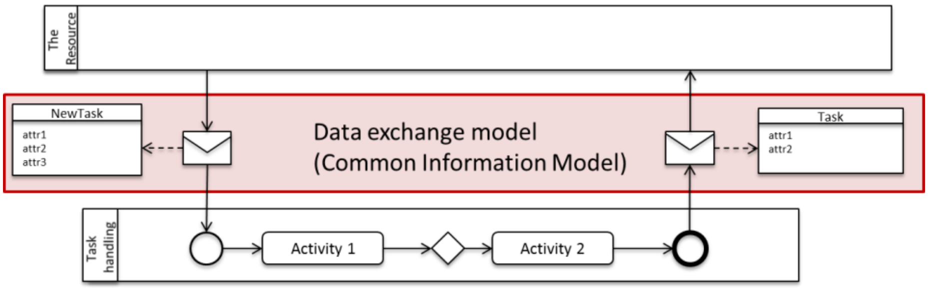


Fig. 4. Data exchange model.

The simple model shown in Figure 4 also illustrates that our approach combines BPMN and UML artefacts in the same model in the same diagram. By doing so, we take the step from process models and information models that live their own lives, to a combined representation of the two that focuses on the role information plays in the process – we take the step from information to knowledge.

3.2.3 Identifying DIM as part of Application Processes Modelling

The Domain Information Model is identified as the need for data objects to support a given application. Our approach is to model the processes that the application is performing and identify their need for information in the same model.

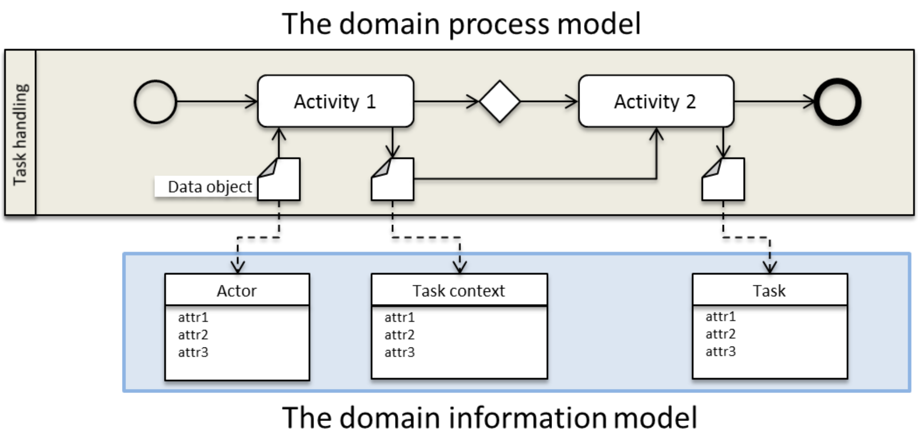


Fig. 5. Domain information model.

This is illustrated in Figure 5, which shows a standard BPMN model that includes data flow, combined with UML classes that detail the data objects in the process model. By combining the two, the information modeller understands the role the information will play in the application and has a much easier task of providing a good result. Again we take the step from information to knowledge.

3.2.4 Knowledge Models and Information Models

The detailed knowledge models are a result of detailing the UML classes with the actual attributes both for the data exchange (CIM) models and the domain specific (DIM) models.

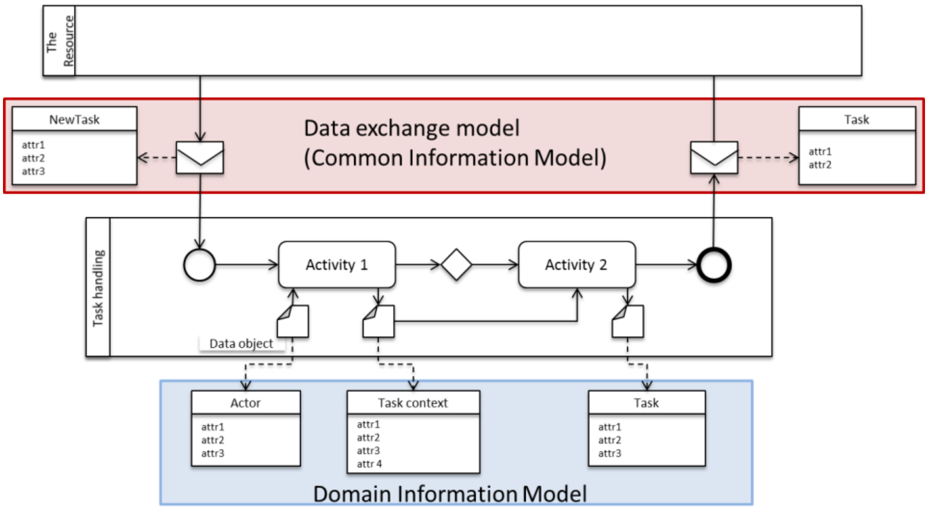


Fig. 6. Process model combined with CIM and DIM models.

Figures 4 and 5 show CIM and DIM as separate models. Figure 6 has combined the two to show how the Common Information Model and the Domain Information Model both relate to the process model. CIM and DIM will always talk about some of the same entities, but it is not required to be in the same way.

There are no restrictions to how DIM represents data, but to avoid complex transformations it is recommended that the CIM and DIM representations of a given entity are as similar as possible. Working with both in the same diagram will make this easier to achieve.

In addition to the combined knowledge models CIM and DIM will have their own diagrams showing them as standard UML class models. But these are primarily derived from the work done in the combined process and information models.

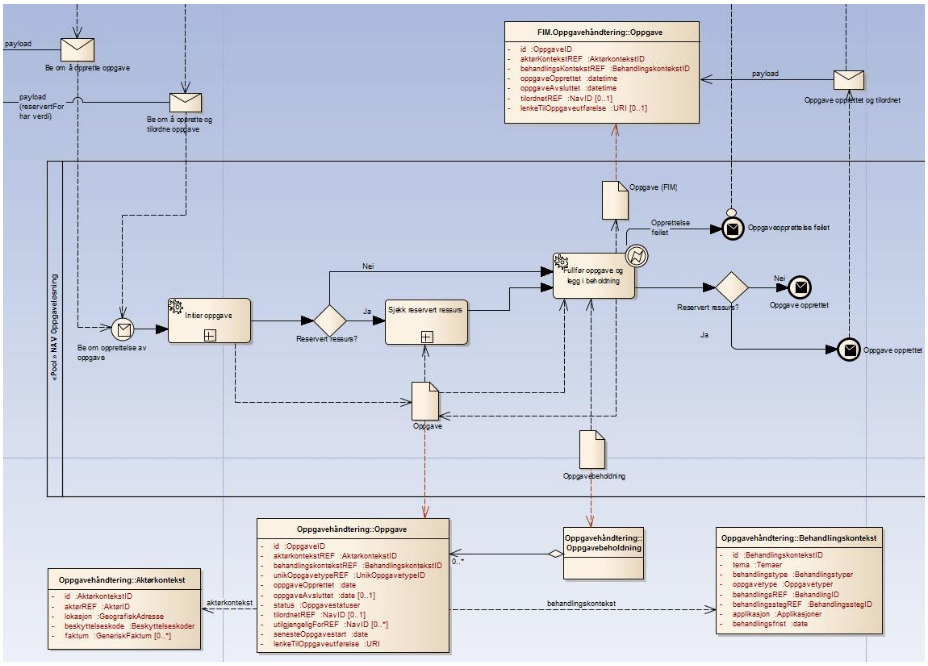


Fig 7. Real-world example of a process model combined with CIM and DIM models.

4 Conclusion

The novelty of the proposed requirements engineering approach as well as the usage experience so far is summarised below.

4.1 Related work / added value

The new approach presented in this paper allows us to build knowledge models that support the development process from conceptual to logical design models, allowing us to combine several dimensions in the same view.

The approach described is similar to the use of holistic and integrated (sub)models in enterprise modelling [12] [13], with two major differences. Firstly, the domain here is not "enterprise modelling" as such, but requirements engineering as part of information system development. Secondly, the approach is based on integrating industry-standard modelling languages (BPMN and UML); modelling languages that were not designed for each other, but which currently are being used in the industry at a large scale.

4.2 Experience with the approach

LNOP has four major streams of IS development, and the approach was crafted in one of these streams and later has spread to two of the other streams. For each of these streams, there is a set of model producers (representing the customer, partly staffed with contractors) and an audience of model consumers (from the contractors). In the originating stream, there were two full-time process modelers, and 7-8 information modelers (dealing with the CIM, for all streams). The model consumers roughly are split into two groups, service developers and application developers. Service development employs 3-4 developers at the customer side (related to CIM) and 4-5 developers at the contractor side (related to DIM and consuming CIM services). The application developer model audience is between 10 and 20 developers per IS development stream. In principle, behind the scenes there is one central information model and one central process model that are adapted to their audiences in tens of "model diagrams" each representing a particular view of the underlying holistic models. These model diagrams are not "uncoordinated sketches", they are all mutually consistent views of the central model. The described modeling approach for requirements engineering in LNOP so far has been considered vital for the purpose of coordinating requirements engineering and development work – within the streams and between the streams.

References

1. The Open Group: The Open Group Group Architecture Framework (TOGAF), version 9.1. Internet (2011), http://pubs.opengroup.org/architecture/togaf9-doc/arch/
2. The Open Group: Service-Oriented Architecture, Architecture for Boundaryless Information Flow (2011), http://www.opengroup.org/subjectareas/soa
3. Erl, T.: Service Oriented Architecture. Concepts, Technology, and Design. Prentice Hall (2005) ISBN 0-13-185858-0
4. Service-Oriented modelling (SOMA), Wikipedia (2013), http://en.wikipedia.org/wiki/Service-oriented_modeling
5. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley (2003) ISBN 0-321-20068-3
6. Sutherland, J.V., Schwaber, K.: Business object design and implementation. In: OOPSLA'95 workshop proceedings, p. 118. The University of Michigan (1995) ISBN 3-540-76096-2
7. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice Hall (2002), ISBN 0-13-067634-9
8. Documentation for JIRA 6.0, Atlassian, 2012, https://www.atlassian.com/software/jira
9. Documentation for CONFLUENCE 5.1, Atlassian, 2012, https://www.atlassian.com/software/confluence
10. Business Process Model & Notation (BPMN), OMG, 2013, http://www.bpmn.org/
11. UML (Unified Modeling Language) Resource Page, OMG, 2013, http://www.uml.org/
12. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises, Springer (2008), ISBN 978-3-540-79416-5
13. The Open Group: Archimate, version 2.0 Internet (2013), http://www.opengroup.org/subjectareas/enterprise/archimate/

Comprehensive Business Process Management through Observation and Navigation

Stefan Schönig, Michael Zeising, and Stefan Jablonski

Applied Computer Science IV, University of Bayreuth, Germany
{stefan.schoenig, michael.zeising,
stefan.jablonski}@uni-bayreuth.de

Abstract. Most real-world business processes involve a combination of both well-defined and previously modelled as well as unforeseen and therefor unmodelled scenarios. The goal of comprehensive process management should be to cover all actually performed processes by accurate models so that they may be fully supported by IT systems. Unmodelled processes can be observed by the *Process Observation* system which generates models reflecting the recorded behaviour. Modelled processes may be of different natures: while so-called "automation" processes involve little human participation and mainly orchestrate services and applications, so-called "knowledge-intensive" processes are based on human expert participation. Both types of models may be enacted by the *Process Navigation* system. This contribution introduces the integration of both systems which leads to an approach for supporting the full range from unmodelled processes to both automation and knowledge-intensive processes as well as the transition from unmodelled to modelled processes.

Keywords: Business processes, workflow, process observation, declarative process modelling, process mining.

1 Introduction

Business process management (BPM) is considered an essential strategy to create and maintain competitive advantage by modelling, controlling and monitoring production and development as well as administrative processes [1, 2]. Many enterprises and organizations adopt a process model-based approach to manage their operations. Ideally, this involves surveying and modelling the as-is processes and designing to-be processes in a way so that they may be supported by BPM technologies. In reality, areas remain in which the as-is process may not be documented in a formal way so that they may not be supported by traditional workflow systems. As a consequence, these areas are then excluded from IT support. However, it is desirable that these unmodelled processes are supported by an IT system as well and that, ideally, this system offers suggestions for modelling these processes. This paper presents an approach for fluently covering both situations: the execution of modelled business processes and the support during unmodelled situations with a reconstruction of the actually performed process. Figure 1 gives an overview over the structure of the approach. The Process Observation (PO) system covers the support of unmodelled processes. Based on collected execution information, best practice patterns are provisioned to users. These patterns are emerging with progressing enactment as the data basis grows. On the other hand, the Process Navigation (PN) system executes modelled processes. We differentiate between so-called "knowledge-intensive" and "automation processes". In Section 2 we will have a closer look at this differentiation. We will justify why the application of different modelling concepts, the declarative modelling approach for knowledge-intensive processes and the imperative modelling approach for automation processes respectively, is suitable. Both types of process models can be executed by the PN system. For supporting both modelled and unmodelled business processes, the two components PN and PO must be integrated. Hence, PO makes use of collected information and generates executable process models. These models can finally be enacted by the PN system.

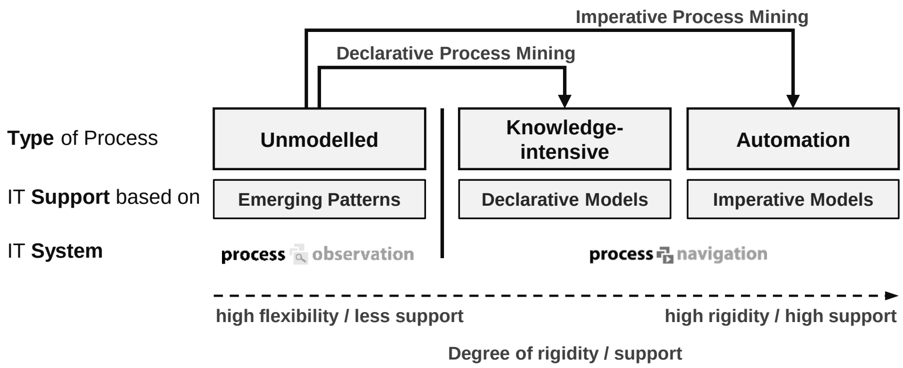


Fig. 1. Different components and conceptual structure of the approach

This paper is structured as follows: Section 2 outlines the basic principles of the execution component for modelled processes. Section 3 guides through the ideas behind the observation component for firstly unmodelled processes. Section 4 shows how the two components are integrated. Section 5 outlines the related work on flexible process execution and process mining. Finally, section 6 concludes the article, goes into the current limitations of the approach and provides an outlook on planned future work.

2 Navigating through Modelled Business Processes

Generally, IT support for business processes requires a compromise between control and flexibility [3]. Current solutions for executing modelled processes primarily focus on control. They support processes with little human participation, predetermined paths and predictable choices that focus on orchestrating services and applications [4].

We call this type of processes the “automation processes”. They are well understood and highly evolved solutions exist on the market. “Knowledge-intensive processes”, on the contrary, are driven by human participation, often contain unforeseen paths and mostly depend on human decisions. The goal of the PN system is to execute this type of business processes models.

2.1 Declarative Process Modelling

Following the terminology of programming languages, there are two paradigms of describing business process models: the imperative and the declarative style [6]. The imperative way corresponds to imperative or procedural programming where every possible path must be foreseen at design time and encoded explicitly. If a path is missing then it is considered not allowed. Classic approaches like the BPEL [5] or BPMN [4] follow the imperative style and are therefore limited to the structured type of processes. In declarative modelling, on the other hand, process models specify the possible ordering of events implicitly by constraints instead of explicitly specifying all the allowed sequences of tasks. As a result, Process Navigation relies on a declarative representation for supporting knowledge-intensive business processes.

2.2 Cross-Perspective Modelling

Declarative modelling is based on constraints that relate events of the process and exclude or not recommend certain correlations. Both constraints and events must be able to involve all the perspectives of a business process like, e.g., incorporated data, agents performing the work and utilized tools [7]. On this way it becomes possible to express realistic correlations like, e.g., the actual performing agent of a step affecting the type of data used in another step [8].

2.3 Different Modalities and Explanation

A business process usually consists of several “facets” like, e.g., a legal framework (mandatory, “must”) and best practice (recommended but facultative, “should”). Classic approaches like the BPMN only allow for describing one of these facets per model. Combining both of them in one model greatly enhances its documentary character and allows for a BPM system to act more flexibly. An action that, e.g., is contrary to best practice but conforms to the legal framework is offered but marked as not recommended. The BPM system may even explain why the action is not recommended by tracing it back to the process model.

2.4 Implementation of the Declarative Execution Core

The PN engine interprets declarative process models and recommends tasks to participants and manages process data. Therefore, it has to find next feasible actions based on the process constraints (model) and the already performed actions (log). Additionally, feasible actions need to be categorised into feasible but not recommended and recommended actions. This task is represented as a planning problem and solved by the search-based optimisation framework JBoss Drools Planner. Based on the current event log, it generates all feasible next actions within the boundaries of the process's hard constraints (e.g. legal restrictions) and scores these actions on the basis of the soft constraints (e.g. best practice). Each time an action violates a soft constraint this violation is serves for explaining to participants why the action is not recommended. The details of this implementation can be found in [9].

2.5 Imperative Execution Core and Adaptability

As mentioned above, knowledge-intensive processes do not replace automation processes. Situations remain where an imperative style of description is best suited. To come up with this situation, the PN consists of two distinct execution cores for each paradigm on a common foundation layer. Both automation and knowledge-intensive processes may be executed by the same system and many common aspects like persistence, transaction management and management of process data are shared.

3 Observing and Analysing Unmodelled Business Processes

Even though Process Navigation provides more flexibility to participants by supporting knowledge-intensive processes, the execution of business processes is still based on a predefined process model. In situations where no model can be foreseen the process must be performed without support. Comprehensive process management requires methods to overcome the separation of modelling and execution phase by applying observation and analysis methods of executed processes [10].

3.1 Observing Process Execution

In order to support unmodelled situations, the "actually" performed process needs to be recorded. Process Observation (PO) [11] provides a solution where participants record, i.e., "digitize", what they are currently doing, i.e., they provide information about the process they are performing. By providing information about the process steps as well as incorporated data objects, the system accumulates execution information that can be used to automatically generate process models and dynamic guidance feedback for future process execution [11, 14]. Additionally, it is desirable that the observation of unmodelled processes is already supported by an IT system. We identify two different types of support mechanisms: structural and behavioural support functionality.

Structural Support during Observation. Structural support mechanisms depend on process "skeletons" which consist solely of process steps. These serve as static guidelines, i.e., independent of the users' behaviour. By starting a special process, these mandatory steps are displayed to the user. This way, participants can leverage "templates" and additionally complete process information with dynamically occurring missing steps. Structural support mechanisms are implemented by adopting concepts of Adaptive Case Management (ACM) systems [12, 13].

Behavioral Support during Observation. Behavioural support functionality makes use of the recorded process execution information. PO discovers workflow patterns that provide guidelines through the process. As the data basis grows with progressing enactment, quantity and quality of discovered workflow patterns will dynamically change. We adapted association rule mining to analyse process execution logs [18]. The resulting association rules are used for guiding process participants through process execution. Therefore, the collected process execution information of the PO is periodically transformed to an input dataset for association rule mining. The transformation algorithm is described in [18]. Subsequently, the Apriori algorithm is applied to this dataset. The algorithm extracts a set of association rules. Rule (1), e.g., claims that every time process step B has been performed after step A, process step C followed (numbers represent time steps).

$$A(0) \wedge B(1) \rightarrow C(2) \tag{1}$$

The PO system manages currently extracted association rules. If a users' behaviour satisfies the left-hand side of a rule, i.e., in this case a user performed step B after step A, the right-hand side of the rule is recommended, i.e., to continue by process step C.

3.2 Evolution of Imperative Process Models

Although, the PO system supports users by structural and behavioural guidelines, the development of complete process models, e.g., for documentation purposes or workflow management system (WfMS) deployment, remains the main goal. Therefore, the PO system extracts knowledge from event logs using process mining [15] techniques. In cases where the recorded information represents a structured process, where every possible path can be described clearly, the generation of an imperative process model, e.g., a BPMN model can be initiated. There are several well-known imperative process mining techniques that can be used to generate this kind of model [15]. Since the PN system provides an engine component for imperative modelling languages like BPMN [9], extracted models can deployed and executed by the navigation system. The PO system allows for the evolution of complete process models that form the basis for future process execution with the help of a WfMS.

3.3 Extraction of Declarative Process Models

Imperative process mining techniques construct models explicitly encoding all possible behaviours [15]. In contrast to imperative modelling, declarative models concentrate on describing what has to be done and the exact step-by-step execution order is not directly prescribed. There are several process mining approaches that are discovering declarative constraints. The approach of [16] is included in the PO system and enables the extraction of declarative process models that can be executed by the declarative engine component of the PN system [9]. As the declarative models always consider all possible solutions, the number of paths through the model can become incredibly large [14]. This is why guidance through a flexible process model is necessary [17]. Therefore, best-practice workflow patterns serve as guidelines through the process. The combination of declarative guardrails and best practice guidelines finally forms an all-embracing input for powerful execution support through the application of the PN system.

4 Integrating Process Observation and Navigation

For seamlessly supporting both modelled and unmodelled business processes, PN and PO must be integrated. The first task is to find a common data model for representing the entities of process models (e.g. processes, resources, data objects) and their logs (e.g. events, projects, values). Processes may be represented as BPMN or DPML (declarative process modelling language) process models and may consist of sub-processes. Further model entities are the resources that perform process steps and data objects that may be produced or consumed. When a process is executed an instance is created. Process steps may be activated, i.e., assigned to potential performers and completed by an actual performer. When a step is completed values of data objects may be consumed and/or produced. The PN component writes execution logs, i.e., events when executing processes and may read reconstructed process models for executing them. The PO component also writes events when observing unmodelled processes and writes models after analysing event logs.

4.1 Specialization of Process Steps

One way of integrating navigation and observation is the refinement of a certain process step that has been modeled as an atomic step in the first place. The participant selects the step and chooses to let his/her work to be observed. When the originally atomic step - which is now a composite process - is finished, PO analyses the occurred events and generates a process model reflecting the recorded behavior. Here, it is necessary to choose a suitable mining method: less-structured work should be analyzed by declarative process mining while structured routine work should be analyzed by imperative mining methods. The resulting model of the composite process may now be embedded into the surrounding process model.

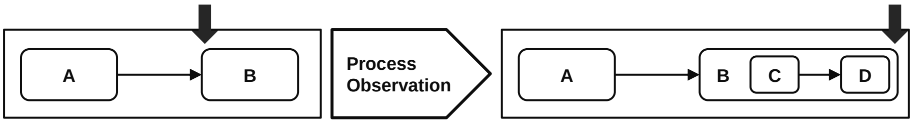


Fig. 2. Specialization of process steps by the use of Process Observation

Consider the example from Figure 2. At the beginning, a process participant is guided through a predefined process model, i.e., the workflow from step A to B, by the process navigation system. Being on the brink of performing step B, the user, who is an expert in his field, has the opinion that the abstract description of process step B could be refined. Future process performers should benefit from a more detailed description of the work to be done. This is why the user starts the PO interface where he has the possibility to record his activities, in this case two atomic process steps C and D, and preserve them for subsequent analysis. Finally, the user marks process step B as completed and returns to the navigation interface where he continues the predefined workflow. The above scenario covers a situation where existing models could be refined by expert staff while performing the process. The described functionality becomes even more useful, when we consider the application in the field. Here, process participants frequently find solutions to problems on their own. This knowledge should be preserved for future cases.

4.2 Executing Modelled Parts in Free Situations

Until now the integration of PO and PN manifested in the specialization or extension of already predefined models by observing the actual execution by the use of PO. Therefore, the PN system invokes the PO system when needed. However, there are also cases where predefined model parts can be used in free situations where process execution is completely carried out only with support of PO, e.g., the observation of whole processes that have never been modelled before. Here, users dynamically instantiate processes from predefined templates, i.e., skeletons of process models without any control flow information, and add newly occurring processes. In case that a sub-process has already been modelled before, the available process model can be executed with support of PN. Hence, the PO system invokes the PN system delivering the process to be performed. On this way, process participants can leverage a full WfMS in case of a predefined process model.

5 Related Work

The most recent approach in the field of declarative process execution is the Declare framework [19]. It is based on linear temporal logic (LTL) and therefore allows for relating process steps by temporal and existential constraints. These constraints may not contain statements on data, agents or tools. The only way of relating the temporal order of steps to these perspectives is to make the constraints depend on certain conditions. Such a conditional constraint only applies if its condition evaluates to true. Though a condition could then contain statements on data, agents and tools, the actual constraint remains limited to temporal order and existence of steps. The other perspectives cannot be constrained, which reduces the expressivity of the supported process modelling languages. For execution, the LTL formulae of a process are transformed into a finite state automaton which will then accept every trace of events that complies with the formulae. In order to reach a technically feasible size of the automaton, only the completion of a step is considered. Though a distinction between optional and mandatory constraints is made in the theoretical preliminaries, distinct modalities are not supported because only one automaton is generated for the mandatory formulae. Both the LTL formulae and the automaton must be transformed and reduced for necessary optimization reasons. Due to that, it becomes impossible to draw a connection between the automaton's transitions and the originally modelled constraints. Therefore, Declare cannot support traceability during execution as the proposed actions cannot be explained. In spite of the simplifications and reductions, the LTL-based implementation of Declare suffers from scalability issues [20]. Process models of realistic size lead to large automata which have to be generated completely before execution. There are several approaches that are very similar to Declare. In the work of Sadiq et al. [21] and also in the work of Wainer et al. [22], temporal constraints like, e.g., serial, order and fork are used to relate steps. As for Declare, these constraints may neither depend on nor influence perspectives like data, agents or tools and modalities are not supported either.

Adaptive Case Management (ACM) reflects a more flexible approach to supporting work [12, 13]. Instead of predefining every possible process step or path, ACM systems allow participants to dynamically instantiate processes from templates as well as newly occurring processes when needed. There are already mature implementations of ACM, e.g., [17]. However, existing ACM approaches lack the use of recorded information for guidance feedback and process model evolvement and the integration with WfMS. A further process flexibility approach is the ADEPT framework [26] that enables participants to dynamically change process definitions at run time. However, the approach is still based on an imperative prescription of process models that is often not suitable to describe less-structured processes. The work at hand provides an approach to combine both worlds of process support. It relies on the recording, i.e., logging, of actually performed processes and the subsequent analysis of the accumulated execution data. We already introduced an approach for manually generating process execution data in [11]. However, this solution was not operational enough, since users where not supported, e.g., by providing process templates. Through the integration of ACM-concepts, usability considerably increased. Latest pattern recognition methods offer the possibility to extract complete process models [15] that can be deployed in WfMS. Van der Aalst et al. developed techniques and applied them in the context of workflow management under the term process mining [15]. There are several algorithms that aim at generating process models automatically and focus different perspectives of process data. Many of these traditional mining algorithms are imperative approaches [15]. These methods construct imperative models explicitly showing all possible behaviours. Other ways to mine for process models are declarative approaches. There are several declarative discovery algorithms like [16, 23].

In cases where no complete process model could be extracted, workflow pattern mining methods can be used to find unknown coherencies in process logs. Representatives are sequence [24] or episode mining [25] that extract frequently occurring fragments of processes. However, these methods are limited to the extraction of rules considering the execution order of processes. Other types of process information, like incorporated data or agents are neglected.

6 Conclusion, Limitations and Outlook

This contribution demonstrates how to support the full range from highly controlled to fully flexible processes by integrating Process Navigation and Process Observation. Previously modelled business process parts (usually office work) are executed by the Process Navigation engine while unmodelled process parts (usually field work) are supported and reconstructed by the Process Observation system. A drawback of declarative models is that rule-based descriptions of processes generally are known to suffer from understandability issues [6]. One way of addressing this problem is to continuously simulate the execution of a process model. Therefore, a further objective is to develop a framework for the stepwise simulation of declarative process models so that their behaviour may be completely understood. Discovering workflow patterns using the Apriori algorithm may result in a high number of constraints. Many of them are trivial like, e.g., the fact that the performer of a process step is always a person. A future task is to reduce the number of constraints by identifying the trivial one.

Acknowledgement

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

References

1. Muehlen, M., Ho, D.T.: Risk Management in the BPM Lifecycle. BPM 2005 Workshops. pp. 454–466. Springer-Verlag Berlin Heidelberg (2006).
2. Zairi, M.: Business Process Management: a Boundaryless Approach to Modern Competitiveness. Business Process Management Journal. 3, 64–80 (1997).
3. Pešić, M., Schonenberg, H., Van der Aalst, W.M.P.: Declarative Workflow. In: ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., and Russell, N. (eds.) Modern Business Process Automation. pp. 175–201. Springer (2010).
4. Object Management Group Inc.: Business Process Model and Notation (BPMN) Version 2.0, http://www.omg.org/spec/BPMN/2.0, (2011).

5. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services - Version 1.1, (2003).
6. Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., Zugal, S.: Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. Enterprise, Business-Process and Information Systems Modeling (10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009, held at CAiSE 2009). pp. 353–366. Springer, Amsterdam, The Netherlands (2009).
7. Jablonski, S., Bußler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation. Thomson, London (1996).
8. Igler, M., Faerber, M., Zeising, M., Jablonski, S.: Modeling and Planning Collaboration in Process Management Systems using Organizational Constraints. 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing. pp. 1–10. IEEE, Chicago, IL, USA (2010).
9. Zeising, M., Schönig, S., Jablonski, S.: Improving Collaborative Business Process Execution by Traceability and Expressiveness. 8th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing. pp. 435–442. IEEE Computer Society, Pittsburgh, PA, US (2012).
10. Schönig, S., Seitz, M., Piesche, C., Zeising, M., Jablonski, S.: Process Observation as Support for Evolutionary Process Engineering. International Journal on Advances in Systems and Measurements. 5, 188–202 (2012).
11. Schönig, S., Günther, C., Jablonski, S.: Process Discovery and Guidance Applications of Manually Generated Logs. 7th International Conference on Internet Monitoring and Protection (ICIMP 2012). pp. 61–67 (2012).
12. Swenson, K.: Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize The Way That Knowledge Workers Get Things Done. Meghan-Kiffer Press (2010).
13. Fischer, L., Swenson, K., Palmer, N., Silver, B.: Taming the Unpredictable: Real World Adaptive Case Management: Case Studies and Practical Guidance. Future Strategies, Inc. (2011).
14. Günther, C., Schönig, S., Jablonski, S.: Dynamic Guidance Enhancement in Workflow Management Systems. Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012). pp. 1717–1719. ACM Press, New York, NY, USA (2012).
15. Van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011).
16. Schönig, S., Günther, C., Zeising, M., Jablonski, S.: Discovering Cross-Perspective Semantic Definitions from Process Execution Logs. 2nd International Conference on Business Intelligence and Technology. pp. 1–7. , Nice, FR (2012).
17. Kurz, M., Herrmann, C.: Adaptive Case Management – Anwendung des Business Process Management 2.0-Konzepts auf wissensintensive schwach strukturierte Geschäftsprozesse. Dienstorientierte IT-Systeme für Geschäftsprozesse, Bamberg, (2011).
18. Schönig, S., Zeising, M., Jablonski, S.: Adapting Association Rule Mining to Discover Patterns of Collaboration in Process Logs. 8th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing. pp. 531–534. IEEE Computer Society (2012).
19. Pešić, M.: Constraint-Based Workflow Management Systems: Shifting Control to Users, (2006).
20. Westergaard, M.: Better Algorithms for Analyzing and Enacting Declarative Workflow Languages Using LTL. BPM 2011. pp. 83–98. Springer Berlin Heidelberg (2011).
21. Sadiq, S., Orlowska, M., Sadiq, W.: Specification and Validation of Process Constraints for Flexible Workflows. Information Systems. 30, 349–378 (2005).

22. Wainer, J., Bezerra, F.: Constraint-based Flexible Workflows. In: Favela, J. and Decouchant, D. (eds.) Groupware: Design, Implementation and Use. pp. 151–158. Springer, Autrans, FR (2003).
23. Chesani, F., Lamma, E., Mello, P., Montalo, M., Riguzzi, F., Storari, S.: Exploiting Inductive Logic Programming Techniques for Declarative Process Mining. Transactions on Petri Nets and Other Models of Concurrency II. 5460, 278–295 (2009).
24. Srikant, R., Agrawal, E.: Mining Sequential Patterns: Generalization and Performance Improvements. 5th International Conference on Extending Database Technology (EDBT '96). pp. 3–17 (1996).
25. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery. 1, 259–289 (1997).
26. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support. Computer Science-Research and Development 23.2: 81-97 (2009).

Building a High-Level Process Model for Soliciting Requirements on Software Tools to Support Software Development: Experience Report

Ilia Bider[1,2], Athanasios Karapantelakis[3], Nirjal Khadka[1]

[1] Department of Computer and System Science (DSV) of Stockholm University, Kista, Sweden
[2] IbisSoft AB, Stockholm, Sweden
[3] Ericsson Research, Kista, Sweden
ilia@{ibissoft|dsv.su}.se, athanasios.karapantelakis@ericsson.com, nirjal@gmail.com

Abstract. Use of software tools to support business processes is both a possibility and necessity for both large and small enterprises of today. Given the variety of tools on the market, the question of how to choose the right tools for the process in question or analyze the suitability of the tools already employed arises. The paper presents an experience report of using a high-level business process model for analyzing software tools suitability at a large ICT organization that recently transitioned to scrum-based project methodology of software development. The paper gives overview of the modeling method used, describes the organizational context, presents a model built, and discusses preliminary findings based on the analysis of the model.

Keywords: Business process, requirements elicitation, software development, Scrum, project management, tool support, business process modeling.

1 Introduction

This paper is an experience report from a project that had both practical and research objectives. The project was conducted at a large ICT organization that recently went through reengineering of their software development process. The reengineering concerned transforming this process:

- from a traditional phase-based development approach with local software development teams
- to working in an iterative manner using the Scrum project management methodology and employing geographically distributed teams that have cultural differences and may work in different time zones.

The following two practical objectives were identified for our project:

1. Gain better understanding of current software development practice and map this understanding to a model

2. Use the model to analyze the suitability of project structure and software tools currently supporting the development process and find potential improvements both in the structure as well as the tools themselves.

For completing these tasks, we used a business process modeling technique from [1]. A model build with this technique is expressed in terms of a small number of steps (phases) of the process and relationships of different kinds between these steps. Some of these relationships are well known, e.g. input/output relationships [2]; others are new, like weak dependencies and team intersections. [1] suggests using such a model for choosing an appropriate cloud service to support the process in question. In this paper, we apply the model from [1] for different purpose, see 2 above. In connection to using modeling technique from [1], the following two research objectives were identified for the project:

1. Test the technique suggested in [1]. In particular, we wanted to see whether a high-level process model suggested can be built for a relatively complex process.
2. Test whether such a model can be used for a different purpose than originally suggested.

The rest of the paper is structured in the following way. In Section 2, we describe the project and its organizational context. In section 3, we give a brief overview of the framework suggested in [1]. In Section 4, we present a high-level model of the software development process built based on the framework. In Section 5, 6 and 7, we analyze tools used for supporting the process and give our suggestions for improvements. Section 7 summarizes the findings and draws plans for the future.

2 The project

In order to achieve our objectives as outlined in Section 1, we studied the software development process in one of the Product Development Units (PDU) of a large ICT organization. The product under development was a complex, commercial-grade telephony switching software solution. The organization in question has recently transitioned from a waterfall-like, phase-based software development approach to more flexible iterative development process, using the Scrum framework for project management. Given the novelty of this transition to a new development approach, there was vested interest in potential findings of our study not only from us researchers, but also from project managers from the development unit, who were interested in using our results to enhance the performance of the development process for future product development.

In order to gain understanding on how the development process worked, we started with brainstorming sessions between the members of the research team, without involving external stakeholders. Our research team had access to relevant documentation on the development process such as project plans, process workflow charts, and software development tools documentation.

The results of the aforementioned brainstorming sessions allowed us to form an initial hypothesis around how the process worked. We formalized this hypothesis into a mental model, focusing on identifying the actors participating in the development process and the interactions between them. The intention was to use this hypothetical model as support to our discussions with representatives from the development units. Through those discussions, the model would be refined and finalized. Table 1 list and describes the different actors in the development process.

Table 1. Description of actors and job roles in the product development

Actor	Description	Job Role
Project Management (PM)	Project management is responsible for coordinating geographically distributed software development teams by ensuring that each team's deliverables are finished on time. A sub-project manager is responsible for communicating requirements from project management to the leader of each team, and receiving feedback on the team's progress from the team leader. A Total Project Manager holds regular meetings with sub-project managers to stay informed on the progress of individual teams.	Total Project Manager, Sub-project managers
System Management (SM)	System management has a broad technical perspective of the product under development and is responsible for extending the business requirements drafted from external stakeholders to technical requirements to be used by software development teams.	System Manager
Feature Team X (FT-X)	Feature teams are geographically distributed software development teams. Each team develops a specific part of the product (one "feature"). A team leader is responsible for facilitating communication between the feature team and a representative from PM.	Scrum Master, SCRUM team member
Release Management (RM)	Release management is responsible for integrating all features delivered from the feature teams into a complete product and authorizes the software for commercial release by doing quality assurance ("acceptance testing").	Quality Assurance (QA) Manager, QA-team

Using data from Table 1, we identified a number of individuals representing different actors and interviewed them. The interview questions concerned the type and quality of interactions of the actor being interviewed with other actors, in order to identify potential issues hindering the efficiency of these interactions (for example a sub-optimal performance of a tool leading to inefficient communication).

3 Overview of the high-level business process modeling

According to [1], a high-level business process model consists of the following elements (syntactical units of the modeling language):

- Steps that represent work-packages to be completed in the process, each step having a unique name.

- Relationships between the steps. Relationships are typed and there can be more than one relationship between a pair of process steps, different relationships belonging to different types. Depending on the type, relationships can be symmetrical, or asymmetrical. In case of asymmetrical relationships there can exist up to two relationships for a given pair of steps. An example of an asymmetrical relationship between the steps is the input/output relationship. If a relationship exists between steps A, B it means that there is formalized output from step A that is used as input for step B, but not the opposite.
- We introduce two ways of depicting a model for a particular business process, namely using a graphic notation or relationships matrices.
- In the graphical form, the model consists of a set of diagrams, one for each type of relationships. In a diagram, steps are represented as rectangles (boxes) that have step names as labels inside them. Relationships are represented as lines between the step boxes.
- In the matrix form, the model consists of a set of square matrices, one for each type of relationships, where both columns and rows correspond to the process steps. Intersection between a row and a column in a matrix shows a relationship between the two steps. The type of content in the cells depends on the relationships type.

In this paper, we mostly use the matrix form for depicting a model, as it is easier to work with matrices in a formal way. Some relationships are basic, i.e. they are determined when building a model. Other relationships are derived to be used for determining requirements on tools to support the process. A derived relationships matrix is obtained via transforming a matrix of one of the basic relationships, or via merging two or more other matrixes.

We assume that the number of steps chosen for building a model is rather small (under 10, preferably 5 or 6), so that the whole model is compact in both its matrix and graphical form. Having a small number of steps means that each step represents a rather large work package, which on its own can be split in smaller steps later.

A detailed description of the relationship matrices, which constitute the semantics of our modeling language, is presented in the next session wherein we describe the model built for our case.

4 The model described

4.1 Steps

We identify seven steps in the PDU software development process as illustrated in Fig. 1 (together with input/output relationships, see section 4.2). The central and most complex part of the diagram in Fig. 1 is the development of separate software modules (part of the total software product, also known as “features”) and their integration coordinated by project management. Due to the size and complexity of the product, software development is split to parts and assigned to different teams across the world. These parts are also known as product features, representing a subset of the functionality of the complete product. Each team delivers their own feature; hence the teams are given the name "feature teams". Feature teams periodically deliver their part of functionality to Release Management for integration with other features. Note that in some cases, there exist inter-dependencies of features; in this case, a feature developed by one team may depend on a feature developed by another team. In such cases, one team has to wait for the other to deliver, which can cause delays to the release schedule (see section 4.2).

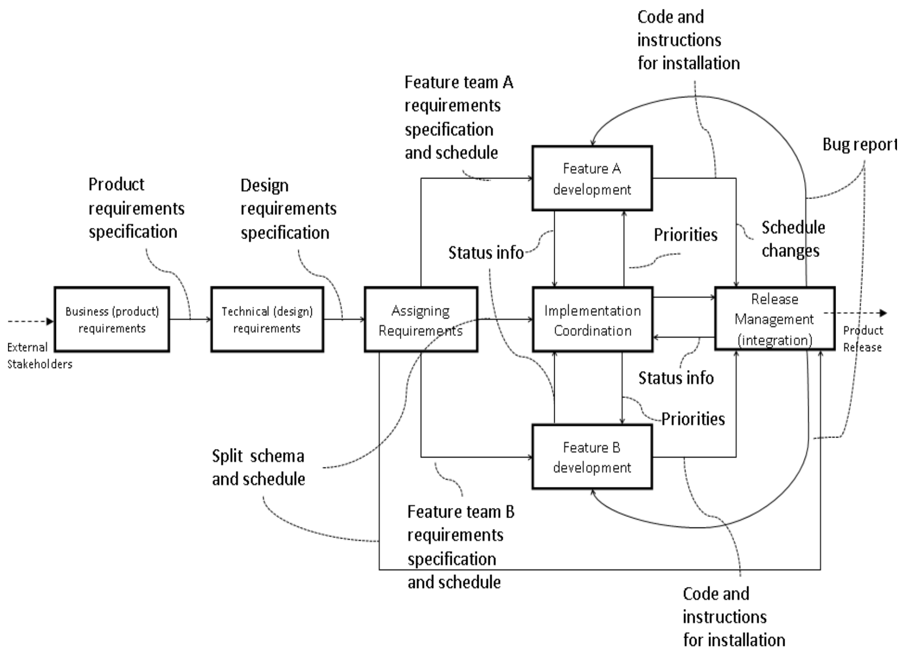


Fig. 1. Software development steps and input/output relationships (partial view).

For reasons of simplifying the model (see Fig. 1), we present only two feature development steps (features A and B, developed by feature team A and B respectively). In a realistic scenario, there can be more than two such steps; however, we consider the abstraction to two steps sufficient to capture feature teams interdependencies, wherein feature B depends on feature A. The rest of this subsection describes the steps in our model in greater details:

1. BR: Business *(product) requirements* – The business requirements are created as a result of a process involving identification of a business opportunity, discussion on its commercial viability, and ultimately drafting of a formal business requirements document stored in a product management tool.
2. TR: *Technical (design) requirements* – business requirements are converted into design requirements using the expertise of the System Management group (see Table 1). The outcome is a list of technical requirements stored in a requirements management tool. Typically, from one high-level business requirement, a number of technical requirements are created.

3. AR: *Assigning requirements* to feature development teams – the technical requirements are analyzed and their implementation is assigned to various teams. The process of analysis and assignment is done in meetings with participants from the feature teams, as well as project and system management. At the same time the schedule of delivery to Release Management is designed, with the intention to meet the deadlines set by the project timeline.
4. FD-A: *Feature A development* – this step represents the internal process of a team developing feature A. In case of feature dependencies, team A has to deliver both to the Release Management team for integration with other deliverables, but also to team B – for integration with their own deliverable, which depends on feature A. Feature team A in this case would receive bug reports – if any – on their deliverable from both team B as well as the release management team.
5. FD-B: *Feature B development* - this step represents the internal process of a team developing feature B. In this case, the team expects to receive some part of functionality of feature A from feature team A before it can deliver to integration. There is an open line of communication via a bug reporting system towards feature team A in case bugs are discovered.
6. *IC: Implementation Coordination* – This step contains the activities of project management, which is responsible for coordinating development among feature teams and the release management team. Project management receives feedback on the development progress from the feature teams and communicates task prioritization based on the current project status and needs. Project management therefore ensures that release management receives the software deliverables from feature teams in time and maintains an open line of communication with release management, which reports back on whether the integrated feature set meets their quality requirements or there are any residual defects (e.g. bugs, missing functionality) found in one or more of the deliverables.
7. *RM: Release management (integration)* - In this step, a set of engineers forming the RM team integrates all deliverables from the feature teams. After integration, the RM team tests the product feature set in a process known as acceptance testing. If integration is successful and acceptance testing meets the quality criteria of the RM team, then the integrated feature set can be released to the customer (in what is known as Product Release – PR). In any other case, the product release schedule towards the customer is delayed and the issues observed in acceptance testing are submitted through an online bug tracking system to the feature teams (project management is also notified).

4.2 Input/output relationships in detail

Input/output relationships show dependencies of one step on the results achieved in another. In graphical form, the dependencies are represented as arrows between the steps, where the text attached to an arrow explains the nature of the output from one step serving as an input into another (see Fig. 1, 2).

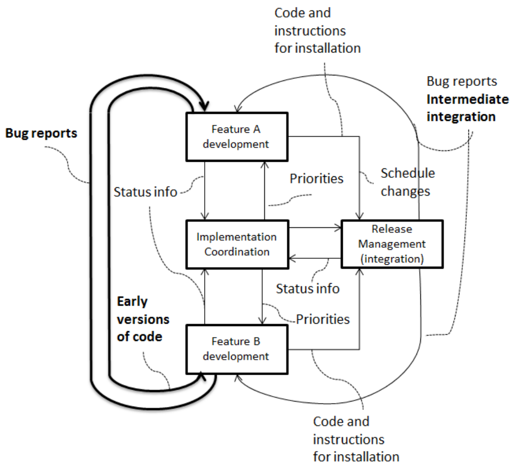


Fig. 2. Detailed view of the input/output relationships between the last 4 steps of the software development process under study. Relationships missing from Fig.1, are presented in bold. They show a feature inter-dependency scenario, where feature B is dependent on feature A.

Table 2. Input/output relationships as a matrix

Output Input	**BR**	**TR**	**AR**	**FD-A**	**D-B**	**IC**	**RM**
BR							
TR	*Product req. spec.						
AR		*Design req. spec.					
FD-A			*Feature A req. spec. ,Schedule			Priorities	Bug reports, Preliminary integration.
FD-B			*Feature B req. spec., Schedule	Early versions of code		Priorities	Bug reports, Preliminary integration
IC			*Split schema, Schedule	Status info	Status info		Status info
RM			*Split schema, Schedule	*Code and instruct.	*Code and instruct.	Schedule changes	

In matrix form, input/output relationships are shown in the following fashion: Cell (a,b) in the matrix, where a refers to a column and b to a row, specifies what result (i.e., output) from column step a (if any) is used as input to row step b. In addition to the name of result, a cell can be marked with asterisk (*) indicating that the result is required for step b to be started the first time. The input/output matrix as shown in Table 2, represents formalized casual relationships between the steps that are analogues to input/output connections in IDEF0 specification [2]. In addition to the formalized relationships, there could be informal input/output relationships between the steps, which are explained in Section 4.4.

4.3 Parallel execution and dependencies

The *parallel execution* matrix shows whether two steps are allowed to be executed in parallel. If ongoing activity inside step *a* does not forbid carrying out activity in step *b*, then both cells (*a*,*b*) and (*b*,*a*) are marked with *x* (the matrix is symmetrical). If none of the steps can run in parallel, the parallel execution matrix will be empty. This would be the case if the system development process in our example was carried out in successive phases (e.g. waterfall model). In our case, *Business requirements*, *Technical requirments* and *Assigning requirements* are executed in the sequential fashion, while *Feature A,B dvelopment*, *Implementation coordination* and *Release management* run in parellel. Due to the lack of space, we do not show this matrix.

[1] suggests combining the *input-output* matrix with *parallel execution* matrix to get a new view on complexity of the process. Table 3 is produced by merging Tables 2 and a corresponding parallel execution matrix according to a simple rule: cell(*a*,*b*) get crossed in the new table only if the cell is non-empty in both input-output matrix and parallel execution matrix. As only the last four steps are completed in parallel, we show in Table 4 only dependencies between the last four steps (all other cells will be empty. We will refer to the merged matrix as to *parallel dependencies* matrix. The cross in cell(*a*,*b*) in this matrix means that steps *a* and *b* can run in parallel at the same time as *b* is dependent on results from *a*.

The parallel dependencies matrix shows the potentially weak points in the process that require special consideration, otherwise the process will not run smoothly.

Table 3. Parallel dependencies between the last four steps of the model

	FD-A	**FD-B**	**IC**	**RM**
FD-A			x	x
FD-B	x		x	x
IC	x	x		x
RM	x	x	x	

4.4 Weak dependencies

Weak dependencies show whether one step might require information from another step that is not part of the relationships formalized in the *input/output* matrix. For example, IC may want to inspect the source code of a troublesome module to determine the severity of the problem when re-scheduling delivery. Cell (*a,b*) in this matrix specifies what kind of information from step *a* might be needed to complete step *b*. A part of the weak dependencies matrix for our systems development process is given in Table 4. This part concerns the most troublesome spots in the development process – the one that have parallel dependencies.

The concept of *weak dependencies* reflects the needs for informal communication in the frame of a process instance. It is not always possible to include everything that might be needed for the next step in the formal results, as different instances might require completely different information from the previous steps. It is better to start looking for this information on the demand basis, i.e., when there is a need for it.

Table 4. Weak dependencies

	FD-A	**FD-B**	**IC**	**RM**
FD-A		Explanations of code behavior		Explanation of test results/bug reports
FD-B	Details on the status Explanations of test results/bug reports			Explanations of test results/bug reports
IC	Details on the status	Details on the status		Details on the status
RM	Explanations of code behavior	Explanations of code behavior		

4.5 Teams and their relationships

The *teams* matrix shows the presence of collaborative teams and their relationships. The presence of teams is shown in the diagonal of the *team*s matrix :cell (*a,a*) is marked with the light gray color if the team for step *a* consists of more than one person. The non-diagonal elements show whether the teams participating in different steps intersect. If the teams for steps *a* and *b* intersect but not coincide, we mark both cells (*a,b*) and (*b,a*) with the light gray color. If the teams coincide, we mark these cells with the dark gray color. We also use the lighter gray color if the intersection is very “thin”. In our case, all steps have teams, and many teams intersect. The composition of the teams in terms of Table 1 is as follows (due to the lack of space, we do not present the matrix itself):

- BR team = External stakeholders (see Section 4.1) + SM
- TR team = SM + PM + representatives of FT-X, and RM
- AR team = SM + PM + representatives of FT-X, and RM
- FD-A team (FD-B has the same structure) = FT-X + Subproject-manager from PM
- IC team = PM

- RM team = RM

By merging the *weak dependencies* matrix with the *teams* matrix (Section 4.5), we get a view on the needs for inter-step collaboration. The part of the merged matrix for the last four steps is shown in Table 5. The cells with weak dependencies but without grey background warrant special attention when considering tool support. They mean there are needs for informal exchange, while teams do not intersect. Even the non-empty cells with the background color very light may require special attention as intersection may not be strong enough to serve as the only channel for informal exchange. As can be seen from Table 5, in our process, there are a number of empty cells with the white background, or only a slightly gray one. This requires special attention when analyzing the tools used for process support.

Table 5. Weak dependencies merged with Teams matrix

	FD-A	**FD-B**	**IC**	**RM**
FD-A		Explanations of code behavior		Explanation of test results/bug reports
FD-B	Details on the status Explanations of test results/bug reports			Explanations of test results/bug reports
IC	Details on the status	Details on the status		Details on the status
RM	Explanations of code behavior	Explanations of code behavior		

5 Requirements on tool support

Our previous research on business process support services lists a number of capabilities to be expected from process support tools [1]. In this paper, we will consider only three of them, namely: (1) *Information Logistics Support* (*ILS*), aimed at providing process participants with all information they need to complete their work without being overwhelmed by the details that are not relevant; (2) *intra-step collaboration support* aimed at providing a team working on the same step with means to store/retrieve intermediate results and communicate internally synchronously and/or asynchronously; (3) *inter-step collaboration support* for providing the teams, or individuals working on different steps with means to access intermediate results obtained in each other's steps and communicate between them synchronously and/or asynchronously.

ILS is important to have when there are input/output dependencies between the steps the teams of which do not coincide and can be provided in two ways: (a) by sending the results to the next step team, e.g., via email; (b) by providing a shared space where the results are stored and made available for the participants of the next step. The second type of logistics is more appropriate when there are loops in input/output flow and/or there are parallel dependencies. Loops can be detected via analysis of the input/output matrix (see Section 4.2). Presence of loops is identified by symmetrical non-empty elements in this matrix (or a derived matrix with transitive input/output relationships [1]).

In our case, Table 2 shows several loops (they are also visible in Fig. 1 and 2). Presence of loops indicates that the same input can be provided several times which is best handled by a shared space with version control. Parallel dependencies identify that input can be sent in portions, and the next portion can negate what has been sent in the previous one. When such disrupted input is expected, prompt notification of the new portion arrival is needed. This can be arranged via shared spaces supplemented by notification mechanisms. Table 2 shows a number of parallel dependencies which should be taken care of by process support tools.

Intra-step collaboration support is required when there is a team in at least one step. The presence of step teams can be seen in the *teams matrix* (see Section 4.5). As was mention in Section 4.5 there are teams for each step identified in our process, which warrants the needs for intra-step collaboration support.

Inter-step collaboration support is required when there is informal information exchange between the steps the teams of which do not intersect, or their intersection is too "thin". These situations can be identified by analysis of the derived matrix acquired by merging the *teams matrix* with the *weak dependencies* matrix, see Section 4.5. Table 5 includes non-empty elements with white background (no intersection) or very light background ("thin" intersection), showing importance of *intra-step collaboration support* for our process.

6 Tools in use

General Purpose Tools (GPT) such as email, word/spreadsheet processors and instant messaging are used when necessary while the specialized tools listed below have specific functions:

- AB – Automated Build, continuous integration tool for building snapshots of code
- BR – Bug Tracker, an in-house developed bug reporting tool
- DE – Development Environment for authoring code.
- FT – Tool for function and unit test - automated test framework
- FS – Secure file server, for feature teams to deliver code to Release Management for integration or to other feature teams in case of feature inter-dependencies.
- PPM – Product and Portfolio Management tool for project management.
- RC – Requirements Composer for defining and following up the requirements.
- ST – Automated System Testing tool for regression testing
- TM – Test Management Tool for documenting test cases
- ST – Scrum Tool for backlog management.
- RP – Version control system, for collaborative code development in feature teams.

The mapping of tools to teams that use them is summarized in Table 6. The diagonal shows which tools are used in each step, other cells show the tools used for transferring input/output (Table 2) or as channels for weak dependencies between the steps. Thus, the non-diagonal part of Table 6 comprises two matrixes: one describes tools used for *ILS* (input/output), while the other is related to *intra-step collaboration* (weak dependencies).

Table 6. Tool usage

	BR	TR	AR	FD-A	FD-B	IC	RM
BR	PPM						
TR	PPM,GPT	RC,PPM					
AR		RC	RC,PPM				
FD-A			RC	DE,FT,FS, RP,ST,TM	GPT	GPT	BR,FS
FD-B			RC	FS,BR,GPT	DE,FT,FS, RP,ST,TM	GPT	BR,FS
IC			RC	GPT	GPT	RC,PPM	BR,GPT
RM			RC	FS, GPT	FS,GPT	GPT	AB,BR,ST,TM

7 Analysis of tools in use

Through analysis of tools, we found lack of support for weak dependencies between IC on one hand, and features teams and RM on the other (see the IC-row in Table 5). In order to learn about the status of the features development, project management manually checks with each feature team their progress (i.e. through e-mail or phone conversations), and records information on it in a spreadsheet. Because of the subjective nature of the progress reports, the project manager we have interviewed has to invest additional time to keep track of progress for some of the teams. To improve support for weak dependencies, a tool that provides integrated picture of the teams activities seems necessary. Such a tool could analyze activities around different requirements assigned to a team, e.g. code compilations, bug reports, etc. and give warnings on suspicious too much activity or luck of such.

8 Concluding remarks

In this paper, we applied the theoretical framework for selecting software tools for business process support from [1] to a real case of software development process in a multinational telecommunications provider. From our analysis, we found that the software development process was missing a tool for communicating the progress of software development from the features teams to Implementation Coordination which could result in delayed release schedules. In this way, we got a partial prove of applicability of the framework to complex processes and its usefulness.

References

1. Bider, I. and Perjons E. Preparing for the Era of Cloud Computing: Towards a Framework for Selecting Business Process Support Services. *LNBIP*, No 113, 2012, Springer, pp. 16-30
2. *Standard for Integration Definition for Function Modeling* (*IDEF0*), National Institute of Standards and Technology (NIST), FIPS publication, 183, December 1993.

Towards Implementing IT Service Management in an ERP for the IT Service Industry

Johannes Hintsch and Klaus Turowski

Center for Very Large Business Applications
Faculty of Computer Science
Otto von Guericke University Magdeburg
{johannes.hintsch,klaus.turowski}@ovgu.de

Abstract. ERP systems were fundamental in achieving efficiency gains in traditional manufacturing companies. In order to make the benefits of ERP utilizable by IT service providers, their processes need to be operationalized in ERP. IT service management (ITSM) frameworks describe these processes. They are however described using natural language. To be able to operationalize ITSM, the current state of the art of ITSM framework formalizations is analyzed. Also, it is determined which ITSM frameworks should be chosen for operationalizing IT service providers' processes. Finally, research gaps are identified.

Keywords: IT service management, ITIL, COBIT, ISO/IEC 20000, Meta-model, Ontology, Enterprise Resource Planning, Structured Literature Review

1 Introduction

Information Systems researchers have intensively discussed the industrialization of IT [1]. Standardization is one of industrializations core concepts achieving efficiency gains in value creation.

Enterprise Resource Planning (ERP) systems have been fundamental in traditional industries in order to support standardization. They are packaged application systems, support all processes and functions of an enterprise on a common database, and provide managers with a comprehensive view of the company's state [2, 3, 4]. ERP systems such as SAP's ERP usually ship with predefined processes that are proclaimed to be best practice and serve the standardization concept.

An IT service is "a service provided by an IT service provider. [It] is made up of a combination of information technology, people and processes."[5] The IT service industry consists of providers offering such services.

This paper argues that ERP systems, as they have supported companies of traditional industries in their value creation, are also suitable to support certain members of the IT service industry. In particular they can further standardization with a common process model. IT service management is "the implementation and management of quality IT services that meet the needs of the [customer's] business. ITSM is performed by IT service providers through an appropriate mix of people, process and information technology."[5]. IT service management represents the core of IT service providers' business which therefore should also be the core business concept of a provider's ERP.

The ERP envisioned will address the needs of members of the IT service industry that exhibit, in their service production, a high degree of standardization and automation with resulting economies of scale. These providers will often serve external customers offering large quantities of the same service. IT providers offering very individualized services, e.g. some specialized software developing, are not seen as users of such an ERP system.

An IT service management framework implemented in an ERP will serve the goal of standardizing processes within IT service providers. However, no academically researched reference model of an ERP system for the IT service industry taking an IT service management framework as a basis exists [6].

There are a number of IT service management frameworks that define the processes of IT service provider's core business. They provide process descriptions which could serve as a basis for creating the predefined processes that are part of ERP software.

The ITSM frameworks' processes are mostly defined using natural language, but, if they shall be operationalized by implementing them within an ERP system they need to be formalized. This paper defines formalizations as models and meta-models, but also as taxonomies and ontologies as they can formalize natural language definitions [7]. Thus, the first research question of the paper is the following. **RQ1:** What is the current state of the art of formalizations of IT service management frameworks?

Many authors describe the IT Infrastructure Library (ITIL) as the worldwide de facto standard for IT service management (e.g. [8]), which was also confirmed by the framework's documentation itself [9]. Often ITIL is called best practice and some authors even go as far as awarding it generic reference model status [10]. However, ITIL is a framework of common practices [11] which lacks scientific foundation [11, 12]. Thus, because more than twenty other frameworks exist [6] that are related to or intended for ITSM, the second research question is: **RQ2:** Which ITSM framework should be chosen as a basis for describing ITSM processes in an ERP for the IT service industry?

The identified works will provide starting points for the conceptualization of an ERP system. However, only some points are addressed. Thus, the task of the third research question is the following. **RQ3:** In what areas is further research necessary to build an ERP for the IT service industry?

The following section presents the literature review that was conducted to answer the research questions. These answers are developed in section 3. The paper ends with a conclusion section.

2 Literature review

Seuring and Müller describe how a literature review can be conducted in four steps [13]: collection, descriptive analysis, category selection and material evaluation.

2.1 Material collection

In material collection the scope of the search has to be set: finding publications that contain formalizations of ITSM frameworks. Thus, articles' abstracts should contain the abbreviation or full spelled word IT service management as well as the different defined types of formalization. ITSM is strongly related to IT Governance, therefore, the term was included as well.

To gain up-to-dateness publications not older than five years were searched, setting the timeframe to 2007–2013. Databases were selected that have a good coverage of information systems and computer science topics. Not all potentially relevant journals and conferences are covered this way. However, the references in identified relevant papers were also analyzed. Thus, the limitation of ignoring outlets is alleviated. Checking the references also limits the constrainedness of scope set by the timeframe. The parameters of the material collection are summarized in table 1.

Keywords: ((ITSM OR "IT service management" OR "IT Governance" OR "IT-Governance") AND (model OR "meta-model" OR meta-model OR meta-modelling OR "meta-modelling" OR taxonomy OR ontology)
Timeframe: 2007 – 2013
Databases: ACM Digital Library (ACM), AIS Electronic Library (AIS), Directory of Open Access Journals (DOAJ), EBSCOhost (EBSCO), IEEE Xplore - Digital Library (IEEE), SciVerse ScienceDirect (SVSD), and SpringerLink (SL)

Table 1. Material collection: keywords, timeframe, and databases

Papers had to be published in a journal or conference proceeding and be written in English. When all the articles were retrieved their abstracts were read, and irrelevant papers were sorted out. After the remaining papers were read in depth irrelevant papers were sorted out again. Papers were excluded whose topic was curriculum related, or which were introductory articles such as prefaces or editorials. Some authors formalize relevant content, Ebert et al. [14] for example present an IT service model, but do not base their work on an IT service management framework. These papers are also considered irrelevant.

Table 2 shows the number of papers that were analyzed in the different stages.

The high rate of irrelevant publications may be explained by the very general word model. Also some IT Governance publications were not relevant. Searching the references of the identified publications from table 2 identified six relevant papers [22, 23, 24, 25, 26, 27]. The structured literature review was extended

Source	Collect	Abstract	Full text	Papers
ACM	7	1	1	[10]
AIS	35	7	2	[15, 16]
EBSCO	13	1	0	-
DOAJ	3	0	-	-
IEEE	106	36	4	[17, 18, 19, 20]
SVSD	4	2	1	[21]
SL	135	2	0	-
SUM	303	49	10	

Table 2. Relevant papers in each stage: Collecting the papers, reading the abstracts and reading the papers' full texts.

by an unstructured search which found two additional papers on Google Scholar (GS) [12, 28]. Furthermore, several different management frameworks exist which include or are primarily designed for ITSM. A comprehensive collection of these frameworks was identified in [6]. These were analyzed for containing ITSM formalizations. IBM Tivoli Unified Process [29] was selected from amongst the 24 frameworks because it formalizes ITSM processes with workflow diagrams.

The descriptive analysis of a structured literature review will often feature a trend analysis of the number of papers identified per year in the search frame. However, due to the small number of found papers such an analysis is not performed here. The next section will describe the category selection and subsequent material evaluation.

2.2 Category selection and evaluation

In total, twelve works were identified that report on a formalization of an ITSM framework. The difference between sixteen identified papers and twelve works exists because Valiente et al. and Goeken and Looso report in different papers on the same model in different stages. A first result, as demonstrated by table 3 which categorizes the works by ITSM framework, is that considerable academic effort has been invested into the modelling of ITIL. Alongside ITIL, COBIT and ISO/IEC 20000 have also been researched.

The results also show that different versions of the framework were used, but only Huang et al. appear to have used a version of ITIL that was outdated at the time of publishing their paper. As none of the works are based on the newest versions of the frameworks, they need to be updated if applicable. The update from ITIL v3 to ITIL 2011 only comprises error corrections that are not substantial [30]. However, changes made in the update from COBIT 4.1 to COBIT 5 include newly added processes and the improvements are described as considerable [31]. The revisions made to ISO/IEC 20000 include alignment to other standards, change of terminology, and clarification of concepts [32].

Consequently, different versions of the ITSM frameworks have to be considered when updating the process model of an ERP system in order to align it

Authors	Framework	Framework version	Articles
Goeken and Looso	COBIT	COBIT 4.1	[23, 24, 15]
Spies	COBIT	COBIT 4.1	[19]
Braun and Winter	ITIL	ITIL v2	[10]
Jäntti and Eerola	ITIL	ITIL v2	[25]
Huang et al.	ITIL	ITIL v2	[18]
Valiente et al.	ITIL	ITIL v3	[27, 28, 20, 21]
Strahonja	ITIL	ITIL v3	[26]
Goeken et al.	ITIL	ITIL v3	[12]
Baiôco et al.	ITIL	ITIL v3	[22]
Rohloff	ITIL	ITIL v3	[16]
IBM	primarily ITIL	ITIL v3	[29]
Brenner et al.	ITIL, ISO/IEC 20000	ITIL v3, ISO/IEC 20000:2005	[17]

Table 3. Papers categorized by author and by ITSM frameworks

with the perhaps revised practices. This will come with challenges when processes were customized.

2.3 Limitation

This study is limited because it searches in a limited timeframe and a limited number of publication outlets. However, the analysis of the references of the relevant works somewhat alleviates this. In addition, the search was restricted to abstracts. This was done because databases have different search fields, but with all databases abstract based search was possible and therefore it seemed to be more consistent not to look in different parts of the paper in different databases. However, more relevant papers might be found if title and keywords fields would also be included. Last, the keywords represent a set of formalization terms which might not be exhaustive. This was tried to be addressed with the very general term model.

Now, considering the previous section ITIL, COBIT and ISO/IEC 20000 are only three frameworks of over 20 frameworks. Thus, the results section starts with answering the second research question.

3 Results

3.1 Choosing an ITSM framework

Considering the results depicted in table 3 it seems clear which ITSM framework the authors whould choose: "ITIL, of course, maybe COBIT and maybe ISO/IEC 20000". Why would they be right, considering that there are alternatives and ITIL, COBIT and ISO/IEC 20000 have not been scientifically conclusively evaluated? The three frameworks are based on extensive industry experience and have all undergone a rather extensive review and maturement process. They can be seen as proven practice solutions to manage the IT services of a firm. There might be other good or better frameworks. But these are the most widely approved and most widely adopted frameworks which exist to date. Thus, when conceiving an ERP for the IT service industry, which is supposed to be adopted by a lot of firms, a framework should be chosen that is close to what companies are already doing. Alternative approaches should however be analyzed in order to improve ITIL, COBIT and ISO/IEC 20000 if applicable.

Ensuing RQ2 comes the question of why COBIT and ISO/IEC 20000 should be used when ITIL is used. To answer this it is helpful to first look at the frameworks themselves. According to the first book of ITIL's documentation [5], ITIL shall provide guidance to IT service providers for their ITSM practices. It underlines the wide pervasion but also makes clear that organizations need to adopt ITIL according to their needs because it is not a standard. The authors of COBIT [33] underline the governance and management character of the framework. In contrast to ITIL it calls itself a holistic framework for enterprise IT, not specifically for ITSM. With the standard ISO/IEC 20000 [32] organizations can get certified that they manage their IT services according to the requirements outlined by the standard. The three frameworks are not mutually exclusive. COBIT is in parts based on and aligned with ITIL and ISO/IEC 20000 [33]. Whereas COBIT describes more what to do, ITIL describes how to do it [33]. ISO/IEC 20000 can be seen as a smallest common denominator of what is necessary to manage IT services. These assessments are also substantiated by looking at the randomly selected process of capacity management. Its description covers 22 pages of prose text in ITIL [34], four structured pages featuring bullet-lists and tables in COBIT [35], and a half of a page in ISO/IEC 20000 with requirements list and limited prose text. According to Sahibudin et al. [36] ITIL and COBIT cover the same processes except incident management.

Thus, as the frameworks are not mutually exclusive they can all serve different aspects of building an ERP for the IT service industry. This will be elaborated on in the next sections.

The following section answers the first research question.

3.2 Current state of the art of ITSM formalizations

Table 4 shows the different works with their type of formalization, the scope, and the type of disclosure. The scope column describes if an individual process or multiple processes are formalized. In order to conceptualize an information system it is necessary to have full access to the formalization. Thus, the fourth column describes to what extent the formalization is disclosed.

Goeken and Looso, Goeken et al. and Strahonja build meta-models of COBIT and ITIL. Amongst the three, Goeken and Looso's work is the one that is most broadly covered which is why it is presented here. They justify their choice because COBIT is structured and attempts to provide a holistic view on IT management. Goeken and Looso develop their meta-model in order to improve the scientific basis for ITSM frameworks. The model, which is presented in extended entity relationship notation, describes the structure of COBIT processes. This,

Authors	Formalization	Scope	Disclosure
Goeken and Looso	meta-model	cross-process	full
Spies	ontology	cross-process	partial
Braun and Winter	meta-model	IT service	full
Jäntti and Eerola	conceptual model	problem management	full
Huang et al.	meta-model	incident management	full
Valiente et al.	ontology	cross-process	full
Strahonja	meta-model	cross-process	partial
Goeken et al.	ontology	cross-process	partial
Baiôco et al.	ontology	configuration management	partial
Rohloff	process model	cross-process	partial
IBM	process model	cross-process	full
Brenner et al.	information model	cross-process	partial

Table 4. Papers categorized by type of formalization, by scope, and by type of disclosure.

for example, enables the addition of compliant new processes. The meta-model is built by adhering to a set of established modelling principles which Goeken and Looso extend by three principles of their own, particularly developed for meta-models. Among the found literature, theirs is the most extensive description of methodology used for creating a formalization.

Both Valiente et al. and Spies use ontologies. While the former authors formalize ITIL, Spies formalizes COBIT. They further define constraints and are consequently able to add semantics to their ontology. The work by Valiente et al. is more extensive which is why it is discussed in detail.

Valiente et al. develop their ontology of ITIL because the framework lacks semantics needed for automated processing. They integrate their ontology with the general purpose ontology OpenCyc. The full ontology is disclosed in [28]. By additionally defining constraints they can achieve semantic model consistency. Using the Semantic Web Rule Language, they define, for instance, that if an incident management activity is coordinated by a specific process, then that activity must be coordinated by an incident management process:

```
itil:Activity(itil:ICTD_IM_Activity) ∧ itil:Process(?p) ∧
itil:coordinatedByProcess(itil:ICTD_IM_Activity,?p)
→ itil:IncidentManagement(?p)
```

This enables them to define or test processes according to ITIL compliance. Valiente et al. validate their approach at a Spanish IT service provider who implemented the IT incident management process.

Jäntti and Eerola, Huang et al., and Baiôco et al. build conceptual models, a meta-model and an ontology for problem management, incident management, and configuration management. Rohloff builds process models for all ITIL v3 processes, but only discloses the model of one process as an example. IBM Tivoli Unified Process (ITUP) however, provides process models for all IT service management processes of ITIL and fully discloses them. According to the authors, ITUP is also closely aligned with several other ITSM frameworks, but the main source is ITIL. ITUP has no scientific foundation, but its authors claim that it is based on extensive industry experience. Thus, when using ITUP in a scientific context, it should be scientifically evaluated.

Braun et al. developed a meta-model of an IT service in order to integrate IT service management with enterprise architecture. The meta-model could be used as a data model, but, Brenner et al. provide a more convincing approach although they do not fully disclose their work. They adapt and extend the Shared Information/Data Model which is related to the enhanced Telecom Operations Map (eTOM). Their information model is developed to be compliant with ITIL and ISO/IEC 20000. It is intended to be used by companies that share processes and therefore need to exchange information in a standardized way.

Some work has been conducted in order to formalize ITSM. One difficulty is that the ITSM frameworks comprehensively cover their domain. The identified formalizations often only cover certain processes or aspects of the frameworks. Also, complete and soundly scientific evaluated formalizations are rare. Amongst the identified work the one by Valiente et al. stands out in these regards.

In the next section the third research question is answered: What future work needs to be done in order to build an ERP for the IT service industry?

3.3 Future work in conceptualizing an ERP for the IT service industry

Scheer describes five views that need to be considered when developing information systems [37]: organization, data, control, function, and output. When conceptualizing an ERP system these views offer a structure for determining what needs to be conceptualized. The identified literature partially fits to three of these views.

An information system is designed to support an organization. Different users have different duties, needs, or may have different authority levels. Therefore, access rights and functional roles need to reflect the organizational structure. COBIT provides structured and extensive information on process responsibilities and could consequently serve as a basis for this part of the ERP.

In the data view an information model for all relevant data must be developed. Here, the proposal from Brenner et al. [17] provides a promising foundation. Brenner et al. chose ISO/IEC 20000 for their data model. This was a good choice because a standard compliant data model will serve cross organizational collaboration and a common understanding of ITSM concepts.

The control view connects the organizational, the data, the functional and the output view into a sequential and logical series of process steps. Considering the analyzed literature, two alternatives exist for integrating ITSM processes as "best" practices into an ERP system.

The first is utilizing complete formalizations of ITIL such as done by IBM [29]. This has the advantage that the processes can be used as is. It has, however, been stated that ITIL should not be implemented by dogmatically following the guidelines [5, 38] but also considering the individual firm's requirements.

An alternative is presented by Valiente et al. [21] who formalize the knowledge of ITIL in an ontology and define constraints by which they can add further semantics. For an ERP system this approach could be used in order to allow customers to define their own processes that would be compliant with ITIL. Of course this means that when taking the system into operation an initial investment has to be put into implementing the processes.

As mentioned in the previous section the identified works lack the comprehensiveness of the formalized ITSM frameworks. Thus, the first open research gap is to evaluate and consequently extend the proposals for completeness.

The function view is described in ITIL: How are the services composed? What functions are required to provision a certain service? These questions need to be answered for firms individually, but a general formalization is still required. A potentially fitting concept might be the universal service description language (USDL) [39] which holds ready appropriate concepts.

Furthermore, the output view has not been addressed yet. Two aspects are how to publish and bill services. Again the possibilities of the USDL should be investigated here. It also needs to be considered how far the value creation (in this case the service provision) should be integrated into the ERP system. Different tools exist to support provisioning and it needs to be determined how much of their functionality the ERP system shall provide. This question is closely linked with potentially missing competitive advantage when an IT service provider uses standard software, because the more standardized the more difficult it is to have a competitive edge.

The mentioned starting points for the individual views all represent mere conceptual work. Also, their compatibility has not been evaluated. Thus, these concepts should be implemented and their combination tested. Also it needs to be decided how everything should be fit together. Will the control view for example be implemented by a monolithic system or could it be distributed on different systems via a service oriented architecture?

The last section concludes the paper.

4 Conclusion

In order to conceptualize an ERP for the IT service industry, this paper argues that ITSM frameworks must be formalized in order to operationalize them within ERP. A literature review was conducted in order to capture the respective current state of the art. Meta-models, ontologies, conceptual models, and process models were used for formalizations. Interesting approaches towards formalization include the work by Valiente et al. and Brenner et al.. Valiente et al. provide means, via an ontology and a rule set, to determine if an ITSM process is ITIL compliant. Brenner et al. use an information model which was closely aligned with the eTOM framework to adapt it to ITIL and ISO/IEC 20000.

Their model can serve as a data model for an information system supporting ITSM.

The paper also comes to the conclusion that ITIL, COBIT and ISO/IEC 20000 are the frameworks that should be focused on when developing an ERP system for the IT service industry.

This study identified the following points which need future work. The identified work needs to be evaluated for completeness. Furthermore, for the output view foundation the following needs to be determined: How can an IT service provider describe its services and functions adequately? How can the output of an IT service provider be supported? How comprehensive shall an ERP solution be in order to enable IT providers to maintain their competitive edge through individual IT solutions? Additionally, how can the different views be fitted together in an architectural framework?

References

1. Erbes, J., Motahari Nezhad, H.R., Graupner, S.: The Future of Enterprise IT in the Cloud. Computer **45**(5) (2012) 66–72
2. Chen, I.: Planning for ERP systems: analysis and future trend. Business Process Management Journal **7**(5) (2001) 374–386
3. Davenport, T.H.: Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, Boston (2000)
4. Klaus, H., Rosemann, M., Gable, G.: What is ERP? Information Systems Frontiers **2**(2) (2000) 141–162
5. Best Management Practice: ITIL – Service Strategy. The Stationery Office, Norwich (2011)
6. Hintsch, J.: ERP for the IT Service Industry: A Structured Literature Review. In Shim, J., Hwang, Y., Petter, S., eds.: 19th Americas Conference on Information Systems (AMCIS 2013), August 15–17, Chicago, IL, USA, AIS (2013)
7. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics (2001)
8. Wu, M.S., Huang, S.J., Chen, L.W.: The preparedness of critical success factors of IT service management and its effect on performance. The Service Industries Journal **31**(8) (2011) 1219–1235
9. The Stationery Office: Best Practice for Service Support. (2000)
10. Braun, C., Winter, R.: Integration of IT service management into enterprise architecture. In: Proceedings of the 2007 ACM symposium on Applied computing, March 11–15, New York, NY, USA, ACM (2007) 1215–1219
11. Hochstein, A., Zarnekow, R., Brenner, W.: ITIL as common practice reference model for IT service management: formal assessment and implications for practice. In: IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2005), March 29 – April 1, Hong Kong, China (2005) 704–710
12. Goeken, M., Alter, S., Milicevic, D., Patas, J.: Metamodelle von Referenzmodellen am Beispiel ITIL: Vorgehen, Nutzen, Anwendung. In Fischer, S., Maehle, E., Reischuk, R., eds.: Informatik 2009: Im Focus das Leben, Beitrge der 39. Jahrestagung der Gesellschaft fr Informatik e.V., September 28 – October 2, Lübeck, Germany, GI (2009) 3701–3714

13. Seuring, S., Müller, M.: From a literature review to a conceptual framework for sustainable supply chain management. Journal of Cleaner Production **16**(15) (2008) 1699–1710
14. Ebert, N., Uebernickel, F., Hochstein, A., Brenner, W.: A Service Model for the Development of Management Systems for IT-enabled Services. In Hoxmeier, J.A., Hayne, S., eds.: Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS 2007), August 9–12, Keystone, CO, USA, AIS (2007) 455–462
15. Looso, S., Goeken, M.: Application of Best-Practice Reference Models of IT Governance. In Alexander, P.M., Turpin, M., van Deventer, J.P., eds.: 18th European Conference on Information Systems (ECIS 2010), June 6–9, Pretoria, South Africa, Department of Informatics (2010) Paper 129
16. Rohloff, M.: A Reference Process Model for IT Service Management. In Benbasat, I., Montazemi, A.R., eds.: Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS 2008), August 14–17, Toronto, ON, Canada, AIS (2008) 1–10
17. Brenner, M., Schaaf, T., Scherer, A.: Towards an information model for ITIL and ISO/IEC 20000 processes. In: 11th IFIP/IEEE International Symposium on Integrated Network Management, June 1-5, Long Island, NY, USA, IEEE (2009) 113–116
18. Huang, X., Shen, B., Chen, D.: IT Service Support Process Meta-Modeling Based on ITIL. In: International Conference on Data Storage and Data Engineering (DSDE), February 9–10, Bangalore, India, IEEE (2010) 127–131
19. Spies, M.: Continous Monitoring for IT Governance with Domain Ontologies. In Hameurlain, A., Tjoa, A.M., Wagner, R.R., eds.: 23rd International Workshop on Database and Expert Systems Applications (DEXA), September 3–6, Vienna, Austria, IEEE (2012) 43–47
20. Valiente, M.C., Garcia-Barriocanal, E., Sicilia, M.A.: Applying Ontology-Based Models for Supporting Integrated Software Development and IT Service Management Processes. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **42**(1) (2012) 61–74
21. Valiente, M.C., Garcia-Barriocanal, E., Sicilia, M.A.: Applying an ontology approach to IT service management for business-IT integration. Knowledge-Based Systems **28**(0) (2012) 76–87
22. Baioco, G., Costa, A., Calvi, C., Garcia, A.: IT service management and governance modeling an ITSM Configuration process: A foundational ontology approach. In: IFIP/IEEE International Symposium on Integrated Network Management-Workshops (IM '09), June 1–5, New York, NY, USA, IEEE (2009) 24–33
23. Goeken, M., Alter, S.: Representing IT Governance Frameworks as Metamodels. In: Proceedings of the 2008 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE 2008), July 14–17, Las Vegas, NV, USA (2008) 48–54
24. Goeken, M., Alter, S.: Towards Conceptual Metamodeling of IT Governance Frameworks Approach – Use – Benefits. In Sprague, R.H., ed.: Proceedings of the 42ndAnnual Hawaii International Conference on System Sciences (HICSS 2009), January 5–8, Big Island, HI, USA, IEEE (2009) 1–10
25. Jäntti, M., Eerola, A.: A Conceptual Model of IT Service Problem Management. In: International Conference on Service Systems and Service Management, October 25–27, Troyes, France, IEEE (2006) 798–803
26. Strahonja, V.: Definition Metamodel of ITIL. In Barry, C., Lang, M., Wojtkowski, W., Conboy, K., Wojtkowski, G., eds.: Information Systems Development: Challenges in Practice, Theory, and Education. Volume 2. Springer (2009) 1081–1092

27. Valiente, M.C., Rodríguez, D., Vicente-Chicote, C.: Defining the Semantics of It Service Management Models using OWL and SWRL. In Filipe, J., Dietz, J.L.G., eds.: Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2010), October 25–28, Valencia, Spain, INSTICC Press (2010) 378–381
28. Valiente, M.C.: Improving IT Service Management using an Ontology-Based and Model-Driven Approach. PhD thesis, Universidad de Alcalá (2011)
29. IBM: IBM Tivoli Unified Process. http://www-01.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html, accessed on the 7th of February 2013 (2008)
30. The Stationery Office: ITIL UPDATE FAQs - October 2011. http://www.best-management-practice.com/gempdf/ITIL_Update_FAQs_Oct_2011.pdf, accessed on the 3rd of July 2013 (2011)
31. IT Governance Network: Summary of differences between CobiT 4.1 and CobiT 5. http://www.itgovernance.com/changes%20in%20cobit5.pdf, accessed on the 3rd of July 2013 (2011)
32. ISO/IEC: ISO/IEC 20000-1: Information technology – Service management – Part 1: Service management system requirements. (2011)
33. ISACA: COBIT 5 – A Business Framework for the Governance and Management of Enterprise IT. (2012)
34. Best Management Practice: ITIL – Service Design. The Stationery Office, Norwich (2011)
35. ISACA: COBIT 5 – Enabling Processes. (2012)
36. Sahibudin, S., Sharifi, M., Ayat, M.: Combining ITIL, COBIT and ISO/IEC 27002 in Order to Design a Comprehensive IT Framework in Organizations. In Al-Dabass, D., Turner, S., Tan, G., Abraham, A., eds.: Second Asia International Conference on Modeling Simulation (AICMS 08), May 13–15, Kuala Lumpur, Malaysia, IEEE (2008) 749–753
37. Scheer, A.W.: ARIS - vom Geschäftsprozeß zum Anwendungssystem. 3 edn. Springer, Berlin [u.a.] (1998)
38. Sharifi, M., Ayat, M., Rahman, A., Sahibudin, S.: Lessons learned in ITIL implementation failure. In Zaman, H.B., Sembok, T.M.T., van Rijsbergen, K., Zadeh, L., Bruza, P., Shih, T., eds.: International Symposium on Information Technology (ITSim 2008), August 26–29, Kuala Lumpur, Malaysia, IEEE (2008) 1–4
39. Barros, A., Oberle, D., eds.: Handbook of Service Description - USDL and Its Methods. Springer, New York (2012)

Enterprise Modeling in Complex ERP Adoptions: Learning from the Experience of an IT Company

Jan Trąbka[1], Piotr Soja[1]

[1] Cracow University of Economics, Department of Computer Science,
Rakowicka 27, 31-510 Kraków, Poland
{Jan.Trabka, Piotr.Soja}@uek.krakow.pl

Abstract. This study reports on the experience of enterprise modeling within the ERP (Enterprise Resource Planning) system implementation project in an IT company in Poland. The project was complex due to project-intensive company organization and resulting information requirements, comprehensive logistic and service processes, and the necessity of ERP integration with specialized service applications. The study seeks to analyze the role of enterprise process modeling during the initial phases of large-scale implementation projects. It discusses modeling process from the perspective of human resources, tools employed, and process organization. Conclusions highlight both mistakes and best practices observed in the modeling process. Main findings indicate that the strategic significance and risk of modeling process increase along the scale of company's activities and complexity of processes and environment.

Keywords: ERP, implementation, modeling, pre-implementation analysis.

1 Introduction

ERP system adoptions run the risk of failure which grows with the complexity of a company's business processes and scale of operations. Among critical success factors for this kind of projects, the significant roles are played by an adequate definition of requirements, project team experience, and involvement of the adopting company resources [4]. Considering model approaches to the ERP lifecycle, it appears that the pre-implementation analysis is the main stage which ends in the agreement and definition of the requirements for the target system [10, 14]. In consequence, we may conclude that a good pre-implementation analysis is a critical precondition for a successful ERP adoption project.

During pre-implementation analysis, the modeling of enterprise and its business processes is performed [1, 3]. The importance of these activities grows along the level of changeability of a company's economic setting. This particularly refers to transition economies, i.e. economies in transition from communist style central planning system to free market system [11]. The fast changing business environment in transition economies results in the necessity to treat enterprise modeling as a separate project with a separate contract and agreements [15].

The goal of this study is to report on the experiences in enterprise modeling and pre-implementation analysis performed within the ERP adoption project conducted in a company from IT industry in Poland, a transition economy. The focal company is characterized by a project-oriented management approach, complex internal processes, and extensive range of internal IT systems. This study provides details on the modeling process and discusses observed best practices and mistakes. This report concludes with the discussion of the effectiveness of the whole adoption project and possibilities of its improvement through good practices applied during the modeling and analysis stage.

2 The Case Company and ERP Implementation Background

A company from IT industry, named "IT Firm" in this report, is the focal organization in this study. A company providing implementation services in the considered ERP adoption project in named "ERP Supplier". IT Firm specializes in computer system integration and company activities include the following:

- **System integration** – IT Firm is a nationwide integrator of computer systems and its activities and services include analysis of customer needs and resources, systems design, pilot project implementations, final project implementations, acceptance tests, and post-implementation maintenance.
- **Building automation systems** – IT Firm offers a majority of currently available systems used in modern buildings and provides technical consultancy, integrated design, and project implementation and maintenance.
- **IT services and outsourcing** – implementation and maintenance of the ICT infrastructure of the company's clients, on a both standard and outsourced basis. The company has 18 branches scattered over the whole country and offers its services to companies and institutions operating in public administration, banking and financial sector, telecommunication, manufacturing, trade, and service. The company employs 400 people.

IT Firm is a project-driven organization and its activities are divided into projects conducted for the company's clients and governed by the signed contracts. The key information required by the company's management refers to the projects profitability, which requires the granular and multidimensional cost and profit accounting. For the purpose of this study we will call various dimensions of cost and profit accounting (i.e. projects, organizational units, product and service types etc.) controlling cross-sections. The data gathered within an organization (payroll, purchase and sale invoices, etc.) have to be classified (recorded) simultaneously into all controlling cross-sections that are important from the perspective of managerial analysis. The company needed the reporting mechanism able to present the aggregated data in multidimensional controlling cross-sections. IT Firm, before making a decision on the implementation of a new software solution, was using an ERP system which did not have any dedicated module for project management or analytical tool satisfying the requirements of a project-driven approach. As a result, improvement in project reporting became the first goal of the new system implementation.

The second goal included the optimization and integration of business processes within the whole organization. The most difficult area involved services, with a special focus on so called service logistics. The company employed a dedicated portal to manage service requests, accessible by both customers and company's employees from various departments such as service, logistics, and call center. Using service portal, a client registers its service requests and then is able to track their status. The service portal was not integrated with the ERP system which resulted in process discontinuity and excessive workload required in order to meet service deadlines. In consequence, optimization and integration of IT processes and systems became the second goal of the new system implementation.

3 Enterprise Modeling Process in IT Firm

3.1 Implementation Methodology

The implementation methodology, adopted in the project conducted in IT Firm with the support of ERP Supplier, is based on three pillars:

- international project management standard PRINCE 2 [2, 8],
- agile project management methodologies such as SCRUM [12, 13],
- flexible architecture of the system being implemented and the provider's extensive experience gained during a few hundred implementation projects in various industries.

The implementation methodology hinges upon three basic rules:

- phased approach to project planning and control,
- project tasks progress monitoring on the basis of project products,
- prototyping during the phase of user requirements implementation.

The whole implementation project cycle is depicted in Fig. 1. Next sections shortly describe two aspects of the methodology: main stages of the project run (Pre-implementation analysis and Requirements implementation) and project task verification rules.

- **Pre-implementation analysis** – involves specification of processes, with a map of top-level processes as a starting point for creating a hierarchical list of processes. Next, the processes are being decomposed into the elementary processes.
- **Requirements implementation** – is the longest stage of the project and is conducted together with trainings, which is imposed by the prototype approach. This approach involves creating prototypes of elementary processes defined at the analysis stage. The prototype is delivered to the Key Users for testing. Then, after introducing corrections to the prototype, repeated testing takes place and such an iteration repeats until the final acceptance of the prototype, which becomes part of the new final solution.

The project stages and methods of progress verification illustrate the key role of pre-implementation analysis, which should include appropriate definitions of processes being implemented in the new solution. Processes become project products that, in the next stages, form a framework for the project schedule and control mechanism.

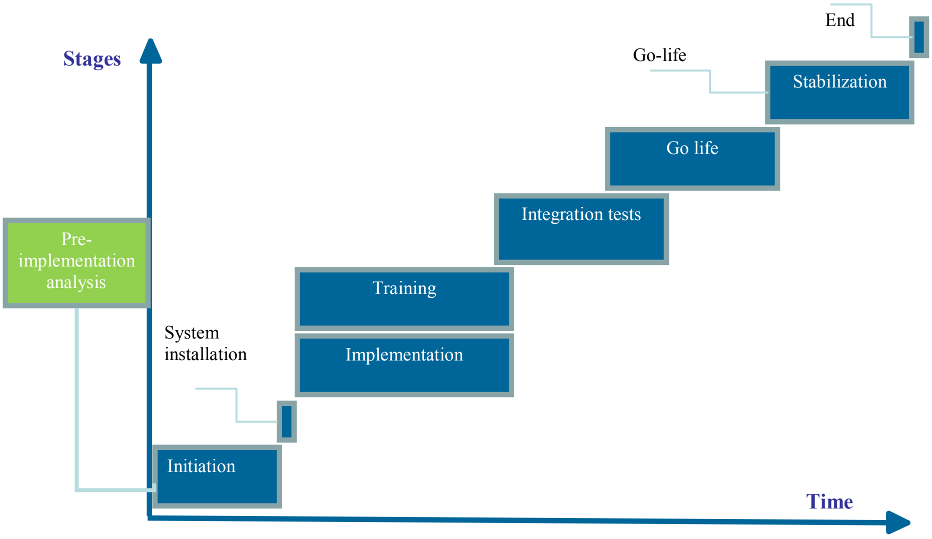


Fig. 1. Implementation project stages in IT Firm

3.2 Implementation Project Run

ERP system selection process in IT Firm started in 2006. In May 2007, a decision on enterprise system implementation had been made. The chosen system has been produced and implemented by the ERP Supplier. Two elements determined the system choice: (1) an extended project management module integrated with the other modules and (2) the overall system flexibility caused by its multi-layer architecture and a range of software tools enabling system customization. The general project schedule covered a one year time period with a productive start scheduled for January 1, 2008.

The general project schedule was divided into the following stages:

- Project preparation (PP)
- Analysis of business requirements (ABR)
- Implementation, divided into functional areas
  - ✓ Logistics and sales
    - System adaptation
    - System testing
    - Productive start
    - Stabilization
  - ✓ CRM (with similar sub phases as in the case of logistics and sales)
  - ✓ Finance and accounting, fixed assets, HRM

The analytical works started in June 2007. In practice, phases PP and ABR have been merged. During the first meeting, project managers from both sides agreed the rules of project team composition, methods of communication and control, and the schedule for the first month of analytical works. A project initiation document (PID) has been prepared, which was presented during the first meeting of the analytical team (kick-off meeting).

3.3 Analytical Team

The project team involved the following stakeholders:

- **Steering committee** – a body of top management representatives from both companies delegated to the project supervision. In the analyzed project, the committee has been lead by a vice-president of IT Firm, who also served as a project sponsor. The supplier's side was also represented by a person from top management – a director of operations in ERP Supplier.
- **Project manager** – was responsible for supervision and coordination of activities conducted by units involved in the project from the IT Firm's side. Project manager was responsible for communication in the project team and with the steering committee. Project manager, together with project coordinator, made operational decisions in the project. In the analyzed project, project manager role was played by a director of IT department in IT Firm.
- **Project coordinator** – was responsible for supervision and coordination of activities conducted by units involved in the project from the ERP Supplier's side. In the analyzed project, project coordinator role was played by a director of implementation department in ERP Supplier.
- **Key users** – a team of specialists from various areas of IT Firm involved in all stages of the project and responsible for verification of all solutions being implemented (project products). In the analyzed project, key users were recruited among managers of departments and teams working in the areas affected by the implementation project.
- **Key developers** – employees of ERP Supplier with broad implementation experience in individual functional areas. Responsible for creating and delivering project products. A team for analytical support was involved among ERP Supplier's representatives. Its members were responsible for implementation methodology, analytical tools, and documentation.

3.3 Analytical Tools

A document named Pre-implementation Analysis (PA) was the main product of the analysis stage. It was a model of information system in the organization managed with the help of the new IT system. In the analyzed case, PA also included organization- and project-related elements (e.g. schedule). The adopted approach to enterprise's information system was based on the structural analysis and design [16] where models of data and processes were the most important elements of the system. PA document was divided into the following parts: Analysis organization, Organizational characteristics, Map of processes with proposed solutions, Project organization, and Schedule.

Process model. Process model included Data Flow Diagrams (DFD) and process specification.

- Data Flow Diagrams (DFD) – depicts the system as a grid of information processes connected by data flows and data repositories. In the analyzed project, the modeling process started at the topmost level (level 0) which illustrates all main information processes of the organization (see Fig. 2). Next, each main process is decomposed and detailed until the elementary process.

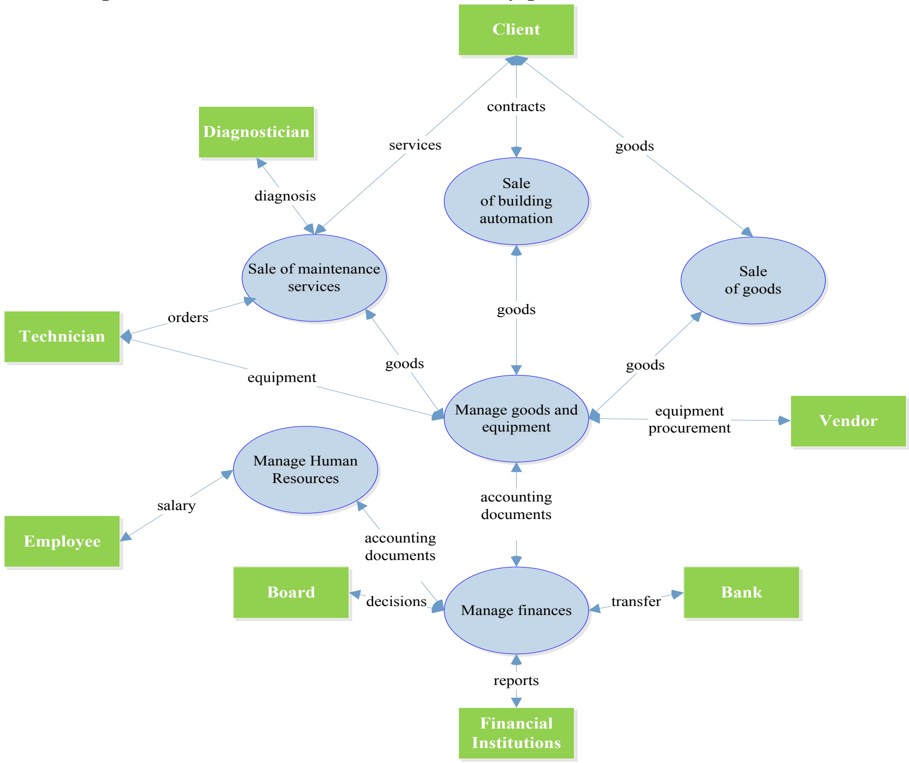


Fig. 2. IT Firm's DFD 0

- Process specification – each elementary process was defined with the help of a so called process card. Specifications were created in natural language; however, thanks to a formalized layout of the process card, they were unambiguous and precise. On the whole, IT Firm's model included 82 processes specified in this way.

Data model. The most important tools in data model included data dictionaries.

- Data dictionaries – included: element name, description, type, necessity, and default value. The field "description" was especially important as it defined and clarified the understanding of attributes in the analyzed organization.

3.4 Analysis Run

During the preparatory stage of the project, the project team was divided into domain teams created for the following areas: service, logistics, sales/CRM, finance, and IT. The adopted division turned out to be not adequate as, from the very beginning, people from service and logistics areas worked together. The schedule assumed meetings of teams at the same time and place so that mutual experience could be exchanged. However, in time, such a discipline disappeared and in consequence area teams worked according to their individual schedules and often in distant regions of the country.

Activities in the area of service and logistics. The team met on a regular basis and adopted the most detailed analytical perspective. The IT Firm's logistic team had a previous experience in business process modeling gained in past projects, when they used MS Visio and created process diagrams similar to BPMN notation [9]. This experience was very useful during the analysis and definition of company processes using DFD and process cards. The logistic team was very involved and motivated which resulted from the large scale of the team activity, as they had to handle a few thousand of service requests per month. In fact, a very difficult aspect of service-logistic activities was connected with the necessity of using two separate software tools. Customers, servicemen, call-center workers used the service portal for registering and handling requests. Logisticians, in turn, were handling materials and service equipment in the ERP system. In consequence of the analysis, the required interface was defined.

Activities in the area of finance. The financial team adopted an assumption that only processes relevant for the activities of financial department have to be modeled, such as cash register, bank transfers, and chart of accounts. The logistic team adopted an opposite assumption and presumed that the financial team is responsible for the definition of accounting schema and business rules binding logistic and accounting operations. In consequence, these connections were defined just during the project run, which was sometimes connected with changes in processes. Overall, the requirements overlooked origin of data and impact of other areas' requirements on controlling processes. It was assumed that members of other teams would handle controlling-related issues in their processes. Overall, processes were defined at a very general level and did not reach elementary level.

Activities in the area of sales/CRM. The sales/CRM team worked separately from other groups due to its distinct location. Before the project start, this area employed the largest number of nonintegrated software solutions, mainly desktop applications. In consequence, the vision of a uniform, integrated solution was very difficult to develop. The team did not put any effort into a detailed modeling of the client acquisition process and preparation of offers or contracts depending on a client's business background. The process definition was restricted to a text-based description linked to the abovementioned sales procedure and other official regulations. The data structure analysis nor process decomposition was not performed.

4 Discussion and Lessons Learned

The discussion of findings has been conducted using an approach similar to the perspectives employed in the previous section, i.e. human resources, tools, and analytical process run. We arrived at this decision considering the three main components of information systems: organization, management, and technology [5] and also drawing from product development and operations management area [6].

4.1 Human Resources

Project management staff empowerment. In the analyzed project, the empowerment of the chief of the steering committee, who was also the project sponsor, played a very significant role. The appointed person was a member of the company board, which assured access to company's resources. The project manager, who was an IT director, assured an effective project organization and good communication with the ERP Supplier's team. Nonetheless, the key users' empowerment raises doubts because, despite having adequate knowledge, they did not have a crucial influence on organizational changes or team members' availability.

Knowledge exchange between the analytical and implementation teams. The key determinant of the overall ERP implementation success is connected with transferring knowledge about the organization from the analysis stage to further stages. This transfer may refer to explicit knowledge (e.g. analytical documents) and tacit knowledge (e.g. gathered in the analysts' minds) [7]. The applied implementation methodology, proposed by ERP Supplier, assumed that the key developers were also leaders of the area teams. Such a solution turned out very beneficial during the later phases as the key developers started to get to know people, organization and their problems right from the beginning of the project.

Coordinating the effects of analytical works. The role of the project manager was to manage the organizational aspects of the project. She was not responsible for the quality of the final product, which was the PA document in the investigated case. Building on the experience of the analyzed project, the suggested recommendation is to empower the project manager with authority to control the final effect of conducted works.

4.2 Tools

Modeling organizational and system structure. The formal organizational structure was missing in the PA document. Such a structure is a basis for developing roles and user rights in the system. Skipping roles in process modeling prevents the analysts from discovering possible organizational responsibilities and interdependencies which may result from differences between allocated and actually performed organizational duties.

Process modeling. In the applied process modeling, the context level was missing, where the organization should be modeled as a black box with emphasis put on objects from the environment handled by the information system. The most convenient tool for illustrating and negotiating system behavior, used by many system analysis and design frameworks, are context diagrams [16]. In practice, lack of this perspective leads to overlooking major system stakeholders. In the analyzed case, missing context level resulted in lack of answer to a fundamental question: for whom the system is being built?

Modeling inter-system interfaces. Interfaces between information systems are usually difficult and risky elements; therefore, they should be carefully modeled during the analysis stage. In the investigated case, the modeling was restricted to the textual description what the resulting changes were in the service system when a particular process activity occurred in the ERP system. A data exchange mechanism useful for software developers was not modeled, although the system was supposed to work in an on-line mode. Building on the experiences of the analyzed study, the authors suggest to develop a prototype of a partial interface in order to verify if the project assumption were satisfied.

4.3 Analysis Run

Phased approach and budget. In the investigated project the pre-implementation analysis was a separated stage; however, its results could not have had impact on the project budget and time. The authors' suggestion therefore is to keep an implementation contract not signed until the analysis stage is finished, even with the possibility of canceling the whole project.

Domain-based analytical works. The division of the analytical team on the basis of business areas might not match the developed process model. It is difficult to prevent such a situation; however, it is beneficial to be aware that the initial division might be subject to change. Learning from the investigated project, it is suggested to delegate "inspectors" of the analysis integrity. In the discussed case, such inspectors might have been recruited from the controlling or project management departments.

5 Conclusion

This study investigated experiences of enterprise modeling gained during the complex implementation of an ERP system in a company operating in IT industry in Poland. Such projects bear significant risk of failure which increases with growing complexity of a company's business processes and scale of operations. The performed analysis indicates that the risk of failure is inversely proportional to the quality of a developed enterprise model and this relationship is influenced not only by technical factors, but also by human-related and organizational elements. The investigated case illustrates that the following factors had the most significant influence on the modeling quality:

- too general level of process definition,
- unclear definition of interfaces between the ERP system and legacy systems,
- lack of consistency in the application of the adopted methodology,
- lack of supervision over the whole modeling process.

In general, this study's findings suggest that the properly conducted pre-implementation analysis is a very significant instrument in minimizing risk of implementation project failure. Therefore, increased resources invested in a high quality analysis are strategically justified and should pay off.

References

1. Berio, G., Vernadat, F.: Enterprise modelling with CIMOSA: functional and organizational aspects. Production Planning & Control, 12(2), 128--136 (2001)
2. Bradley, K.: Understanding Prince 2. Spoce Project Management Limited, Dorset (1997)
3. Dalal, N.P., Kamath, M., Kolarik, W.J., Sivaraman, E.: Toward an Integrated Framework for Modeling Enterprise Processes. Communications of the ACM, 47(3), 83--87 (2004)
4. Finney, S., Corbett, M.: ERP implementation: A compilation and analysis of critical success factors. Business Process Management Journal, 13(3), 329--347 (2007)
5. Laudon, K., Laudon, J.: Essentials of Management Information Systems. 10th Edition, Prentice Hall, Upper Saddle River, New Jersey (2012)
6. Morgan, J.M., Liker, J.K.: The Toyota Product Development System: Integrating People, Process and Technology. Productivity Press, New York, USA (2006)
7. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, Inc., New York (1995)
8. OGC: Managing Successful Projects with PRINCE2® 2009 Edition. The Stationery Office, London (2009)
9. Panagacos, T.: The Ultimate Guide to Business Process Management: Everything you need to know and how to apply it to your organization. Theodore Panagacos (2012)
10. Ross, J.W., Vitale, M.R.: The ERP Revolution: Surviving vs. Thriving. Information Systems Frontiers, 2(2), 233--241 (2000)
11. Roztocki, N., Weistroffer, H.R.: From the special issue editors: Information technology in transition economies. Information Systems Management, 28(3), 188--191 (2011)
12. Schwaber, K.: Agile project management with Scrum. Microsoft Press, Washington, USA (2004)
13. Schwaber, K., Sutherland, J.: Scrum Guide, Scrum Org, http://www.scrum.org/Scrum-Guides (2011)
14. Somers, T., Nelson, K.: A taxonomy of players and activities across the ERP project life cycle. Information & Management, 41, 257--278 (2004)
15. Themistocleous, M., Soja, P., Cunha, P.R.: The Same, but Different: Enterprise Systems Adoption Lifecycles in Transition Economies. Information Systems Management, 28(3), 223--239 (2011)
16. Yourdon, E.: Modern Structured Analysis. Prentice Hall (1989)

Methodology of Building ALM Platform for Software Product Organizations

Ivo Pekšēns

AS Itella Information Mūkusalas 41b, Rīga, LV-1004 Latvija
ivo.peksens@itella.com

Abstract. This work investigates Application Lifecycle Management (ALM) and elaborates a methodology for building an ALM platform for organizations dealing with manufacturing of software products. The meaning of platform is defined and available ALM platforms on the market are analyzed as a part of the methodology execution. Emphasis is put on basic principles coming from PLM – Product Lifecycle Management which are about integration of different parts and roles in an organization with the purpose of better information exchange that positively impacts business agility, performance and visibility.

Keywords: Application Lifecycle Management, Software products, Tool integration.

1 Introduction

Application Lifecycle Management (ALM) is a term coined in various sources, (examples [1], [2], [3]) around Information Technology (IT) industry during the last decade, but no two definitions are the same. Often the scope of ALM is narrowed down to software development activities only, including maintenance at best. A more detailed investigation (examples [4], [5], [6]) shows that ALM is very similar to PLM – Product Lifecycle Management which covers much broader scope – from idea till end of life of the product. The question is – if software is a product why cannot ALM cover the same scope as described by PLM, especially taking into account the maturity of the latter? (Term PLM is rooted in manufacturing and has been widely used for years describing the whole lifecycle of the physical/hardware products).

The idea behind PLM is to solve the problem of un-integrated work of different roles and parts of an organization that collaborate on product throughout its lifecycle [7]. The main elements of the solution are product information flow, visibility and availability that make the work integrated and effective which is crucial for product innovation in today’s fast pace business world.

The ALM platform in short encompasses all the technical means that enable the above stated qualities of the information throughout the software product lifecycle. While it is possible to build such a platform from existing tools and systems, market offers out-of-box ALM solutions [8], [9], [10], [11] specifically addressing the software product lifecycle management. Vendors like IBM, Microsoft, Hewlett Packard and others are the key players in this market niche.

Organizations participating in software business usually have some set of tools in place therefore obtaining an out-of-box ALM platform is a huge responsibility and requires a thorough fit-gap analysis prior to making the decision.

This article describes the elaborated methodology for building the ALM platform based on current state of an organization producing software products. At the end, execution (partial) of the methodology with fit-gap analysis method to four ALM platforms available on the market is presented.

2 Structure

The elaborated methodology consists of two parts:

- ALM Readiness check – describes/defines target organization's product lifecycle against ALM Reference model;
- ALM Tool investigation – investigates tools that support the described product lifecycle and evaluates them accordingly to ALM Reference Requirements (derived from the ALM Reference model).

Figure 1 illustrates the mapping of the methodology foundation (ALM Reference model and ALM Reference requirements) to the product lifecycle and its tools.

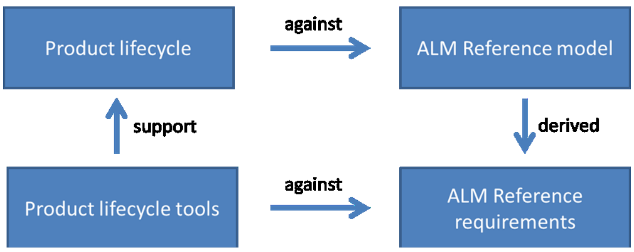


Fig. 1. The mapping of the methodology foundation to the product lifecycle.

2.1 ALM Reference model

In order to build the ALM platform, reference points are needed. Investigation of the information about IT industry shows that there is a lack of independent, objective guides/methods for ALM type of solutions. Variety of sources [12], [13], [14], [15], [16], [17] has been researched and none of it describes the whole lifecycle of the software product in a clear, concise way. For this reason originally constructed ALM Reference model for building ALM platform is presented that defines and integrates ideas from various approaches into one common model that covers the whole lifecycle of a software product very similar to how PLM does [7]. See Figure 2.

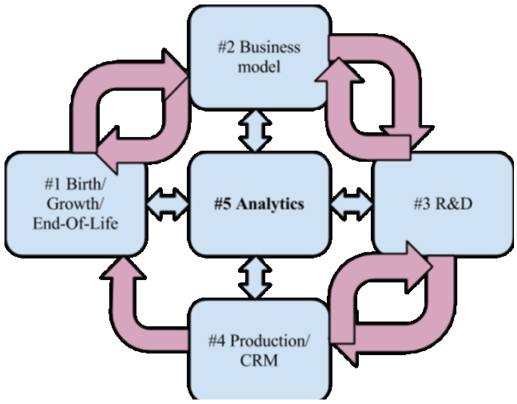


Fig. 2. ALM Reference model.

Birth/Growth/End-Of-Life block is either a starting, improvement or end-of-life point for the product. Depending on organization, this may be a small set of few separate products reaching to huge program and product portfolio management processes that are targeted in this block. This block maintains product's full information that is updated on frequent basis.

Business model block is the place where product's feasibility is validated before the actual product development is started. It is included in the model as extension to classical project management. It is proposed to function according to innovative way of creating business models as proposed by A. Osterwalder [18]. Its main benefit is in being a visual one-pager style look [19] at the project/product/idea from many perspectives which leads to much more precise validation of financial information.

R&D block represents Research & Development phase of the model which includes finding the right technology and running software development according to some of known software development approaches [20].

Production/CRM block is the last phase which mainly includes processes like Release and Customer Relationship Management (CRM). Challenges in here are release strategy choice, actual release management, customer feedback incorporation into development and similar. The key idea of this block's relation with block 1 is getting the real production statistical data and CRM data as input into block 1 for correct and timely decision making about the fate of the existing products (termination, further development, new product).

Analytics is the core of this model as it is meant as continuously analyzing part of processes and information in other blocks. It can be divided into internal and external parts. Internal is any type of analytics solution up until enterprise level business intelligence implementations [21], whereas external is interface to something as Big Data[1].

[1] From http://visual.ly/big-data: "Big Data is data that is too large, complex and dynamic for any conventional data tools to capture, store, manage and analyze. The right use of Big Data allows analysts to spot trends and gives niche insights that help create value and innovation much faster than conventional methods"

Main feature of the model is *Integration* which makes all parts of it work as one system. This way it achieves high agility and visibility of information under processing.

2.2 ALM Reference requirements

In order to execute partially the methodology with fit-gap analysis method, twenty nine ALM Reference requirements are derived from the ALM Reference model that represent the model's blocks in more details. See Appendix: ALM Reference requirements[2].

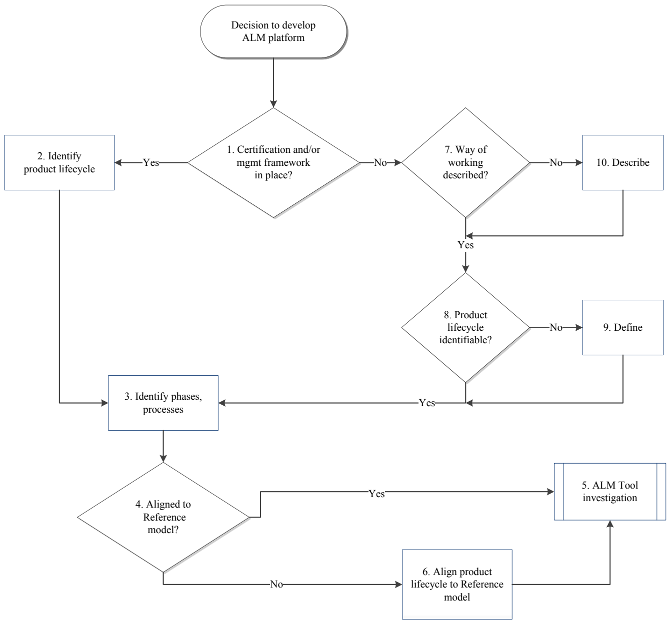


Fig. 3. ALM Readiness check flowchart – first part of the elaborated methodology.

3 ALM Readiness check

Figure 3 shows the first part of the methodology as introduced in Section 2. Its goal is to verify if the target organization that wants to build an ALM platform is ready for this undertaking. The ALM Readiness check identifies and reveals the software product lifecycle inside the organization and if necessary aligns it with the defined ALM Reference model. The result of this check is a clear description of the product lifecycle, for example, in a widely used swim lane format.

[2] Some of requirements are not shown due to space limitations.

4 ALM Tool investigation

The second part of the methodology – ALM Tool investigation is based on the fact that all processes need tools in order to be executed; therefore it deals with the investigation of the technical means, which are mainly software application tools that support the execution of identified software product lifecycle. See Figure 4.

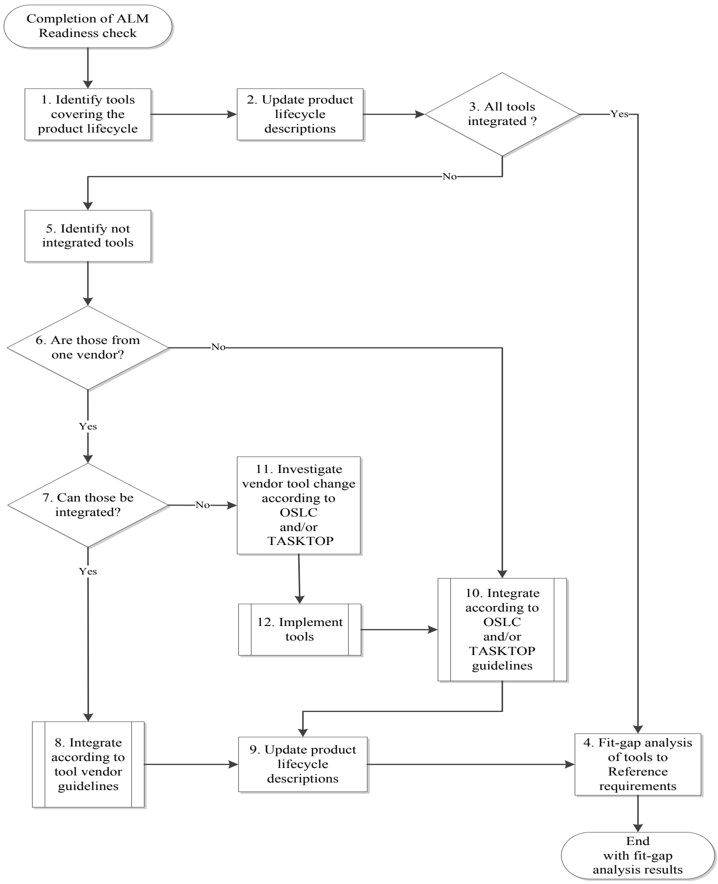


Fig. 4. ALM Tool investigation flowchart – second part of the elaborated methodology.

As it can be seen (Figure 4), Activities 10 and 11 references to OSLC[3] and Tasktop[4]. The reason for that is their knowledge in integration of diverse set of vendors/tools

[3] http://open-services.net/ - Open Services Lifecycle Collaboration - An open community building practical specifications for integrating software.

that produce software product manufacturing tools and applications for use by IT industry players. Emphasis is on the fact that ALM platform might be either one complete solution from one vendor like, for instance, IBM or it can consist of various tools made by various vendors. The key thing is to make them all work integrated, so they can be called the ALM platform. ALM Reference requirements representing the ALM Reference model can be considered as an instrument to verify the conformity of the tools that form the ALM platform to the proposed ALM Reference model.

5 ALM platform analysis

For execution (partial) of the elaborated methodology, test organization is introduced that has passed the first part of the methodology and has one vendor ALM platform in-house. This way the second part of the methodology gets executed that deals with the tools. Four participants were chosen according to Gartner [12] and Forrester [13] recommendations and those are IBM Rational, Microsoft Visual Studio ALM, Rally Software ALM and Atlassian based ALM ([8], [9], [10], [11]). Analysis approach is based on information investigation which is available mainly on the vendor websites. Neither real world testing, nor customer feedback survey was used. Fit-gap analysis method was used with the twenty nine requirements as described in section 2.2. Tables 1 – 7 show ALM platform analysis scores.

Table 1. ALM platform overall analysis score.

	Fit	Partial fit	Gap
IBM Rational	12	8	9
Microsoft Visual Studio ALM	11	6	12
Rally ALM	11	7	11
Atlassian ALM	5	7	17

Table 2. ALM platform Birth/Growth/End-Of-Life block analysis score.

	Fit	Partial fit	Gap
IBM Rational	1	1	1
Microsoft Visual Studio ALM	0	2	1
Rally ALM	2	1	0
Atlassian ALM	0	1	2

[4] http://tasktop.com/ - Commercial organization, specializing in software development tool integration.

Table 3. ALM platform Business model block analysis score.

	Fit	Partial fit	Gap
IBM Rational	0	1	2
Microsoft Visual Studio ALM	0	0	3
Rally ALM	3	0	0
Atlassian ALM	0	0	3

Table 4. ALM platform R&D block analysis score.

	Fit	Partial fit	Gap
IBM Rational	10	3	1
Microsoft Visual Studio ALM	10	2	2
Rally ALM	5	4	5
Atlassian ALM	4	4	6

Table 5. ALM platform Production/CRM block analysis score.

	Fit	Partial fit	Gap
IBM Rational	1	2	2
Microsoft Visual Studio ALM	1	0	4
Rally ALM	1	1	3
Atlassian ALM	1	1	3

Table 6. ALM platform Analytics block analysis score.

	Fit	Partial fit	Gap
IBM Rational	0	1	3
Microsoft Visual Studio ALM	0	2	2
Rally ALM	0	1	3
Atlassian ALM	0	1	3

Table 7. ALM platform integration feature analysis score.

	Fit	Partial fit	Gap
IBM Rational	1	3	0
Microsoft Visual Studio ALM	1	0	3
Rally ALM	2	2	0
Atlassian ALM	0	2	2

Overall (Table 1) the highest score goes to IBM Rational and none of the platforms completely conforms to the proposed ALM Reference model. Tables 2 – 5 show scores for other blocks. Integration feature (Table 7) the highest score goes to Rally Software ALM.

6 Conclusions

Emphasis was put on looking at software products similarly as it is done in other areas, for example, car manufacturing. The reasons for this are to underline that the making process of software products is very much alike. Integration of tools and information flow becomes very crucial in this case. The elaborated methodology for building the ALM platform is model based, the latter being introduced also as part of the work. The reason for this is shortage of such models as shown by the research. Although this is an original work, the model itself is put together from already available information about the subject of software based products. The methodology itself consists of two parts – product lifecycle definition and lifecycle tool investigation. As our work shows, in majority of cases, ALM platforms are built on top of existing tools not obtained as out-of-box solutions. Nevertheless, testing the model and reference requirements on ALM platforms available on the market allowed us to conclude that none of those conform the model for 100%. The reasons behind this might still be very diverse interpretation of what is ALM by different vendors resulting into delivery of appropriate solutions as well as shortage of solid, industry accepted knowledge about ALM the same like it is about PLM.

References

1. Chappel D.: What is application lifecycle management?, http://www.davidchappell.com/WhatIsALM--Chappell.pdf/
2. Rossman B.: Application Lifecycle Management, pp 218, Tebbo (2010)
3. Woody L.S.: Where the acronym ALM does come from?, http://leonardwoody.com/2011/07/18/where-does-the-acronym-alm-application-lifecycle-management-come-from/ (2011)
4. BigLever software, Inc: PLM, ALM and PLE, http://www.biglever.com/technotes/plm_alm_spl.html
5. Klassen M.: PLM and ALM – strange but necessary bedfellows - http://www.eclipsecon.org/2013/sessions/plm-and-alm-%E2%80%93-strange-necessary-bedfellows, The Eclipse Foundation, 2013 – 27 p.
6. Serignese K. SD Times – Organization works to blend application, product lifecycle management - - http://www.sdtimes.com/link/34381, 2010.
7. PLM Technology Guide, http://plmtechnologyguide.com/site/?page_id=435/
8. IBM. Collaborative Lifecycle Management, http://www-01.ibm.com/software/rational/alm/collaborate/
9. Microsoft. Visual Studio ALM - http://www.microsoft.com/visualstudio/eng/alm
10. Rally Software. Rally ALM platform - http://www.rallydev.com/platform-products/deliver-features-users-love-faster.
11. Atlassian. - http://www.atlassian.com/software.

12. Gartner. Selection Criteria for Success in Choosing ALM Products - http://www.gartner.com/id=1839116/
13. Gartner. Magic quadrant 2012 - http://www.gartner.com/technology/reprints.do?id=1-1AX7YJM&ct=120614&st=sb
14. Forrester. Wave - http://www.forrester.com/pimages/rws/reprints/document/60080/oid/1-LMW1C9.
15. Ovum. Independent, objective analysis - http://ovum.com/section/home/
16. Qvum. Decision matrix on selecting ALM - http://download.microsoft.com/download/7/A/0/7A007614-A916-4A7C-ADC1-1B1EBC1F9A57/Report%20-%20Ovum%20Decision%20Matrix%20on%20Selecting%20ALM.pdf
17. TechExcel. ALM Tool vendor comparison - http://techexcel.com/comparison-of-alm-tools/.
18. Osterwalder A., Business model innovation: Wiley; 1 edition, 2012 – 288 p.
19. Business Model Foundry - http://businessmodelgeneration.com/.
20. A federal government website. Selecting a development approach - http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf
21. Decision Sciences Institute. Business Analytics: The Next Frontier for Decision Sciences - http://www.decisionsciences.org/DecisionLine/Vol43/43_2/dsi-dl43_2_feature.asp

Appendix: ALM Reference requirements

1	**ALM Reference model: Birth/Growth/End-Of-Life**	
1.1	Product profile	There must be support for seeing general product information - name, status (idea, development, maintenance, ending), versions, lifespan and similar. Multiple products (portfolio) support is mandatory.
1.2	Product backlog	There must be a centralized storage of product backlog. It must be possible to filter it, based on various criteria (proposed, approved, denied, etc.). Important attribute of each entry is its financial value.
1.3	Product roadmap	There must be a way of describing product roadmap and product releases with names, dates and possibly other information.
2	**ALM Reference model: Business model**	
2.1	Idea validation	It must be possible to validate idea of a new product. Approach like “Business model generation” can be used. This feature must allow modeling business value.
2.2	Release validation	It must be possible to validate any new release of an existing product similar to 2.1 and it also must be linked. This feature must allow modeling business value.

3	**ALM Reference model: R&D**	
3.1	Release backlog	Depending on the method used, here must be a possibility to create subset of requirements from the product backlog according to product roadmap.
3.2	Waterfall methodology support	It must be possible to run development projects according to waterfall method.
3.4	Iterative methodologies support	It must be possible to run development projects according to iterative methods.
3.5	UML and other diagramming for requirements definition	It must be possible to complement requirements with diagrams in different notations.
3.6	Source control	Various types of source control must be supported by the platform.
3.7	Programming language independence	This part of platform must also be vendor independent - thus allowing performing work in variety of technologies (.NET, C/C++, Java) or it must be easily extensible to accommodate different frameworks.
4	**ALM Reference model: Production/CRM**	
4.1	Release management	There must be possibility to manage several releases of the product. It must be clearly visible in which state is each release. Besides, generation of release notes must be possible as well.
4.2	Production deployment	It must be possible to deploy the finished release into the production environment or mark the release as ready to manufacturing (RTM) in case of shippable/embeddable product.
4.4	CRM	It must be possible to use the same environment that is used for R&D also for handling support requests as well as perform CRM activities. This part of ALM platform must be integrated with ERP system of the enterprise or any other that holds customer/partner data.
5	**ALM Reference model: Analytics**	
5.1	Connections	Analysis block of the solution must take data from other blocks around it. If there is no built-in Analytics sub-system, that supports integrated work of blocks 1-4, it must declare options on connectivity with other analytic solutions/enterprise systems.
5.2	Social media monitoring	Analysis block must be social networking friendly. It

		must be possible to search in public social media data according to keywords. Example of such solution is “Hoot Suite”
5.3	Web data monitoring	It must be possible monitor and search information in public web according to keywords.
5.4	Existing data warehouses	There must be possibility to connect and make use of existing data marts, databases and other sources of data for analysis purposes.

Experiences from Teaching Enterprise Modelling to Students of Information Systems

Sobah Abbas Petersen[1,2] and John Krogstie[2]

[1] SINTEF Technology & Society, Norway
sobah.petersen@sintef.no
[2] Norwegian University of Science & Technology, Norway
krogstie@idi.ntnu.no

Abstract. In this paper, we describe how Masters students at a university studying Information Systems adapt to Enterprise Modelling. The paper describes an overview of the course and its design and the feedback from the students based on a questionnaire at the end of the course. The feedback indicates that the students found it helpful to be able to select their own cases for their modelling assignment, that they were able to relate their modelling assignment to the real world and the theory taught in the course and that their confidence in modelling improved during the course.

Keywords: Enterprise Modelling, Model Evaluation, Modelling Theory, Modelling Practice, Aspects of modelling.

1 Introduction

The goal of an enterprise model is to support analysis of an enterprise and to model the relevant business processes and how they relate to the rest of the enterprise such as how the work is organized and resource utilisation. Enterprise models and modelling have been defined by several authors; e.g. Fox and Gruninger describes an enterprise model as a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints of a business, government, or other enterprise [1]; Bernus defines Enterprise Modelling as a collective name for the use of models in Enterprise Engineering and Enterprise Integration [2]; and Vernadat defines Enterprise Modelling as a consistent set of special purpose and complementary models describing various facets of an enterprise to satisfy some purpose of some business users [3]. All these definitions suggest that Enterprise Modelling involves modelling several aspects of an enterprise and how they relate to one another. In particular bringing in the business perspective together with the IT perspective is an important role of Enterprise Models.

This paper considers the practice of Enterprise Modelling where theory and knowledge are applied in a practical manner to make sense for an enterprise. This is important to ensure that the knowledge is applied in an appropriate way to gain optimal results from the modelling.

As part of our teaching and research, we take an Action Research [4] approach to investigate the best way to teach Enterprise Modelling to Information Systems students. A traditional cycle of planning, action and reflection has been considered. Our aim has been to improve the curriculum and teaching approach and practice by reflecting upon the previous years' courses and making improvements for the next year. Our focus has been on how the students applied the theory that they had learned to create Enterprise Models and how they ensured that their models served their intended purposes. We believe that in order to teach modelling to university students, they have to practice modelling as a part of their education.

In this paper, we describe how Masters students at a university studying Information Systems (IS) adapt to Enterprise Modelling. The paper will present our experience from three years of teaching Enterprise Modelling. We discuss the design of a course for teaching Masters students studying Information Systems to gain knowledge and acquire the competences necessary to do Enterprise Modelling in business situations. In particular, we discuss the design of the course which is based on theory and practical aspects and engaging students by encouraging them to model their own cases. The main aim of this paper is to validate our teaching approach and to obtain feedback from the students to improve our teaching practice and to identify best practices for teaching Enterprise Modelling.

Feedback from the students has been obtained using a questionnaire at the end of the course. The results of the questionnaire are presented and discussed in this paper. (The paper [5] presented results from 2011 and 2012. Note that a preliminary analysis of the data from 2013 was presented at EMMSAD 2013, but is not provided in the paper.) The rest of this paper is organized as follows: Section 2 provides a background of teaching Enterprise Modelling; Section 3 describes the Enterprise Modelling course on which the paper is based upon; Section 4 describes the observations from earlier years of teaching; Section 5 describe some examples of models created by the students, Section 6 presents and discusses the feedback from the questionnaire and Section 7 summarises the paper.

2 Background

Enterprise modelling requires an understanding of the enterprise or business situation that is modelled. Thus the course aims to bridge the students' understanding of how IT supports business and other situations and the role that IT plays in organizations. The enterprise perspective takes into account the business requirements that will determine the investments to be made by the enterprises to deliver their products and services. This may require IS support for the enterprise to be able to perform their processes effectively and to meet their customers' needs. These may pose requirements on the IS support, thus driving the IS needs for an enterprise. This is illustrated in Fig. 1.

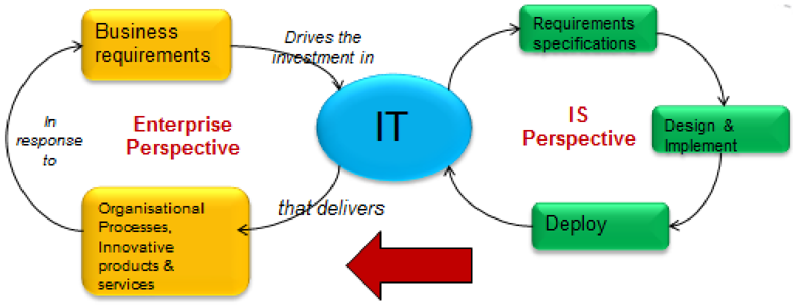


Fig. 1. Enterprise and IS perspectives

Enterprise modelling courses are offered at a few other universities too. For example, at University of Duisberg-Essen in Germany[1], the course on Enterprise Modelling teaches specific modelling methods such as ER and UML and includes presentation by students. A course offered at Jonkoping University[2] in Sweden includes practical hands-on modelling seminars in groups, group modelling sessions, and review seminars of the group assignment. The course reported in this paper is a Masters and PhD course in the Dept. of Computer and Information Science at NTNU. At this stage of their studies, the students have had courses in programming, computer science topics, databases, software engineering, potentially including model-driven software engineering, and information systems, and have experience in modeling with UML and BPMN.

3 Course Design

The main aim of the course is to prepare our IS students to be ready to do Enterprise Modelling when they finish their Masters course. It is a Masters course attended by both Masters and PhD students for one semester. During 2011 and 2012, the course had 15 participants per year. In 2013, 21 students took the course, 3 of whom were PhD students. The students are mainly from the Dept. of Computer and Information Science who had already taken courses in IS-related subjects, which focused on several modelling techniques such as UML and BPMN. A few students are international Masters students and a few are from other faculties such as Mechanical Engineering. The course consists of lectures spread over 11 weeks, and a mandatory modelling assignment. Attending the lectures was optional.

The course aimed at bridging the theory and practice of Enterprise Modelling and the design of the course are influenced by this. Some of the modelling techniques that

[1](http://www.wi-inf.uni-duisburg-essen.de/FGFrank/index.php?lang=en&groupId=1&contentType=Course&generalModuleNumber=test2)

[2] https://hj.se/sitevision/proxy/jth/student/...html/.../TVMD28.pdf

were taught in the course include i* [6] and IDEF0[3]. The course is based on two main activities:

1. The theory part of the course which consisted of traditional lectures. The curriculum is based on a collection of articles. The main topic included modelling methods for stakeholder and the early phases on requirements modelling, Product Modelling, Process Modelling, Enterprise Modelling and Enterprise Architecture. The lectures were given by the course teacher and two guest lectures were given by others.
2. The practical part of the course which consisted of an individual modelling assignment where the students had to analyse a situation, create a model, present it to the class and write a report.

In this paper we focus on the practical part of the course consisting of the modelling assignment. The approach taken was that students learn Enterprise Modelling by applying the theory to practice. The practical part of the course is designed to support learning by doing [7] and reflection [8]. Following Kolb's experiential learning cycle, the students have a concrete experience by creating their models.

Enterprise modelling requires an understanding of the enterprise or business situation that is modelled. In practice, Enterprise modelling is conducted as a team including the modelling expert, who conducts the modelling and the domain expert, who has in-depth knowledge and experience on the situation or the modelling domain [9]. Enterprise modelling involves externalisation of knowledge, sometime knowledge that is tacit. Therefore, modelling experts and facilitators require experience supporting the externalisation of tacit knowledge [10] from the domain experts. In the course, we didn't have any domain experts. Since the students worked individually on their assignments, they had to act as the domain experts. The students were asked to select their own case for the modelling assignment for a number of reasons; they needed a case for which they could act as the domain expert as well as the modelling expert, to ensure a close affiliation between the modeller and the knowledge and ownership of the knowledge to support their learning. To find a realistic case for the students, it was decided that it is best for the students to find their own case that was meaningful for them.

The students were required to present their models to the class 3 weeks before they had to submit the final model and the report. This part of the assignment supports the reflection part of Kolb's experiential learning cycle [7] where the students reflect on what they have modeled as well as make an attempt to articulate their models in a manner that is understandable for the audience. The students then receive feedback from their peers as well as the teacher. Through this, the students also learn providing constructive criticism.

The students were required to use the Metis Enterprise Modelling environment[4], which provides a visual space and metamodels for creating Enterprise Models. Metis was introduced at the beginning of the course and time was allocated during every lecture to provide modelling support to the students as required. Specific requirements were set for the assignment; they were required to describe their cases in detail, to

---

[3] http://www.idef.com/IDEF0.htm

[4] A product of Troux Technologies.

identify the users and stakeholders of the model, to describe the purpose of the model and to use this to define how they will evaluate their model to ensure that the model fulfilled its purpose, to create a model that included at least five aspects of an enterprise (e.g. processes, organisation, applications), use the functionalities provided by Metis to selectively view the contents of the model (e.g. user-specific views of the model), to evaluate their model and to describe the lessons learned from the modelling experience. Metis allows users full-fledged meta-modeling to introduce new modelling concepts and notation to the metamodel, i.e. enhance the modelling language; thus the students were asked to describe how they had enhanced their metamodel wherever appropriate. This was to assess if they were actually capable of appropriately represent a real-world situation as an enterprise model.

The students were graded based on the combination of the modelling assignment and a written exam, where the modelling assignment counted 35% and the written exam counted 65%.

4 An Example Model

In 2013, two thirds of the students chose their own case, while the others chose the suggested case of developing a mobile app for language learning. Although the latter case was suggested by the teacher, all the students had adapted their cases according to their own understanding. The students who had chosen their own case brought ideas from their own experiences and had identified a need for modeling the case. One student modeled the roles and responsibilities for Abakus Linjeforening, the organization set up by Computer Science students to arrange various activities for themselves. This organisation is structured as eight committees, each with their roles and responsibilities. One of the challenges experienced by them is ensuring that their new members, several each year, are able to quickly get to know how Abakus works and their roles and responsibilities and who they should contact for whatever they need. Thus, a model was created to show these explicitly and one that could be use to educate the new members of Abakus. The model shows the different committees, their leaders and goals and the processes that are connected to them. The complete model is shown in Fig. 2.

Another student modelled an online booking system for a physical therapy clinic to design one to simplify the booking process for the patients and to decrease the secretary's workload, while another modelled the processes in a retail chain of stores that he used to work in to see where the problems are and in the hope to raise the awareness of the management. All the students found their models useful for their situations and sometimes identified issues that they had not foreseen prior to the modelling exercise. Almost all the models were of sufficient complexity, addressing at least five domains of the enterprise (e.g. process, goals, organisation, people and documents) and how they related to one another. The students displayed good knowledge of the domain and had clear purposes for their models. They were able to use various viewing capabilities such as selected views and relationship matrices to use their models. An automatically generated relationship matrix from selected contents in the model, of the committees in the Abakus organisation and the people that lead each committee is shown in Fig. 3.

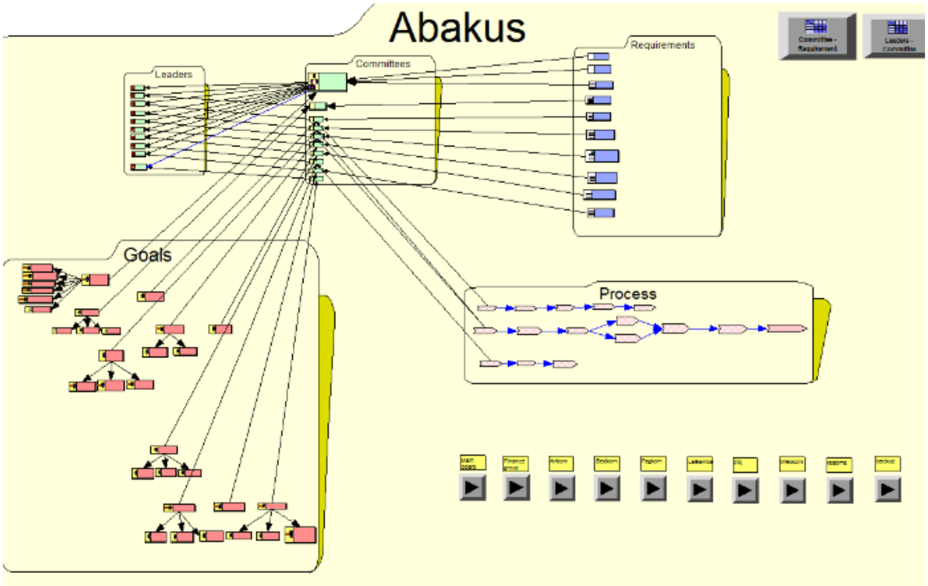


Fig. 2. Example model: Overview of Abakus

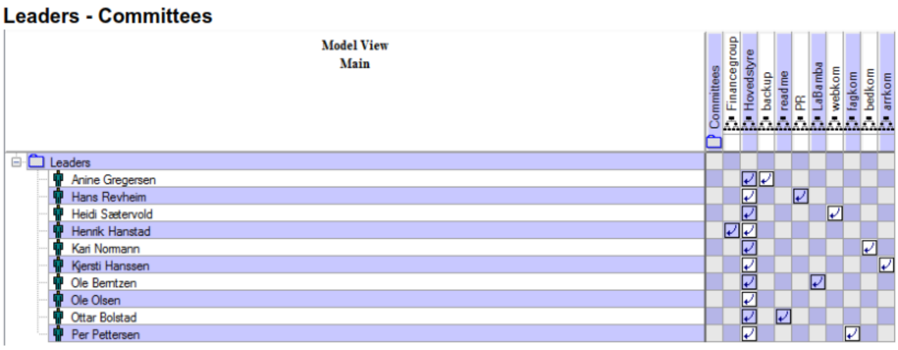


Fig. 3. Example model - a relationship matrix for the Abakus model

5 Feedback from Students

In 2013, we have focused on validating our approach to the design of the course, in particular, the practical part of the course which includes the modelling assignment. Feedback from students was obtained through a questionnaire presented to them at the end of the course. 20 students submitted assignments, 18 took exams, 13 responded to the questionnaire (72% of students that took the exam).

The questionnaire was developed using the SurveyMonkey tool and made available to the students online. The questionnaire was developed to obtain students' feedback on the following:

- General view of the assignment.
- Assignment design and presentation.
- Ability to relate the practical work with theory and the real world.
- Confidence in modelling.

6.1 General view of the assignment

The students were asked: Did you find this course useful? On a scale of 1-3, where 1 is "Not useful at all", 2 is "useful" and 3 is "Very useful", 8 (62%) responded that was useful and 5 (38% responded that it was very useful. Some of the comments provided by the students in the open part of the questionnaire are: "*The course gave me insight in the world of modelling. There were a bit more to it than I first thought, so this course has kind of "opened my eyes*"", "*Gave me more insight in how to model enterprises*" and "*The course considers "the bigger picture", and you learn to think about how (initiality) seperate domains are related. I found this very useful*".

6.2 Assignment design and presentation

The design of the assignment was validated by asking the following questions:

- Did you find the composition of practical work and lectures suitable?
- Did you find it helpful that you were able to select your own case for the modelling assignment?

When asked: Did you find the composition of practical work and lectures suitable?, on a scale of 1-3, where 1 is "Not suitable at all", 2 is "Suitable" and 3 is "Very suitable", 10 (77%) responded that it was suitable and 3 (23%) responded that is was very suitable. The students found it helpful that they could select their own case to model; 6 students (46%) responded that it was helpful and very helpful while only 1 student (8%) responded that it was not helpful at all.

The students provided their views in the open part of the question: some were very positive, e.g. "*Selecting my own case, let me select something I was motivated to model*", "*This made the assignment awesome in my opinion:) I could select something that I was already a domain expert in. Then I could just keep focus on the modelling task*". There were some students that found it a bit challenging or various reasons, e.g. "*It was hard to come up with good ideas about the case, when I did not have anything from the real world I felt I could use*" and "*I think this has both advantages and drawbacks. I felt that selecting my own case ensures that it's a case I find interesting and feel motivated to model. But at the same time it was hard to make sure that it applied to what we learnt and understand....*".

The presentation was validated by asking the following questions:

- Did the presentation clarify things for you?
- Did you receive useful feedback from the class that helped you improve your model?
- Did you receive useful feedback from the teacher that helped you improve your model?

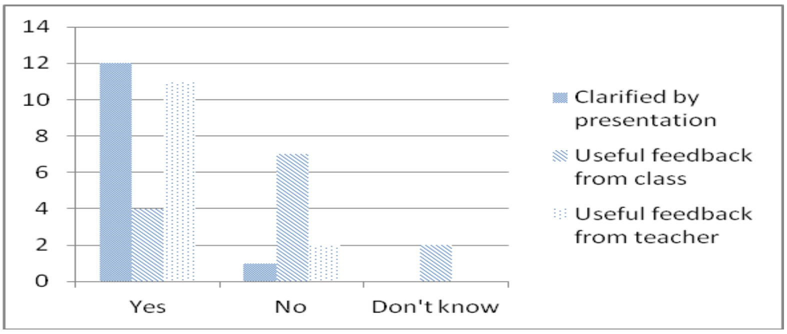


Fig. 4. Clarification through presentation

An overview of the responses to the above questions is presented in Fig. 4. 92% of the students responded yes when they were asked if the presentation clarified things for them. 84.6% of the student agreed that the feedback they received from the teacher was helpful while 15.4% disagreed with that. 30.77% agreed that the feedback they received from their peer students was helpful while 53.84% disagreed with that and 15.4% said that they did not know. One of the aims of the presentations were also to encourage peer reviews among students and to learn to understand others' models as well as to be able to present one's own models. However, there was very little feedback provided by the students during the presentations.

6.3 Relating the practical work with theory and the real world

Validation of the relation between the practical work and the real world and theory were done by asking the following questions:

- How well do you think you were able to relate your case to the real world?
- How well do you think you were able to relate your case to the theory taught during the course?

An overview of the results when asked how well they were able to relate the case that they had chosen to model to the real world and the theory that was taught in the course is shown in Fig. 5. 9 students (69%) responded that they were able to relate their cases to the real world while 4 students (31%) responded that they were able to relate their cases to the real world very well. In the open part of the question, one of the students responded with the following comment: "*It is highly related to the real world such that I am thinking of trying out my model in my work later on :)*".

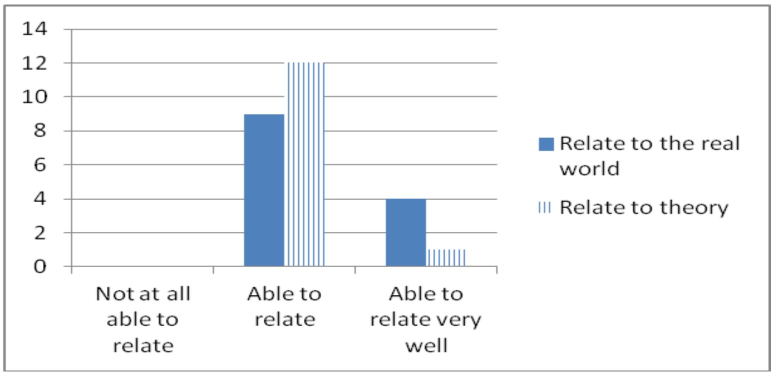


Fig. 5. Ability to relate modelling cases to the real world and theory

When asked how well they were able to relate their cases to the theory that was taught in the course, 12 students (92%) responded that they were able to relate while 1 student (8%) responded that s/he was able to relate the case to the theory very well. None of the students responded that they were not at all able to relate their cases to the real world or the theory. Some of the students comments in the open part of the question are: "*Quite a few of the theory lacks the fundamental "why" question. In other words, why do we need this modeling language is not answered…..*" and "*As often is the case, the real world has many more "grey" areas than the theory teaches, but absolutely saw the link between theory and practice*".

6.4 Confidence in modelling

The students were asked the following question with an open response:

- How confident do you feel in using modelling as an approach in your future work related to software design and development?

Three of the students explicitly stated that they were "very" and "pretty" confident of modeling and three stated explicitly that they will continue modeling and would use it in their future work. Five of the students expressed that they had learnt from the modelling assignment and would know how to create an enterprise model. Three students expressed explicitly that they were not confident in modeling after this course. Some of the comments from the students include: "*I think It was good to learn and I am searching for the next field to apply this knowledge rather than storing this knowledge as garbage in brain*“, "*I would continue this work in my future work*", "*Not too confident, but I do believe with more experience that I should be able to create good models*" and "*Not very much more confident than before the course*".

6 Summary

In this paper, we describe how Masters students at a university studying Information Systems adapt to Enterprise Modelling. The paper describes an overview of the course and the rationale for the design of the course. A questionnaire was used to obtain feedback from the students and the results of the questionnaire are presented. The course was designed to ensure that the students gained practice in Enterprise Modelling and to act as a bridge between the IS courses and the business perspectives of enterprise modeling.

The results of the questionnaire show that the students found the course useful and the ratio of theory and practical parts of the course was suitable. The course included a presentation of the modeling assignment which the students found helped them clarify their cases and how to model an enterprise. In particular, almost all the students found that it was helpful that they had the possibility to choose their own cases for modeling. The students also responded that they were able to relate the practical part of the course to the real world as well as the theory taught. The students responded that they had gained confidence in modeling and some indicated that they would do modeling in the future.

Based on this feedback from the students, we aim to continue improving our course design and teaching approach. In particular, we aim to conduct further studies to understand the process the students follow during their modelling work and explore modelling in a team.

References

1. Fox, M.S. and M. Gruninger, Enterprise Modeling. AI Magazine, 1998. 19(3): p. 109-121.
2. Bernus, P. and L. Nemes. Organisational Design: Dynamically Creating and Sustaining Integrated Virtual Enterprises. in IFAC World Congress, Vol-A 1999. London: Elsevier.
3. Vernadat, F.B., Enterprise Modelling and Integration Principles and Applications. 1996: Chapman and Hall.
4. Water-Adams, S. Action Research in Education. 2006 [cited 2013 29 August]; Available from: http://www.edu.plymouth.ac.uk/resined/actionresearch/arhome.htm.
5. Petersen, S.A. and J. Krogstie, The World out there: from Systems Modelling to Enterprise Modelling, in BPMDS 2013 and EMMSAD 2013, B. et.al., Editor. 2013, Springer-Verlag: Berlin Heidelberg. p. 456-465.
6. Yu, E.S.K., Towards Modelling and Reasoning Support for Early-phase Requirements Engineering in 3rd IEEE Int. Symp. on Requirements Engineering (RE'97). 1997: Washington D.C., USA. p. 226-235.
7. Kolb, D., A., Experiential Learning: Experience as the Source of Learning and Development 1984: Beverley Hills: Sage Publications.
8. Boud, D., R. Keogh, and D. Walker, Reflection: Turning Experience into Learning. 1985: PutledgeFalmer, Taylor and Francis Group.
9. Persson, A. and J. Stirna, Towards Defining a Competence Profile for the Enterprise Modelling Practitioner, in PoEM 2010, P. van Bommel, Editor. 2010, Springer. p. 232-245.
10. Nonaka, I. and H. Takeuchi, The Knowledge-Creating Company. How Japanese Companies create the Dynamics of Innovation. 1995, New York: Oxford University Press.

Do Conceptual Modeling Languages Accommodate Enough Explicit Conceptual Distinctions?

Dirk van der Linden[1,2,3] and Hend*erik* A. Proper[1,2,3]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
dirk.vanderlinden@tudor.lu
[2] Radboud University Nijmegen, Nijmegen, the Netherlands
[3] EE-Team, Luxembourg, Luxembourg*
e.proper@acm.org

Abstract. In this paper we are concerned with the degree to which modeling languages explicitly accommodate conceptual distinctions. Such distinctions refer to the precision and nuance with which a given modeling concept in a language can be interpreted (e.g., can an actor be a human, an abstraction, or a collection of things). We start by elaborating on the notion of conceptual distinctions, while also providing a list of common modeling concepts and related distinctions that are relevant to enterprise modeling. Based on this, we will then analyze a number of conceptual modeling languages to see whether they accommodate the explicit modeling of (potentially important) conceptual distinctions – that is, whether they have specific language elements to model conceptually distinct entities with. We conclude by discussing what impact our findings may have on the use (and creation) of modeling languages.

Keywords: enterprise modeling, modeling languages, conceptual distinction, conceptual understanding

1 Introduction

Most concepts common to conceptual modeling languages and methods (e.g., goal, process, resource, actor, etc.) can be interpreted in a number of conceptually distinct, yet equally valid, ways. For example, in the context of business processes, one may choose to interpret actors as being human beings who take decisions and execute actions. At the same time, however, interpreting them as being abstract agents or dedicated pieces of hardware might be equally valid in another context. One could also choose to interpret actors as being a collection of things that, together, execute some actions (e.g., an organizational department composed of many employees, a cluster of computers) instead of being a single thing executing an act. Depending on the context of the domain to be

* The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University, the University of Luxembourg and HAN University of Applied Sciences (www.ee-team.eu)

modeled, the stakeholders and other modelers we interact with, and the goal of the model itself, we often choose among the different possible interpretations. These different interpretations of the same concept can lead to a host of semantic considerations. For example, if an actor is a human being, one can never be as sure that s/he will behave as expected compared to, say, a computer.

It is important that such different interpretations can be modeled distinctly. It would not do well for the overall clarity and semantic quality of a model if we conflate semantically different interpretations (e.g., human beings, abstract entities and material objects) under the same banner (e.g., 'actor') and pretend that they are one and the same thing. Yet, this is often the case with modeling languages. Frequently, the designers of a modeling language define a type (e.g., actor) and allow it to be instantiated with a wide diversity of entities (humans, hardware, abstract and mathematical entities) which have no common ontological basis. Sometimes modeling languages do accommodate (some) of these conceptual distinctions, but then do so only implicitly. That is, in their specification or meta-model they assume a particular interpretation. As such, all instantiations of a model are then implicitly assumed to abide by that interpretation (e.g., all actors in the given model are assumed to be human things, all goals are assumed to be hard goals). An example of a language doing so is the i* specification as found in the Aachen wiki [10], which defines agents (the acting entities) as having "*a concrete physical manifestation*". This implicitly makes it semantically incorrect to use abstractions (e.g., agents as they are commonly understood) and furthermore, perhaps ontologically incorrect to use composite agents – market segments – as the composition itself is not physically manifested.

It is more useful if a modeling language accommodates such conceptual distinctions *explicitly*, to the extent needed in relation to its expected and planned use. That is, instead of relying on the underlying semantics to define every concept they allow (or perhaps require), to use a notation that explicitly encodes information about our interpretation – and do so by providing distinct notational elements for all the important different conceptual distinctions. This can mean for instance, having exclusive (visual) elements to represent such distinct concepts by (e.g., the amount of 'stick puppets' in in ArchiMate actor type denoting whether it is a single actor or a collection of them). This is important from a cognitive point of view as it improves the quality of the notation by ensuring there is no notational homonymy. These points (and more) were argued for by e.g., Moody in his work on a general "physics of notation" [14]. Several modeling languages have been analyzed to estimate their cognitive quality in terms of this framework (e.g. i* [15], BPMN [7], UCM [6], and UML [13]). However, most of these analyses are aimed at the semantics of the (visual) syntax, and forego a more detailed analyses of the semantics of the individual elements of meaning themselves. By this we mean that they analyzed the semantic quality of the formalization of the syntax (i.e., which elements interoperate in what way), but spent less attention to the question what the elements arranged by this syntax actually means to the users of the language (e.g., what *is* this element called 'agent', what thing does it really represent). From a quality perspective, important related issues are *semiotic clarity* (one-to-one correspondence between semantic constructs and graphical symbols) and *perceptual discriminability* (symbols should be clearly distinguishable) [14].

Thus, in this paper we will specifically look at the cognitive quality of a number of modeling languages and methods in terms of the semiotic clarity of their semantic constructs. These constructs can be both visual (for visual notations) and textual (for textual notations), but both require a proper correspondence between semantic constructs and symbols used for them. To do so we will provide an initial (likely non-exhaustive) overview of different aspects of enterprises that are explicitly modeled today, and show to what degree relevant conceptual distinctions can be explicitly modeled in the languages and methods used for them. The goal of this work is not to provide detailed individual analyses of all the languages involved, but to explore whether there is a trend in modeling languages to support enough distinctions or not, and provide advice on basis of that for modeling language use and design in general.

2 Aspects of Enterprises and Associated Languages

Enterprises are large socio-technical systems encompassing many aspects (e.g., business processes, value exchanges, capabilities, IT artifacts, motivations, goals), which themselves are often the domain of specialized (groups of) people. As these models are produced by different people, often using different languages, integration is a vital step in order to have a coherent picture of the enterprise [11]. Ensuring that different conceptual distinctions are modeled explicitly is thus especially important in this context, as much information can be lost in this integration step, leading to enterprise models that are no longer correct or complete in regards to the semantics intended to be expressed (and possibly only done so implicitly) in the models made of each of the distinct aspects. Traditionally processes and goals received a lot of attention in terms of explicit models and dedicated modeling languages and frameworks, while recently more and more aspects are being considered equally as important to deal with. Other aspects such as motivations and goals, value exchanges, deployment and decision making now have dedicated, often formally specified, modeling languages available. This increases the amount of languages (ideally) capable of explicitly supporting conceptual distinctions important to the individual aspects that are in use, but perhaps at the cost of fragmenting the modeling landscape itself. Table 1 gives a brief overview of some current languages and the aspects they are, or can be used for.

3 Conceptual Distinctions for Aspects & Languages

The different aspects that are focused on in enterprise modeling, typically have a number of (not necessarily overlapping) specific conceptual distinctions, which are important to be aware of. For example, a motivational model describing the things to be achieved by an enterprise and the reasons for wanting to achieve

Table 1: A cross-section of aspects of modern enterprises and some modeling languages used, or usable to represent them.

Aspect of an Enterprise	Related languages
Architecture	ArchiMate [24] (1.0, 2.0), ISO/DIS 19440, ARIS
(Business) Processes	BPMN [17], (colored) Petri nets, IDEF3, EPC [1]
Design decision-making	EA Anamnesis [19], NID [5], OMG DMN (proposed, seemingly unfinished)
Deployment of IT artifacts	ADeL [18]
Goals & Motivations	i*, GRL, KAOS [2], TROPOS [8], AMORE [21], ArchiMate [24] 2.0's motivational extension, OMG BMM [16]
Management of IT artifacts	ITML [4]
Strategy & Capability Maps	TBIM [3], OMG BMM [16], Capability Maps [22]
Value exchanges	e3Value [9], REA-DSL [23], VDML (under development)

them is likely to require more detail (and thus fine-grained conceptual distinctions) for what goals are than, say, a model describing the related process structure. Such distinctions can be for instance whether goals absolutely have to be achieved, whether the 'victory' conditions for achieving it are known, whether the goal itself is a physical thing to be attained or not, and so on. On the other hand, a model describing the process undertaken to achieve a certain goal (e.g., bake a pizza) might require conceptual distinctions like whether the actors involved are human entities or not, whether it is one or more actors responsible for ensuring the goal's satisfaction, and so on. Thus, not all conceptual distinctions that are relevant to one aspect (and the modeling language used for them) will be as relevant (and necessary to model explicitly) for other aspects. In order to systematically talk about whether the selected modeling languages accommodate different conceptual distinctions we need both a set of common modeling concepts and a set of distinctions to analyze. We base ourselves on an analysis of modeling languages and methods commonly used in (enterprise) modeling as reported on in [12].

Table 2: This table gives an overview of a number of relevant conceptual distinctions for common modeling. For each of the concepts, we list relevant conceptual distinctions, show what they are useful for, and what languages support modeling them explicitly, might support it, or (where relevant) make a specific implicit interpretation.

Dimension	Useful to ...	Supported by ...
	ACTOR	

human	Distinguish between actors that can be more fickle than pure rational agents.	BPMN through the explicit use of a 'Human Performer' resource type, VDML does contain a 'Person' subtype of Actor which is specified to be human, but does not distinguish in the visual notation between types of Actors.
composed	Distinguish whether an actual entity acts or whether a group of them does, which impacts responsibility judgments for actions	ArchiMate, TROPOS via 'composite Actor', somewhat as well with differentiation between 'role' and 'position', e3Value somewhat through differentiation between actor and market segments, VDML distinguishes between an 'actor' being a singular participant, and modeling 'collaboration' or 'participant' as potentially multiples.
material	Know whether an actor physically interacts with the world (and can thus be affected by it directly – think hardware vs. software)	i* assumes that an agent is an actor "*with a concrete physical manifestation*" (iStar Wiki)
intentional	Know whether an actor is considered an explicit part of a system, i.e., is expected to act or not on certain things, in contrast to actors from outside the systems scope which may act but were not regarded or thought of to do so	Implicit in most languages, mentioned as such in TBIM, depending on interpretation could be argued to be explicit in OMG BMM with differentiation between internal and external influencer.
specific	Knowing whether an actor is a specific thing (i.e., an instantiation) or a general thing (i.e., a role)	Supported by some (e.g., ArchiMate), through type/instantiation dichotomy, explicit in TBIM by the claim that an agent "*represents a concrete organization or person*" ArchiMate, implicit in e.g., e3Value and RBAC by automatic use of roles (types).
	EVENT	
intentional	Distinguish between events that should, or will happen given a set of circumstances, and events that happen (seemingly) unprovoked.	Arguably explicitly supported by BPMN through the use of 'None' type triggers for Start Events.

	GOAL	
composed	Distinguish between complexity level of goals, i.e., whether they are an overarching strategy or directly needed goals.	TBIM explicitly models composite goals as 'business plan' types, implicit in some other languages focused on strategy/tactics (e.g., OMG BMM).
material	Distinguish between objects and their representations, i.e., is the goal to achieve an increment in the integer on a bank account, or to hold an n amount of currency.	
necessary	Distinguish between goals that have to be attained and those that should.	
specific	Distinguish between goals for which the victory conditions are known and not, i.e., hard vs. soft goals.	Most goal modeling languages/methods/frameworks (e.g., i*, GRL, KAOS, AMORE) support this explicitly.
	PROCESS	
composed	Distinguish between black (closed, singular) and white (open, composed) boxes.	Arguable either way for BPMN with the use of pools, which can function as black boxes, however, those do not allow for linking sequence flow to it, and are thus self-contained.
intentional	Know whether they are part of an intended strategy or something that has to be dealt with (i.e., negative environmental processes)	
specific	Know whether the structure is (supposed to be) clear (deterministic) or not (fuzzy).	
	RESOURCE	
natural	Know whether a resource requires a 'fabrication' process.	Somewhat related, TBIM explicitly models resource types as being either animate or not.
human	Know whether resources can act on their own and produce issues, e.g.., be unreliable, not always generate the same outcomes	
material	Distinguish between objects and their representations, i.e., whether a given resource a collection of paper and ink blobs or the information contained within them.	Explicit in ITML through the use of hardware/software dichotomy.

	RESTRICTION	
natural	Distinguish between restrictions we cannot do anything about and those we can.	
intentional	Distinguish between restrictions we stipulate from those that arise holistically (whether good or bad).	Some languages implicit, e.g., EA Anamnesis, and BPMN through use of 'Potential Owner'.
necessary	Distinguish restrictions that can be broken from those that cannot.	(supported by some GPML, e.g., ORM 2.0).
specific	Distinguish restrictions for which we know when they are broken and not.	
	RESULT	
natural	Know whether a result requires some kind of 'fabrication' process	
material	Distinguish between an object and its representation, i.e. whether the physical pizza or the status update in the IS saying a pizza was baked is the result of a given step in the pizza making process.	
specific	Know whether a result is (supposed to be) clear (deterministic) or not (fuzzy).	Arguably supported in BPMN through the use of 'None' type End Events.

4 Discussion

Around half of the conceptual distinctions we analyzed were explicitly supported by at least one modeling language, with some cases being arguable either way. Languages used for specific aspects do seem to explicitly accommodate some basic (and often widely accepted) necessary conceptual distinctions. For example, the de facto used language for process modeling, BPMN, has explicit support for differentiating between human and non-human actors, which can be important to know for critical steps in a process. Most modeling languages used for motivations and goals also accommodate the distinction between goals with well-specified victory conditions and those with vague or unknown conditions by means of separate hard and soft-goal elements. These explicit distinctions in the notation are likely correlated with the conceptual distinctions being widely accepted as important and having become part of the basic way of thinking. However, taken overall, there does not seem to be a consistent or systematic pattern behind what language explicitly accommodates (or lacks) which conceptual distinctions.

As such, there are a number of conceptual distinctions for which we found no explicit support by any modeling languages. For example, we found no support for explicitly modeling goals and results as being material things. It also did not seem possible to explicitly model goals as being a logical necessity in the investigated languages. The distinction whether results were things that naturally occurred or fabricated was also not supported. When it comes to processes we found no support to model them explicitly as being intentional, and distinguishing between specific (i.e., well-defined) processes and processes more fuzzy in their structure. Modeling resources as being humans was also not supported, while this is likely not an unthinkable interpretation – effective management of 'human resources' being important for large enterprises. Finally, we found no explicit support for modeling restrictions as naturally occurring and specific things. We will discuss some of these distinctions in more detail.

4.1 Some unaccommodated conceptual distinctions

Surprisingly, we found no explicit support for differentiating between goals with varying levels of necessity and obligation. While many common methodologies (e.g., the MoSCoW technique of dividing requirements into must, should could, and would haves) call for such distinctions, many modeling languages conflate them all into a single kind of goal. Arguably in certain aspects it would make sense to make an implicit choice, as in e.g., process modeling it is necessary for certain steps in a flow to be reached before the flow continues, which can be seen as an analog to logically necessary goals. However, goal models in dedicated languages seem to not make this distinction, even though there is a strong focus on differentiating between hard and soft-goals, which seem correlated with different levels of necessity (e.g., one cannot as certainly rely on a soft-goal to be achieved compared to a hard-goal, especially for mission critical goals).

Another seemingly unaccommodated distinction is the necessity of restrictions, that is, whether some restriction (e.g., a rule, principle, guideline) is an alethic condition that cannot be broken or whether it is not and thus can be broken. While in the context of enterprise modeling there is a strong differentiation of terminology used for different kinds of normative restrictions that can be considered breakable, or at least not strictly enforceable (e.g., principles, guidelines, best practice), these often seem to be used outside of modeling languages in their own approaches – e.g., architecture principles [20]. It seems problematic that many languages used for aspects of enterprises, and languages used to describe the actual enterprise architecture like ArchiMate do not have explicit notational support for these different kinds of restrictions. Many models that are analyzed a posteriori (e.g., when they are integrated in other models, and the original modelers are no longer involved or available) then become difficult to interpret, as the notation of different kinds of restrictions can be ambiguous and lead to situations where it is not clear whether a restriction can be relaxed or not. Surprisingly the only language that seems to support this conceptual distinction is ORM (in particular version 2), which supports the explicit modeling of restrictions as being either alethic or deontic conditions through its visual notation.

Thus, it seems necessary to stimulate a move towards more explicit focus on (formalization of) the semantics of the elements of meaning of modeling languages. The lack of coverage for some of the distinctions shown in Table 2 makes it clear that more work on extending the specification of relevant languages with the ability to explicitly distinguish between these different conceptual understandings. Given the existence of a large number of different dialects of modeling languages sometimes only differing slightly (e.g., i*, GRL, TROPOS for goal modeling), it seems that supporting many different conceptual distinctions in a single notation would be welcomed by many.

5 Conclusion and Future Work

We have discussed the importance of explicitly modeling conceptual distinctions and analyzed a number of modeling languages to investigate what kind of distinctions they support. We showed that, while some conceptual distinctions are explicitly supported by relevant modeling languages, there are still a large amount of potentially relevant distinctions that are not accommodated, or implicitly interpreted in a specific way by modeling languages. We proposed that research should be done regularly to keep up to date with conceptual distinctions deemed relevant and important by modelers and stakeholders alike. Our future work will involve investigations into which distinctions are deemed important.

Acknowledgements. This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (`www.fnr.lu`), via the PEARL programme.

References

1. van der Aalst, W.M.: Formalization and verification of event-driven process chains. Information and Software technology 41(10), 639–650 (1999)
2. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program. 20, 3–50 (April 1993)
3. Francesconi, F., Dalpiaz, F., Mylopoulos, J.: Tbim: A language for modeling and reasoning about business plans. Tech. Rep. DISI-13-020, University of Trento. Department of Information Engineering and Computer Science (May 2013)
4. 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 DSM (2009)
5. Gal, Y., Pfeffer, A.: A language for modeling agents' decision making processes in games. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems. pp. 265–272. ACM (2003)
6. Genon, N., Amyot, D., Heymans, P.: Analysing the cognitive effectiveness of the ucm visual notation. In: Kraemer, F.A., Herrmann, P. (eds.) System Analysis and Modeling: About Models, Lecture Notes in Computer Science, vol. 6598, pp. 221–240. Springer Berlin Heidelberg (2011)

7. Genon, N., Heymans, P., Amyot, D.: Analysing the cognitive effectiveness of the bpmn 2.0 visual notation. In: Malloy, B., Staab, S., Brand, M. (eds.) Software Language Engineering, Lecture Notes in Computer Science, vol. 6563, pp. 377–396. Springer Berlin Heidelberg (2011)
8. Giunchiglia, F., Mylopoulos, J., Perini, A.: The tropos software development methodology: processes, models and diagrams. In: Agent-Oriented Software Engineering III, pp. 162–173. Springer (2003)
9. Gordijn, J., Akkermans, J.: Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements engineering 8(2), 114–134 (2003)
10. Grau, G., Horkoff, J., Yu, E., Abdulhadi, S.: i* guide 3.0. Internet (August 2007), http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=67
11. Lankhorst, M.M.: Enterprise architecture modelling–the issue of integration. Advanced Engineering Informatics 18(4), 205 – 216 (2004)
12. van der Linden, D.J.T., Hoppenbrouwers, S.J.B.A., Lartseva, A., Proper, H.A.: Towards an investigation of the conceptual landscape of enterprise architecture. In: T. Halpin et al. (ed.) Enterprise, Business-Process and Information Systems Modeling, LNCS, vol. 81, pp. 526–535. Springer Berlin Heidelberg (2011)
13. Moody, D., Hillegersberg, J.: Evaluating the visual syntax of uml: An analysis of the cognitive effectiveness of the uml family of diagrams. Lecture Notes in Computer Science 5452, 16–34 (2009), cited By (since 1996) 0
14. 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)
15. Moody, D., Heymans, P., Matuleviaius, R.: Visual syntax does matter: Improving the cognitive effectiveness of the i* visual notation. Requirements Engineering 15(2), 141–175 (2010), cited By (since 1996) 0
16. Object Management Group: Business motivation model (bmm), version 1.1. Internet (2010), http://www.omg.org/spec/BMM/1.1/
17. Object Management Group: Business Process Model and Notation (BPMN) FTF Beta 1 for Version 2.0. Internet (2010), http://www.omg.org/spec/UML/2.0/
18. Patig, S.: Modeling deployment of enterprise applications. In: Proc. CAISE Forum, LNBIP 72. pp. 253–256 (2010)
19. Plataniotis, G., de Kinderen, S., Proper, H.A.: EA Anamnesis: Towards an approach for Enterprise Architecture rationalization. In: Printing, S. (ed.) Proceedings of the The 12th Workshop on Domain-Specific Modeling (DSM12). ACM DL (2012)
20. Proper, H.A., Greefhorst, D.: The Roles of Principles in Enterprise Architecture. In: Proper, H.A., Lankhorst, M.M., Schoenherr, M., Barjis, J., Overbeek, S. (eds.) Trends in Enterprise Architecture Research. Lecture Notes in Business Information Processing, vol. 70, pp. 57–70. Springer Berlin Heidelberg (2010)
21. Quartel, D., Engelsman, W., Jonkers, H., Van Sinderen, M.: A goal-oriented requirements modelling language for enterprise architecture. In: Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International. pp. 3–13. IEEE (2009)
22. Scott, J.: Business Capability Maps – The missing link between business strategy and IT action. Architecture & Governance 5(9), 1–4 (2009)
23. Sonnenberg, C., Huemer, C., Hofreiter, B., Mayrhofer, D., Braccini, A.: The rea-dsl: a domain specific modeling language for business models. In: Advanced Information Systems Engineering. pp. 252–266. Springer (2011)
24. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)

Granular Ontologies and Graphical In-Place Querying

Janis Barzdins, Edgars Rencis and Agris Sostaks

Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia
{Janis.Barzdins, Edgars.Rencis, Agris.Sostaks}@lumii.lv

Abstract. The data ontologies in a form of UML Class diagrams are discussed in this paper. We call the data ontology granular, if its corresponding instance diagrams (data) can be divided into separate parts called slices. A typical example of granular ontologies is process ontologies, where slices are run-time instances of these processes. Based on the notion of granularity a graphical in-place query language is presented in this paper. The proposed language is easy to use by domain experts that are not IT specialists. Besides that it has a very efficient (linear) execution time for answering queries.

Keywords: Data ontologies, run-time instances, query language, graphical querying.

1 Introduction

While working with models, we have observed an interesting phenomenon – data can be often divided into separate parts naturally. These parts have their own semantics, which we would like to use while querying the model. If the division of the data ontology is well formalized, it is possible to develop a query language for the ontology that is both very efficiently executable and very convenient and easy-to-use for the end user being the domain expert, not an IT specialist.

In this paper we specify a set of ontologies, for which we can define a natural division in parts. We call these ontologies granular and define the granularity principles very formally in Section 2. After that, we describe the query language in Section 3, which we have developed for granular ontologies. Here we lay out the principles and primitives of the language, as well as define its time-efficiency.

We, however, understand that not all the real world ontologies fall into the class of granular ontologies. Therefore, in future we plan to extend the notion of ontology granularity a bit in order to widen the class of granular ontologies by extending the query language at the same time. The main objective that needs to be taken in mind in the process is that the query language must preserve its time-efficiency.

2 Granular Ontologies

In this paper we inspect data ontologies in a form of UML Class Diagrams. More precisely, we use only a subset of UML Class Diagrams containing classes, oriented associations, typed attributes and generalization.

From syntactic point of view our data ontology language is also a subset of the OWL (see the comparison in [1]). From the semantic point of view there is, however, a significant difference – while OWL uses the open-world semantics, we exploit the closed-world semantics. Our proposed data ontology language is a convenient mean for describing data of concrete domains, e.g., the structure of hospital registry. A very simple example of a data ontology describing study programs is seen in Fig. 1 (we use a traditional shorthand notation for associations, which are oriented in both directions).

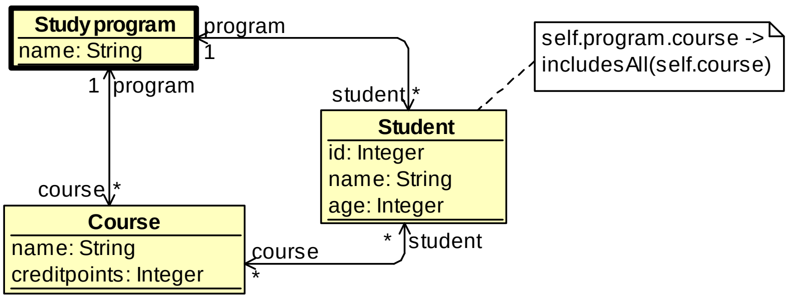


Fig. 1. The Study program ontology.

We will depict the concrete data of the ontology as *legal* instances of the corresponding class diagram. The specification of legality can be performed either only through multiplicities (which must always be satisfied), or additionally through OCL expressions (as in Fig. 1) or in any other way (even using the natural language).

Let us assume we have a data ontology in a form of UML Class Diagram $\mathcal{D}$ and a class A belonging to the ontology $\mathcal{D}$. Let us also assume we have some instance $\mathcal{G}$ of the diagram $\mathcal{D}$. $\mathcal{G}$ consists of two kinds of elements – class instances called *objects* and association instances called *links*. Since we only operate with oriented associations, also the links are oriented. Therefore we can perceive $\mathcal{G}$ also as an oriented graph. Let us now take an arbitrary instance x of the class A such that $x \in \mathcal{G}$ (in shorthand notation: $x \in A \cap \mathcal{G}$). We can now introduce a concept of a *slice respective to the object* x within instance $\mathcal{G}$ being the maximal subgraph of $\mathcal{G}$ (let us denote it with $S(x,\mathcal{G})$) such that $S(x,\mathcal{G})$ consists of the object x and all those objects that are reachable from x via edges.

When we inspect some data ontology $\mathcal{D}$, it always comes together with the set of its legal instances $\mathcal{U}_{\mathcal{D}}$. We will now call a class $A \in \mathcal{D}$ a *Master class*, iff the two following statements are satisfied:

1) $\forall \mathcal{G} \in \mathcal{U}_{\mathcal{D}} \forall x \in A \cap \mathcal{G} \forall y \in A \cap \mathcal{G} (x \neq y \Rightarrow S(x,\mathcal{G}) \cap S(y,\mathcal{G}) = \emptyset)$

2) $\forall \mathcal{G} \in \mathcal{U}_{\mathcal{D}} \bigcup_{x \in A \cap \mathcal{G}} S(x,\mathcal{G}) = \mathcal{G}$

The first statement states that all the slices respective to instances of the Master class are distinct, that is, they do not have common objects. The second statement states that these slices cover the whole instance $\mathcal{G}$.

There is only one Master class in the Study Program ontology seen in Fig. 1. (given the specified OCL constraint) – the class “Study program”. Indeed, if we take an instance of the class “Study program”, its respective slice covers all the courses of that program together with its students. Since the OCL constraint prohibits for any student to take course from a different program than he is attached to, it is clear that respective slices of instances of the class “Study program” are distinct thus dividing any legal instance of the class diagram into slices.

Data ontologies (together with the legality constraints), for which there exists a Master class, are called *granular* ontologies. We depict the Master class with a bold frame in granular ontologies as can be seen in Fig. 1 (in case there is more than one Master class in an ontology we just choose one). Further in this paper we inspect only granular ontologies.

The main objective of dividing the instance graph into slices is that thus we could form natural queries over the instance easily. Since the data are naturally divided into slices, we can formulate questions either within some concrete slice, or over a set of slices. For example, we can take one concrete slice specified by the name of the study program and count the sum of credit points of all courses of that program (e.g., the question “How much credit points are to be collected in the Computer Science Master study program?”). Another example – we can select a set of slices specified by the age of the students and see, in which study programs these students are assigned to (e.g., the query “Please, give me the list of all the programs, in which there are at least one student older than 30 years!”). The means for formulating such kind of queries and getting the results are described in the next Section.

It must be mentioned that the class of granular ontologies is relatively rich. We can see, how the division into slices becomes apparent in case of static class diagram seen in Fig. 1. However, the situation with division into slices is especially characteristic, if the data ontology describes some processes, and instances of the ontology are run-time instances (transactions) of those processes. Good example of such process is the shopping basket process widely used in the field of data mining. However, in this paper we will use another example as a base – clinical processes in a hospital. For describing such processes a special language MEDMOD is introduced in authors’ paper [2].

Formally, the language MEDMOD is defined as a profile on UML Class diagrams according to Fig. 2 (OCL constraints are omitted here). An example clinical process describing the Emergency department management is seen in Fig. 3.

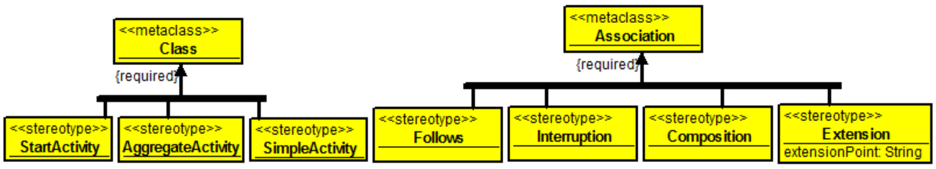


Fig. 2. The UML profile defining the MEDMOD language.

On the basis of Fig. 3 we will now shortly explain the used notations. As is described in the profile, Activities are divided into three categories. *Start Activity*, e.g., “Patient enters the hospital” (called also the Master Activity) is depicted with bolder frame in Fig. 3. *Aggregate Activities* (consisting of subactivities), e.g., “Clinical process in ward” are depicted with dashed frames (see Fig. 3). Simple activities are all the other activities, e.g., the activity “Doctor sets diagnosis” in Fig. 3. As is seen in Fig. 3, some activities are depicted with a multiple frame. That means that several instances (more than one) of these activities can appear in one slice.

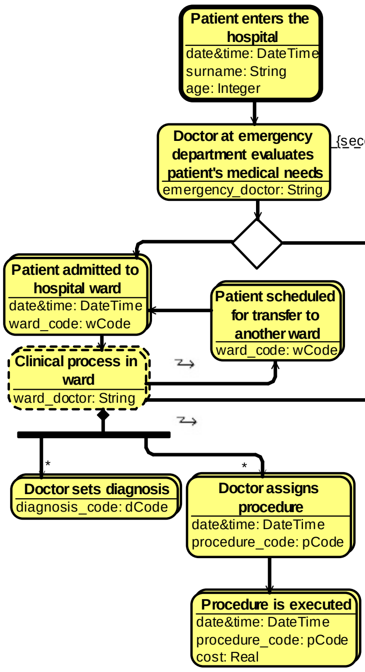


Fig. 3. An example of a MEDMOD process – the Emergency department management process.

Associations are divided into four categories:

1) <u>Follows.</u> This type of oriented relation can be established between two Activities A and B meaning that Activity B can only start after Activity A has ended. It is allowed for several Activities to follow the same Activity – the XOR semantics is implied in this case meaning that only one of those outgoing flows can be executed. We denote this situation by introducing a new diamond-shaped graphical element seen in Fig. 3.
2) <u>Composition.</u> A composition between two Activities can be established, if one Activity (called the Aggregate) semantically consists of one or more other Activities (called the Components).
3) <u>Interruption.</u> If there is an outgoing Interruption flow from the Aggregate Activity A to some Activity B, it means that the Activity A is suspended, when the flow is executed (i.e., when the Activity B needs to be started) meaning that it can no more create new Component instances (already created Component instances continues to execute normally).

4) Extension. Extension is an oriented relation between two Activities A and B meaning that Activity B can be called at some time during the execution of Activity A. The call is triggered, when some predefined condition occurs. The condition is described as an Extension point name and attached to the Extension.

The reason behind developing a new language was that the traditional process modeling languages have found a limited use in the hospital settings (see, e.g., [3], [4]). One of the reasons behind this delay has been the lack of clear definition of the sequence of activities that are carried out in clinical processes.

Since a MEDMOD diagram is formally a Class diagram according to the profile seen in Fig. 2, we can talk about instances of this class diagram, and we can investigate the notion of granularity of MEDMOD diagrams. The Master class comes out very naturally in this case, because the process diagram always has the starting action, which can serve as the Master class. The conclusion is that the ontology given by the MEDMOD language is granular.

Since the instance graph is again divided into slices (assuming we have formulated the instance legality criteria), we can query it either by specifying one concrete slice or several similar slices (e.g., “What is total expenses for the patient Wolf?”), or over a set of slices (e.g., “What is the average age of all patients treated by the doctor Stan Lee?”).

The query language described in the next section is explained based on the MEDMOD example.

3 Query language

If an ontology is *granular* – its underlying instance graph can be divided into slices, then we can define simple and efficient means of querying the instance graph. In this Section we describe an ontology based graphical in-place query language that is easy to use even by non-IT specialists and the result of a query can be retrieved in the linear time $O(n)$ where n is the number of objects in the instance graph.

Since instance graph has been divided into slices accordingly to some granular ontology, questions can be asked accordingly to that ontology. Building a query has two main activities – **filtering** and **retrieving answers**. Filtering, actually, is setting simple constraints on objects. Constraints can be set on any attribute of any class in the ontology. Once a constraint has been set, the instance graph is reduced to those *slices***,** which contains at least one object that meets the constraint. Let’s call it *the filtered instance graph*. We allow to retrieve answers for two types of questions, the first has an answer as a single number, e.g., *“How much did the Dr. Jekyll’s patients cost?”* the second has an answer as a list of objects, e.g., *“Which patients with Pneumonia had no X-ray?”*.

Very important aspect of the query language is that its concrete syntax is based on the data ontologies language used to specify the ontology. An example of a query based on the MEDMOD language is given in the Fig. 4. The real-world examples, of course, are not as tiny as the given example - just 3 patients. An average hospital in Latvia (500 beds hospital) treats about 30 000 patients per year [5]. In order to better understand the query language we give an insight in the process of building queries.

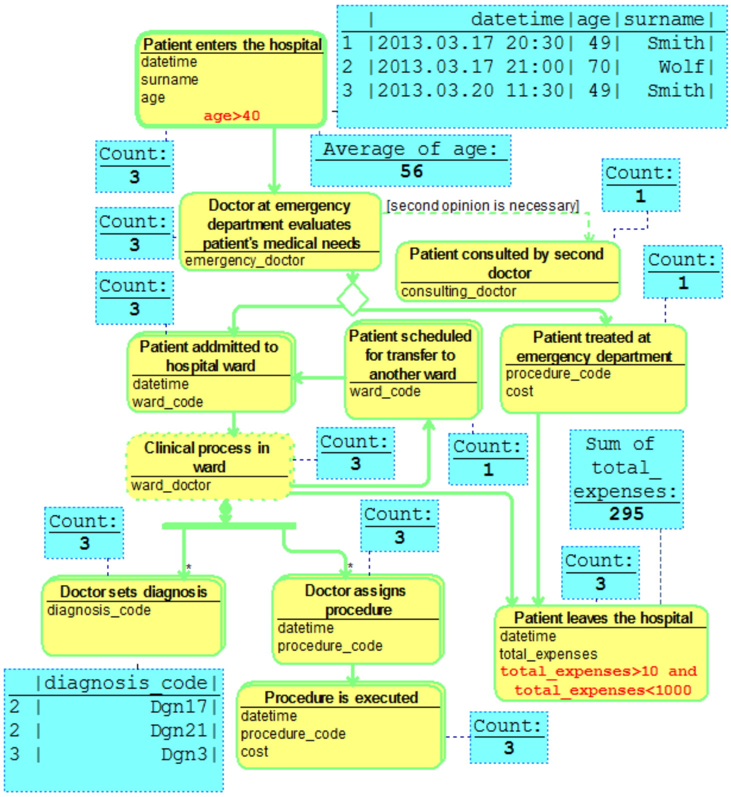


Fig. 4. An example of query based on MEDMOD – the emergency department management.

Let's assume that we have obtained the instance graph conforming to the given ontology (MEDMOD diagram). We leave behind the problem of getting data from hospital's information system. The person in interest (e.g., physician or manager) starts with a query diagram that is based on the given MEDMOD diagram – the query diagram has the same layout and elements as the MEDMOD diagram. It describes the familiar for the physician process of the emergency department management in the hospital. By default the query diagram contains boxes indicating the number of objects of each class in the instance graph. (See Fig. 4, boxes labeled *Count*). These are answers to simple questions like *"How many patients have been treated at emergency department?"* or *"How many procedures have been executed?"* It should be noted that every answer (result) is depicted as a box in the query diagram. Thus ontology, constraints, results - everything can be seen graphically *in-place* - in the same diagram. The same principle is used by spreadsheet applications – the user can make changes in any cell of the spreadsheet and observe the immediate effects on calculated values. In contrast most of query languages, e.g., SPARQL or SQL, have separate representations for data model, query and data. Now one can start filtering data by pointing to a class in the diagram and selecting an attribute. Simple constraints on attribute's values can be set – comparisons like *equals*, *greater than*, *less than*, etc., can be made to the constants of appropriate type. Following the simplicity of spreadsheet applications, no more than two constraints (comparison operations) are allowed on each attribute. Both constraints may be mandatory (logical AND), or at least one of the constraints must be met (logical OR).

When a constraint on an attribute has been set, the instance graph is filtered and all answers (result boxes) in the diagram are reevaluated and all boxes refreshed. Thus the dynamic response to each step in construction of the complete query allows the physician to see immediate reaction to every action. It shortens the learning curve greatly and reduces the number of errors – they can be recognized much earlier. This effect is called *direct manipulation* interaction mechanism [6].

As it was mentioned earlier, all answers were depicted as boxes in the diagram. At any moment these boxes can be removed and new boxes can be added. Possible single number answers are: *Number of objects of given type in the filtered instance graph*, *Sum of values of given attribute in the filtered instance graph*, *Average of values of given attribute in the filtered instance graph*. The only allowed answer that is not a single number is *the list of objects (with attribute values) of given type in the filtered instance graph*. (See Fig. 4 for all types of answers).

Let's define the constraint, the query and the answer more formally. Assume that we have a granular ontology $\mathcal{D}$ which consists of classes which in turn contains attributes. Since the ontology $\mathcal{D}$ is granular, there exists some Master class $A \in \mathcal{D}$. Before one can query the instance graph $\mathcal{G}$, it must be divided into slices respective to objects of class A. Thus the queries must be executed over set of non-overlapping slices S.

$$S = \{s \mid s = S(x, \mathcal{G}) \ \& \ x \in A\}$$

Slices consist of objects with associated key-value lists, where keys are attribute names and values are attribute values. The ontology determines possible attributes and their range of values (type) for objects of given class.

Let a be an attribute of some class $\mathcal{B} \in \mathcal{D}$ and let $T \in \mathcal{D}$ be the type of the attribute a. Then **constraint on attribute** a is the following Boolean expression:

1) One of the simple comparisons – $a > const$, $a = const$, $a < const$, where $const \in T$;
2) Conjunction (*and*) or disjunction (*or*) of two simple comparisons, e.g., "$a<10$ *and* $a>5$".

Such constraint can be checked on an object of class $\mathcal{B}$ in a time that consists of time that is needed to locate the value of the given attribute in the object's list of attribute values and time that is needed to do actual comparison and logical operations. Thus, the total time needed to check a constraint on a single object depends only on the size of the given ontology and implementation (coding) of objects. Therefore for each ontology and its implementation there exists such constant $\mathcal{C}$, that a single constraint can be checked on a single object in time less than $\mathcal{C}$.

As it was mentioned before, the physician is allowed to set just one constraint at once. After the constraint is set it is evaluated immediately. Let's define more precisely, what does it mean to evaluate a constraint c on attribute a of class $\mathcal{B}$ on the instance graph (set of slices S) and obtain the filtered instance graph – the subset of S.

The main idea is to go through all slices and check all objects in particular slice. If there is an object of the given type and the constraint c evaluates to *true* on that object, then the slice is added to the filtered instance graph. It is easy to see, that in the worst case all objects in instance graph have to be checked to evaluate the constraint, but no more, because slices are non-overlapping. However, checking a single object does not require more time than the constant C, thus **the total time needed to evaluate a single constraint on the instance graph $\mathcal{G}$ is O(*n*), where *n* is the number of objects in $\mathcal{G}$.**

The complete **query** Q is the ordered set of constraints. The execution of the query starts with evaluation of the first constraint in the set and continues with gradual evaluation of next constraints on the result of the previous. As it was mentioned above, the typical number of patients treated in an average hospital in Latvia is 30000 per year. It would be the number of slices in the instance graph for the MEDMOD example. Our experience and initial experiments with query language show that last constraints in typical queries are evaluated on much smaller filtered instance graph comparing to the initial instance graph. It may allow us to predict that the execution of complex queries would be efficient for instance graphs even larger than abovementioned 30000 slices.

Now we can define more precisely, what **answers** (result boxes) can be retrieved. Once the filtered instance graph $\mathcal{FS}$ has been obtained, here are possible answers:

1) Number of instances of given class $\mathcal{B}$ in the filtered instance graph $\mathcal{FS}$

2) Sum of values of given attribute *attr* (in class $\mathcal{B}$) in the filtered instance graph $\mathcal{FS}$

3) Average of values of given attribute *attr* (in class $\mathcal{B}$) in filtered instance graph $\mathcal{FS}$

4) List of objects of given class $\mathcal{B}$ in the filtered instance graph $\mathcal{FS}$

Just like in case of constraints, also retrieving an answer does require a single inspection of an object in the instance graph. Thus, the total time to retrieve an answer on the instance graph $\mathcal{G}$ is *O(n)*, where *n* is the number of objects in $\mathcal{G}$. It should be noted that the query language may be extended without loss of efficiency by other means that also can be evaluated in the linear time, e.g., retrieving *average number of instances of given type per slice*, filtering *slices by number of instances of given type,* however we do not describe them all because of limitations of space.

To sum up, the main advantages of the proposed query language are:

- The view on data through "glasses" of familiar ontology (e.g., everybody in the hospital should know, how does it work!);
- The simple and easy-to-perceive means of setting filtering conditions require no more expertise than using spreadsheet applications (like *MS Excel*);
- The dynamic response to each step in construction of the complete query – the doctor sees immediate reaction to every action. It shortens the learning curve greatly and encourages even non-experienced users to try this out;
- The efficiency of query execution. It is required the linear time regarding to the size of the instance graph to filter and retrieve answers.

4 Related Work

Graphical query languages have been interesting to the researchers as long as textual query languages exist. They have been developed as an attempt to fulfill the promises of query languages to give an easy-to-use means for ad-hoc data analysis, because in practice the powerful query languages (like SQL) have not became the mainstream tools for non-IT users. Number of graphical (visual) query languages for relational databases emerged in the late 80-s of the previous century [7, 8, 9]. However at that time the implementation of graphical languages was an expensive and time-consuming, not even thinking of usability issues that came along the involving non-IT users. They tend to cover every feature of SQL and as the result of that we can name just few examples of graphical query building tools, like, query designer in Microsoft Access [10] that provides means to build SQL queries graphically.

At the same time the spreadsheet applications have been widely used by non-IT users. They allow dealing with data in tabular form (no relations). One of the reasons of the spreadsheet's success story is the usage simple concepts like cell, row, column, etc., coming from real paper-based documents. Another reason is the dynamic response on every action that takes place in the spreadsheet – user sees all changes in the document immediately, just like in the query language we propose in this paper.

Nowadays the graphical language workbenches [11, 12] allow building graphical languages quickly. Thus the merely forgotten question about building visual query languages is back on the timetable. Ontologies have become popular in recent years. Therefore, the attention has been shifted from relational databases and ER models to ontologies. Thus the query languages for ontologies have emerged and particularly the graphical query languages for ontologies [13, 14]. And once again, these languages focus on graphical representation of the query, try to cover all features of SPARQL and separate the representations of ontology, query and data.

5 Conclusions and Future Work

One of the main results of this paper is the notion of granular data ontologies. This notion is defined very formally in the paper. Based on the notion of granularity an in-place graphical query language is then introduced. It is partly tested on real end-users – doctors of a hospital. As the first experience has shown, the query language possesses two essential features:

1) it is easily perceptible, and it is therefore easy to use by domain experts that are not IT specialists;
2) it has very efficient (linear regarding to the size of an instance graph) execution time for retrieving answers to queries.

Many noticeable data ontologies turn out to be granular, which means an efficient query language can be developed for them. At the same time there are also lots of other ontologies, which are not granular, and that prohibits us to use our query language for them. One of the main directions of our future research is to specify another meaningful class of data ontologies, which are granular in a wider sense. We will therefore extend the notion of ontology granularity allowing one to use the efficient query language for this class of ontologies. The efficiency of the query language will be preserved, i.e. the time evaluation of the query execution will remain linear. Other future research directions include, but are not limited to the following:

1) To keep on improving the query language and to test it on a wider range of potential end-users;

2) To continue optimizing the language implementation in order to improve the time needed for retrieving answers to queries formed over data containing about 30000 hospital transactions (our goal is to get the answer in less than a second here);
3) To further investigate practical use-cases of our approach in other areas outside the context of a hospital.

Acknowledgments

This work has been partially supported by the European Regional Development Fund within the project Nr. 2010/0325/2DP/2.1.1.1.0/10/APIA/VIAA/109 and by the Latvian National Research Program Nr. 2 „Development of Innovative Multifunctional Materials, Signal Processing and Information Technologies for Competitive Science Intensive Products” within the project Nr. 5 „New Information Technologies Based on Ontologies and Model Transformations”.

References

1. Barzdins, J., Barzdins, G., Cerans, K., Liepins, R., Sprogis, A.: UML Style Graphical Notation and Editor for OWL 2. P. Forbrig and H. Günther (eds.), Perspectives in Business Informatics Research, LNBIP, Vol. 64, Springer, p. 102-113, 2010.
2. Barzdins, J., Barzdins, J., Rencis, E., & Sostaks, A.: Modeling and query language for hospitals. Health Information Science, LNCS, Vol. 7798, Springer, pp. 113-124, 2013.
3. Müller, R., & Rogge-Solti, A.: BPMN for Healthcare Processes. In Proceedings of the 3rd Central-European Workshop on Services and Their Composition, ZEUS 2011, Karlsruhe, Germany, February 21--22, pp. 65-72, Karlsruhe: CEUR-WS.org., 2011.
4. Agt, H., Kutsche, R.D., & Wegeler, T.: Guidance for domain specific modeling in small and medium enterprises. Proceedings of the compilation of the co-located workshops on SPLASH '11 Workshops, 63, 2011.
5. Central Statistical Bureau of Latvia, http://www.csb.gov.lv
6. Shneiderman, B.: Direct manipulation: A Step beyond Programming Languages, IEEE Computer, 16, pp. 57-69, 1983.
7. Angelaccio, M., Catarci, T. and Santucci, G.: QBD*: A graphical query language with recursion. In Proc. Third Human Computer Interaction Conf., 1989.
8. Cruz, I.F., Mendelzon, A.O. and Wood, P.T.: A graphical query language supporting recursion. In Proc. ACM SIGMOD Conf. Management of Data, 1987.
9. Czejdo, B., Embley, D., Reddy, V. and Rusinkiewicz, M.: A visual query language for an E-R data model. In Proc. Int. Workshop Visual Languages, Rome, Italy, 1989.
10. Microsoft Access – Office.com, http://office.microsoft.com/access
11. Sproģis, A., Liepiņš, R., Bārzdiņš, J., Čerāns, K., Kozlovičs, S., Lāce, L., Rencis, E., Zariņš, A.: GRAF: a Graphical Tool Building Framework. Proceedings of the Tools and Consultancy Track. European Conference on Model-Driven Architecture Foundations and Applications, Paris, France, pp. 18-21, 2010.
12. Graphical Modeling Framework (GMF) Tooling, http://eclipse.org/gmf-tooling
13. Zviedris, M., Barzdins, G.: ViziQuer: A Tool to Explore and Query SPARQL Endpoints, The Semantic Web: Research and Applications, LNCS, Vol. 6644, pp. 441 – 445, 2011.
14. Fadhil, A., Haarslev, V.: OntoVQL: A graphical query language for OWL ontologies. In: International Workshop on Description Logics. (2007)

vGMQL – Introducing a Visual Notation for the Generic Model Query Language GMQL

Matthias Steinhorst, Patrick Delfmann, and Jörg Becker

WWU Münster - ERCIS, Leonardo-Campus 3, 48149 Münster, Germany
{steinhorst, delfmann, becker}@ercis.uni-muenster.de

Abstract. This paper presents a visual query notation for the generic model query language GMQL. So far, GMQL allows for specifying pattern queries as complex set-theoretical formulas. This fact impedes the practical usability of GMQL, because specifying and understanding a query is unintuitive. The visual query notation we propose is a first step towards resolving this shortcoming. We derive objectives for this notation, implement it in a working prototype, as well as evaluate the notation using expert interviews and a literature survey.

Keywords: Conceptual Model Analysis, Enterprise Modeling, Pattern Matching, GMQL.

1 Introduction

The generic model query language GMQL has recently been proposed to query large collections of conceptual models [1]. Many companies have started to develop such collections as part of their business process management (BPM) [2] and enterprise modeling (EM) activities [3]. Examples of conceptual model collections include the SAP reference model with around 600 models [4], a model collection maintained by an Australian insurance company containing close to 7000 models [5], or the BIT process library containing about 700 models [6]. These examples demonstrate that such collections may indeed contain hundreds or even thousands of models [7].

Other than a form of documentation, conceptual models are a means of analyzing the aspect of corporate reality they capture in order to derive improvement potential. Given the size and complexity model collections may exhibit many practitioners have expressed the need for automatic or at least semi-automatic support of model analysis [8]. A task that frequently occurs in model analysis is querying a collection of models in order to detect certain patterns in them [7]. A pattern in this context refers to a model fragment that complies with a predefined pattern query.

Pattern detection serves a variety of analysis purposes ranging from model comparison [9-10] to model translation [11-12], model compliance checking [13], or model conflict detection [14]. GMQL was designed to support these model analysis tasks. GMQL is generic in the sense that it is able to query conceptual models of any type or graph-based modeling language. It is based on the idea that essentially any conceptual model is an attributed graph consisting of a set of nodes and a set of edges.

GMQL comes with a significant drawback: a pattern query is essentially a complex set-theoretical formula. Specifying as well as understanding a query is thus very cumbersome and unintuitive. The purpose of this paper is to introduce a visual query notation for GMQL to mitigate this shortcoming. We provide a visual shape for each GMQL construct and explain how these shapes can be used to visually model a pattern query. An initial survey of EM experts suggests that queries defined in the visual notation are much more intuitive to understand than the original formula-based pattern queries (see Section 5.1 for more details). The paper thus contributes to easing the usability of GMQL.

The remainder of this paper is structured as follows. First, we briefly introduce the core concepts behind GMQL. We then deduce objectives for a visual query notation from these concepts (Section 2). We introduce a visual shape for each GMQL construct in Section 3. We demonstrate the applicability of the visual version of GMQL that we call vGMQL by implementing it. We show the applicability of vGMQL by providing visual queries for model patterns presented in the BPM and EM literature (Section 4). We evaluate GMQL by first conducting a survey of EM experts to determine the understandability of the visual queries (Section 5.1). We then evaluate vGMQL against the backdrop of related work (Section 5.2). The paper closes with a summary and an outlook to future research in Section 6.

2 GMQL constructs and requirements for vGMQL

The basic idea of GMQL is that any graph-based conceptual model can be represented by two sets, namely the set O of its objects and the set R of its relationships (see [1] for an exact specification of all GMQL constructs). Objects denote the nodes of the model graph, whereas relationships represent its edges. We define the set of all model elements $E=O\cup R$. GMQL provides set-altering functions and operators that take these basic sets as input and perform certain operations on them. The GMQL functions fall into four classes. Functions belonging to the first class take one set of elements as input and return elements having particular characteristics like a specific type or label. The second class of functions determines elements having a particular number of (ingoing or outgoing) relationships (of a specific type). All functions return a set of sets with each inner set containing one element and all its relationships. Functions of the third class determine elements, their adjacent elements, as well as the relationship connecting these elements. The fourth class contains functions that determine paths or loops between two sets of elements. These paths may or must not contain particular model elements. Again, these functions return a set of sets with each inner set containing one path from one start element to one target element. As the theoretical roots of GMQL lie in set theory, it provides the basic set operators UNION, COMPLEMENT, and INTERSECT that perform denoted operations on two simple sets of elements. The JOIN operator performs a union on two sets if they have at least one element in common. INNERINTERSECT and INNERCOMPLEMENT perform respective operations on inner sets of a set of sets. As some operators and functions expect simple sets as input, the SELFUNION and SELFINTERSECT operators are necessary to turn one set of sets into a simple set while performing a union or intersection. A GMQL pattern query is constructed by nesting these functions and operators. Pattern queries exhibit a tree-like structure with the output of one GMQL construct serving as input for the next. Based on this brief introduction of GMQL, requirements can be deduced for a visual query notation. Visual representations for all four classes of functions as well as all operators need to be defined. The notation should furthermore allow for nesting all GMQL constructs.

3 Conceptual specification

Figure 1 contains the visual representations for all four function classes (subsections A to F) as well as the operators (subsection G). The representation depicted in subsection A denotes an arbitrary element. It can be configured such that is represents an element having a particular type or label. If the shape is not further configured, it represents the set of all model elements. The shape is contained in all other vGMQL shapes represented in subsection B to F of Figure 1. It can furthermore be used as a placeholder for any other vGMQL shape including the operator shape. In doing so, it is possible to nest and concatenate the various constructs to construct pattern queries.

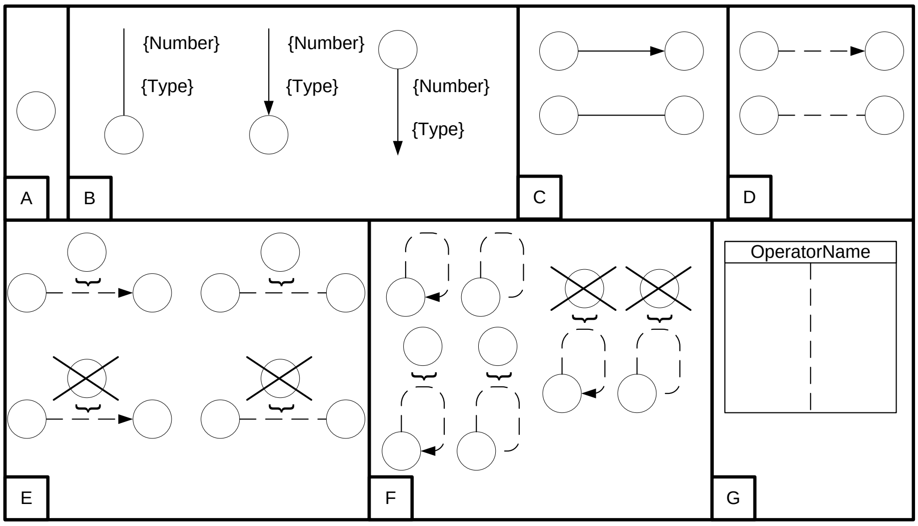


Fig. 1. vGMQL shapes

The shapes depicted in subsection B represent all functions of the second class returning single elements and all their relationships. The set R of all relationships is set to be the second input parameter for these functions. In case of directed edges, the relationships are represented by the outgoing and ingoing arrows. These functions return all relationships of a given element, even though their shapes include only one edge. The edges have two additional attributes called *Number* and *Type*. They indicate that the query is supposed to return elements having a particular number of relationships that are of a predefined type. If one of these attributes carries a NULL value, the shape represents the function taking only the other attribute as input.

The shapes depicted in subsection C of Figure 1 represent the functions of the third class returning adjacent elements and the relationships connecting them. Two different shapes for directed and undirected edges are provided. Note that these functions return all neighbors of a given element and the connecting relationships, although the shapes contain only one neighbor and relationship. Again, the shapes contain the basic element shape which allows for replacing it with any other combination of shapes (see more details below). In case the edges connecting the elements are represented as dotted lines, the corresponding shapes denote the paths functions (cf. subsection D of Figure 1). Different shapes are provided to represent functions for directed and undirected paths. The shapes depicted in subsection E represent functions for directed and undirected paths that must or must not contain specific elements. In case of the latter, the forbidden elements are crossed out. The shapes depicted in subsection F represent corresponding loop functions.

Lastly, subsection G of Figure 1 provides the visual shape for all operators. The name field can be customized to depict the corresponding operator names. The dotted line in the middle of the shape represents the two input parameters of each operator. In case the operator takes only one parameter as input, the line can be deleted.

4 Application examples and implementation

Figure 2 contains three exemplary vGMQL pattern queries for EPC diagrams (A and B) and ER models (C). The EPC queries are based on a language specification that only contains functions, events, as well as AND, OR, and XOR connectors as object types. The ERM pattern is based on a language specification containing only entity types and relationship types. All language specifications furthermore contain the respective relationship types. The pattern query in subsection A of Figure 2 represents a conflict pattern in EPCs reported by Mendling [14] who refers to this structure as an "AND join that might not get control from a splitting XOR". It represents a situation in which an AND following an EPC start event is the target element of a path that starts in an XOR split. If the start event fires and the process has run into a branch other than the one containing the AND connector, this AND will never be executed.

The pattern query depicted in subsection A contains the directed path function as its outermost shape. The first input parameter of the function represents a splitting XOR connector. It is calculated by subtracting the set of all XOR nodes having one outgoing edge from the set of all XORs. The set of XOR nodes having one outgoing edge is inner-intersected with all XORs to cut off the edge. The second parameter represents an AND join that is following an EPC start event. To that end, the shape representing adjacent elements is used. The first input parameter represents the set of all EPC start events. It is calculated by inner-intersecting the set of all events with the set of events having zero ingoing edges. To turn the resulting set of sets into a simple set the SELFUNION operator is used. The second input parameter represents a joining AND connector that is calculated analogously to a split node. This sub-query thus returns an event with no ingoing edges that is followed by a joining AND connector. This structure is inner-intersected with the set of all ANDs in order to cut off the start event as well as the relationships connecting the event and the connector. The remaining AND object is fed to the path function as its second input parameter.

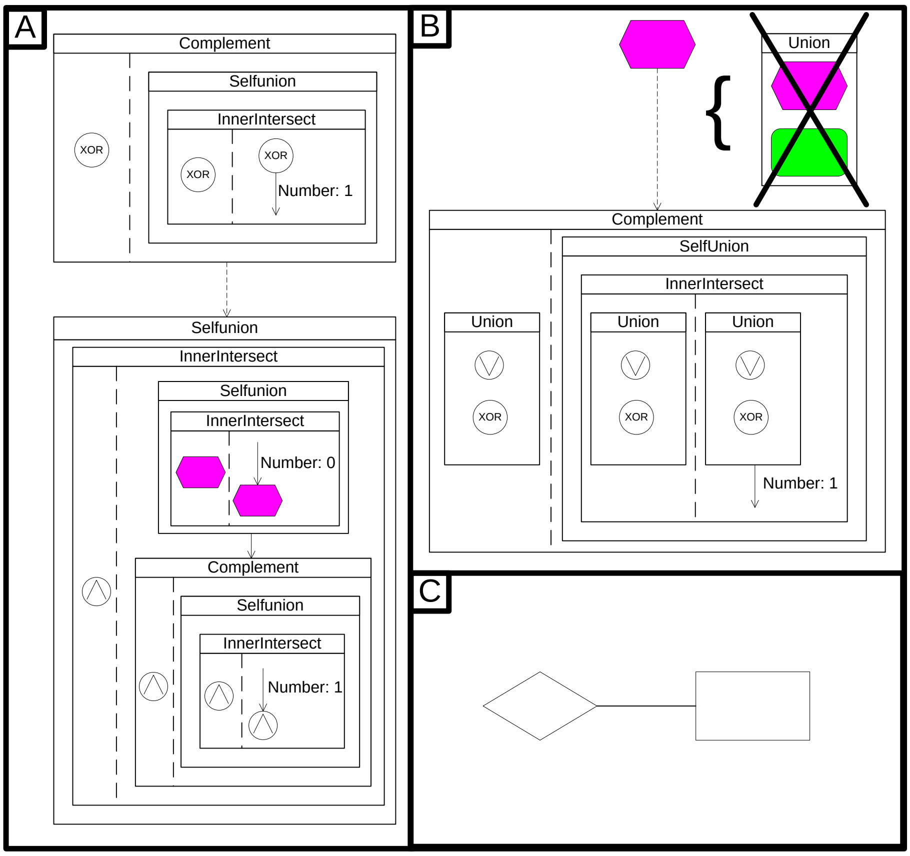


Fig. 2. vGMQL pattern queries for EPCs and ER models

The pattern query depicted in subsection B of Figure 2 represents a common syntactical error in EPC models. This error consists of a decision split after an event. This pattern can be described as an element path that starts in an event object and ends in either an OR or XOR split such that the path only contains connector objects. Functions and events are thus not allowed on this path. To define such a pattern query in vGMQL, the shape representing a path that must not contain particular elements is used. The first parameter represents the set of all event objects. The second parameter represents the set of all decision splits. Again, this sub-query is calculated analogously to the corresponding split-query described above. The only difference is that we are interested in the unified set of all XOR and OR connectors. The third parameter represents the set of all forbidden elements.

The pattern query in subsection C of Figure 2 represents an ERM relationship type that is adjacent to one or more entity types. This query thus returns binary and ternary relationship types.

Figure 3 depicts a prototypical implementation of the visual notation in a query editor. The original GMQL was implemented as a plugin for a meta-modeling tool that was available from a previous research project. The meta-modeling tool allows for specifying modeling languages by defining its object and relationship types. Similar to vGMQL the tool is based on the idea that any modeling language can be represented as the set of its element types. To develop a model, the element types of the corresponding language are instantiated to a set of elements that is used to calculate the basic sets O and R that vGMQL requires for its matching procedure. On meta-level the meta-modeling tool is thus based on the same concept that vGMQL uses to detect patterns in models. This fact allows vGMQL to be language-independent, because pattern queries can be defined for all modeling languages that can be specified using the meta-modeling tool.

The pattern matching functionality provided by vGMQL is integrated into the language editor of the tool which contains functionality for specifying languages. For each defined modeling language pattern queries can be created. All vGMQL shapes are provided on the left-hand side of the editor. The user can drag and drop these shapes on the query editing field on the right-hand side of the editor. The pattern query depicted in Figure 3 represents the EPC syntax error “decision split after event” as described above. As demonstrated in the figure, all vGMQL shapes can take other shapes as input. This allows for nesting the constructs of the query language in order to construct complex query definitions. Upon saving a pattern query, it is parsed into the original formula-based representation that is then fed to the matching mechanism. This mechanism is implemented using the visitor design pattern known from software engineering [15]. A visitor object thus traverses the query-tree in a bottom-up fashion calculating the leaf nodes of the tree first. The corresponding result serves as input for the next higher tree level.

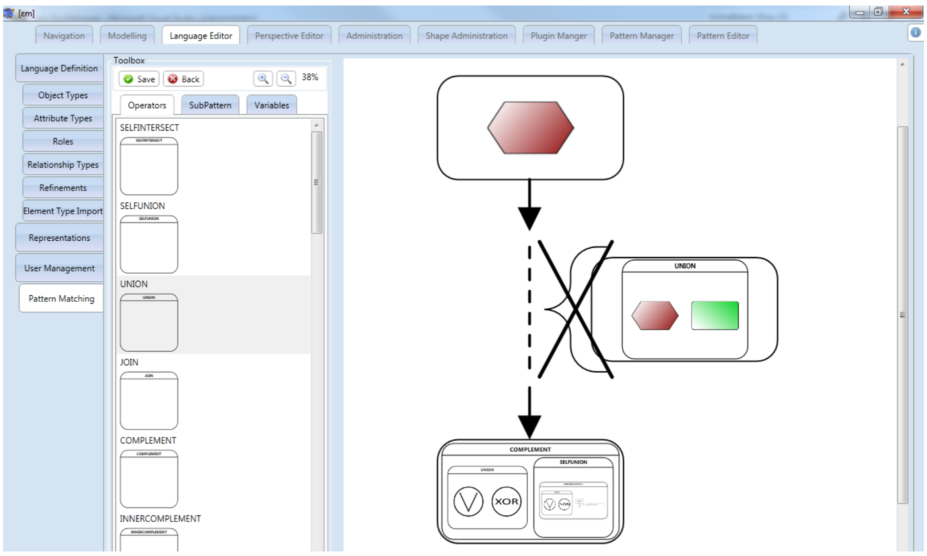


Fig. 3. vGMQL implementation

5 Evaluation

5.1 Survey

To evaluate the visual query notation, we conducted an initial survey of ten EM experts having between one and seven years of work experience. To guarantee an unbiased feedback, the experts did not have any prior knowledge of GMQL and were thus briefly introduced to its underlying concepts. The participants of the survey were then given the EPC syntax error representing a decision split after an event. We presented both the formula-based query as well as its visual counterpart (cf. Figure 2, subsection B) to the participants. They were then asked which of the queries they perceived to be more intuitive to understand. The set of possible answers also included the possibility to express that both queries are equally understandable.

Out of the ten experts involved in this initial survey, seven voted for the visual query and one participant found the formula-based query to be more intuitive to understand. Two participants furthermore perceived both queries to be equally understandable. The participant who found the formula-based query to be more intuitive argued that he is used to reading source-code and thus found the original GMQL query to be easier to understand. One participant who voted in favor of the visual query furthermore argued that the original formula-based query would potentially be as intuitive to understand as its visual counterpart provided an EM analyst possesses the necessary in-depth knowledge of the set-theoretical functions and operators. Given the results of this initial survey, we preliminarily conclude that the visual query notation we propose in this paper indeed eases the usability of GMQL, because visual queries appear to be more intuitive to understand than the original formula-based queries. Future surveys including larger sets of participants as well as pattern queries need to confirm this finding. In addition, this initial survey is limited to comparing the understandability of two given pattern queries. Additional surveys also need to focus on the perceived ease of defining queries in order to provide a complete picture of the language's usability.

5.2 Related work

vGMQL is primarily designed for a structural model analysis. vGMQL, however, is able to consider element types and labels in its matching process. Analyzing element labels is difficult, because studies indicate that conceptual models can vary significantly with respect to terms and phrase structures used to label model elements [16]. This impedes the applicability of conceptual models, because different user groups may understand particular terms differently. This in turn also impedes the applicability of vGMQL, because searching for a particular pattern containing a given label will not return all results if labels contain semantic ambiguities. Prior to searching for patterns with vGMQL it is therefore necessary to terminologically standardize labels in order to avoid semantic ambiguities like synonyms, homonyms, etc. Corresponding approaches [17-18] need to be integrated into vGMQL.

vGMQL is furthermore not designed for analyzing the execution semantics of process models. This can be achieved using finite transition systems [19] or behavioral profiles [20]. We refer to respective literature on analyzing process model execution semantics. As vGMQL includes element types in its matching process, the path functions, however, can be used to detect simple patterns representing violations to specific runtime constraints (see [1] for more detailed examples). Extending vGMQL to include process model execution semantics, however, remains subject of future research.

vGMQL furthermore assumes that there is a predefined pattern query available that can be searched for in a given collection of models. It is thus not suited for analysis scenarios in which this is not the case. Consider for example the work put forth by [21] to identify exact clones in a collection of process models. A clone represents a particular model fragment (i.e., pattern) that frequently occurs in the collection. The algorithm proposed by [21] is able to iteratively construct these patterns without having a predefined query fragment to search for. vGMQL is consequently suited for analysis scenarios in which predefined pattern queries are available. Notable examples presented in the literature include model comparison [9], model compliance checking [13], model weakness detection [22], model abstraction [23], model syntax checking [24], or model refactoring [25].

From a graph-theoretical point of view, the problem of pattern matching can be understood as the problem of subgraph isomorphism. As this problem is known to be NP-complete in the general case [26], the runtime performance of respective algorithms is a primary concern. [27] extend the well-known Ullmann algorithm for subgraph isomorphism [28] to include a filter mechanism that reduces the number of models to be searched for a given pattern. Subgraph isomorphism, however, is concerned with finding exact occurrences of a given pattern in a model. In the context of pattern matching in conceptual models, it is often necessary to find paths of previously unknown length. vGMQL provides this functionality and can thus be more broadly applied than algorithms for subgraph isomorphism.

Lastly, additional multi-purpose process query languages haven been proposed in the literature. Notable examples include BPMN-Q [29], BPQL [30], and BP-QL [31]. vGMQL differs from these approaches as it can not only search process models but also models of any other type or graph-based modeling language. With this paper, we furthermore present a visual notation that allows for visually specifying a query similarly to respective approaches presented in the literature.

6 Summary and outlook

In this paper, we presented a visual query notation for the multi-purpose and language-independent model query language GMQL. Specifying pattern queries thus no longer requires constructing complex set-theoretical formulas. Future research will focus on conducting additional surveys and experiments with EM experts to further test whether this notation is indeed easier to use than the original formula-based one. In addition, we will conduct a survey among modeling experts to determine the applicability of vGMQL in the context of specifying large and complex queries. We will also compare the proposed visualization to alternative ways of graphically modeling pattern queries. This will carve out additional user needs and determine the most intuitive way of formulating a pattern query. We will furthermore explore additional enterprise modeling related usage scenarios of the query language like ad hoc error and inconsistency detection during model development.

References

1. Bräuer, S., Delfmann, P., Dietrich, H., Steinhorst, M. 2013. "Using a Generic Model Query Approach to Allow for Process Model Compliance Checking — An Algorithmic Perspective," in Proceedings of the 11th International Conference on Wirtschaftsinformatik, Leipzig, Germany, pp. 1245–1259.
2. Rosemann, M. 2006. "Potential pitfalls of process modeling: Part a," in Business Process Management Journal (12:2), pp. 249–254.
3. Stirna, J., Persson, A. 2012. "Evolution of an Enterprise Modeling Method – Next Generation Improvements of EKD," in Proceedings of the 5th IFIP WG8.1 Working Conference on the Practice of Enterprise Modeling, Rostock, Germany, pp. 1-15.
4. Keller G., Teufel T. 1998. SAP R/3 process-oriented implementation: Iterative process prototyping, Harlow: Addison Wesley Longman.
5. La Rosa, M., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R., Mendling, J., Dumas, M., and García-Bañuelos, L. 2011. "APROMORE: An Advanced Process Model Repository," in Expert Systems with Applications (38:6), pp. 7029-7040.
6. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K. 2009. "Instantaneous soundness checking of industrial business process models," in Proceedings of the 7th International Conference on Business Process Management, Ulm, Germany, pp. 278-293.
7. Dijkman R., La Rosa, M., and Reijers, H.A. 2012. Managing Large Collections of Business Process Models: Current Techniques and Challenges," in Computers in Industry (63:2), pp. 91-97.
8. Houy, C., Fettke, P., Loos, P., van der Aalst, W.M.P., and Krogstie, J. 2011. "Business process management in the large," Business and Information Systems Engineering (3:6), pp. 385–388.
9. Yan, Z., Dijkman, R., and Grefen, P. 2010. "Fast Business Process Similarity Search," Distributed and Parallel Databases (30:2), pp. 105-144.
10. La Rosa, M., Dumas, M., Uba, R., and Dijkman, R. 2013. "Business Process Model Merg-ing: An Approach to Business Process Consolidation," Transactions on Software Engineering and Methodology (22:2), in print.
11. Ouyang, C., Dumas, M., ter Hofstede, A.H.M., and van der Aalst, W.M.P. 2007. "Pattern-based Translation of BPMN Process Models to BPEL Web Services," International Journal of Web Services Re-search (5:1), pp. 1-21.
12. García-Bañuelos, L. 2008. "Pattern Identification and Classification in the Translation from BPMN to BPEL," in Proceedings of the Confederated International Conferences on the Move to Meaningful Information Systems, R. Meersmann and Z. Tari (eds.), Monterrey, Mexico, pp. 436-444.
13. Awad, A., Decker, G., and Weske, M. 2008. "Efficient Compliance Checking Using BPMN-Q and Temporal Logic," in Proceedings of the 6th International Conference on Business Process Management, M. Dumas, M. Reichert, and M.-C. Shan (eds.), Milan, Italy, pp. 326-341.

14. Mendling, J. 2007. Detection and Prediction of Errors in EPC Business Process Models, Doctoral Thesis, Vienna University of Economics and Business Administration. Vienna, Austria.
15. Gamma, E., Helm, R., Johnson, R. E.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley Longman, Amsterdam (1995).
16. Hadar, I., and Soffer, P. 2006. "Variations in Conceptual Modeling: Classification and Ontological Analysis," Journal of the Association for Information Systems (7:8), pp. 568-592.
17. Thomas, O., and Fellmann, M. 2009. "Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes," Business and Information Systems Engineering (1:6), pp. 438-451.
18. Delfmann, P., Herwig, S., and Lis, L. 2009. "Unified Enterprise Knowledge Representation with Conceptual Models - Capturing Corporate Language in Naming Conventions," in Proceedings of the 30th International Conference on Information Systems, J.F. Nunamaker Jr., W.L. Currie (eds.), Phoenix, USA, Paper 45.
19. Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A. 1998. Deriving Petri Nets from Finite Transition Systems, IEEE Transactions on Computers (47:8), pp. 859-882.
20. Weidlich, M., Mendling, J., Weske, M. 2011. "Efficient Consistency Measurement Based on Behavioral Profiles of Process Models," IEEE Transactions on Software Engineering (37.3), pp. 410-429.
21. Dumas, M., García-Bañuelos, L., La Rosa, M., and Uba, R. 2013. "Fast detection of exact clones in business process model repositories," Information Systems (38:4), pp. 619-633.
22. Becker, J., Bergener, P., Räckers, M., Weiß, B., and Winkelmann, A. 2010. "Pattern-Based Semi-Automatic Analysis of Weaknesses in Semantic Business Process Models in the Banking Sector," in Proceedings of the 18th European Conference on Information Systems, T. Alexander, M. Turpin, and JP van Deventer (eds.), Pretoria, South Africa, Paper 156.
23. Polyvyanyy, A., Smirnov, S., and Weske, M. 2010. "Business Process Model Abstraction," in Handbook on Business Process Management 1: Introduction, Methods and Information Systems, M. Rosemann and J. vom Brocke (eds.), New York: Springer-Verlag, pp. 149-166.
24. Gruhn, V., Laue, R., Kühne, S., Kern, H. 2009. "A Business Process Modelling Tool woth Continuous Validation Support," Enterprise Modelling and Information Systems Architectures (4:2), pp. 37-51.
25. Weber, B., Reichert, M., Mendling, J., Reijers, H.A. 2011. "Refactoring large process model repositories," Computers in Industry (62:5), pp. 467-486.
26. Garey, M.R., and Johnson, D.S. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness, New York: W. H. Freeman & Co.
27. Jin, T., Wang, J., Wu, N., La Rosa, M., and ter Hofstede, A.H.M. 2010. "Efficient and Accurate Retrieval of Business Process Models through Indexing", in Proceedings of the Confederated International Conferences on the Move to Meaningful Information Systems, R. Meersman, T. Dillon, P. Herrero, P. (eds.), Crete, Greece, pp. 402-409.
28. Ullmann, J.R. 1976. "An Algorithm for Subgraph Isomorphism," Journal of the Association of Computing Machinery (23:1), pp. 31-42.
29. Awad, A. 2007. "BPMN-Q: A Language to Query Business Processes," in Proceedings of the 2007 Work-shop on Enterprise Modelling and Information Systems Architectures, M. Reichert, S. Strecker, K. Turowski (eds.), St. Goar, Germany, pp. 115-128.
30. Momotko, M., and Subieta, K. 2004. "Process Query Language: A Way to Make Workflow Processes More Flexible," in Proceedings of the 8th East European Conference on Advances in Databases and In-formation Systems, A. Benczúr, J. Demetrovics, G. Gottlob, (eds.), Budapest, Hungary, pp. 306-321.
31. Beeri, C., Eyal, A., Kamenkovich, S., and Milo, T. 2008. "Querying business processes with BP-QL", Information Systems (33:6), pp. 477-507.

Linking BPMN, ArchiMate, and BWW: Perfect Match for Complete and Lawful Business Process Models?

Ludmila Penicina

Institute of Applied Computer Systems, Riga Technical University,
1 Kalku, Riga, LV-1658, Latvia
ludmila.penicina@rtu.lv

Abstract. Enterprise architecture (EA) models are helpful for describing elements that are necessary for modelling business processes at different architectural layers of the enterprise. Business process models are used to describe detailed enterprise processes in order to analyse and improve them. Business process logic provided in the business layer of EA is very abstract comparing with business process models. Therefore EA models and business process models must be linked to address in detail both structural and behavioural aspects of the information system. However linked EA and business process models do not imply that the models provide complete and lawful descriptions of the information system. The paper uses a theoretical foundation of Bunge-Wand-Weber system's model and evaluates how industry standards BPMN and ArchiMate contribute to creation of complete and lawful business process models.

Keywords: Business process modelling, BPMN, ArchiMate, BWW.

1 Introduction

Nowadays organizations employ industry modelling standards like BPMN to understand and improve business processes. However, BPMN models are only one component of business modelling required for a holistic view of end-to-end business processes. More information is needed to build information systems supporting organizational business processes [1]. BPMN models mainly cover business process flow, but structural aspects such as actors, data objects, existing IT landscape, etc. are outside of BPMN scope. Enterprise Architecture (EA) models can reflect these aspects and are an essential component of creating accurate and complete business process models. Building complete and accurate business process models requires maintaining the relationships with EA models to add a structural context to processes (like actors, objects, etc.) and to refine business process models with an architectural layer perspective, namely, to depicting at what level each process is occurring – business, application, or infrastructure level. ArchiMate enterprise architecture modelling language has been developed in order to provide a uniform representation for diagrams that describe enterprise architectures [2]. In ArchiMate language the existence of business processes model is depicted. However, ArchiMate does not, prescribe to list the flow of activities in detail [2]. Linkage between business process models and EA models would allow looking at the business processes at different layers of the enterprise in detail.

However besides the challenge of linking two modelling languages from different domains - BPMN and ArchiMate – there exists another challenge, namely, analysing completeness and lawfulness of business process models. By "completeness of process models" the author means that models must contain all necessary elements from information system's point of view and by "lawfulness of process models" - compliance with laws related to the system. In this paper Bunge-Wand-Weber (BWW) model is used as a theoretical foundation to evaluate completeness and lawfulness of business process models. BWW model describes the necessary concepts for building an information system [3] and in this research is used to evaluate to what extent BPMN and ArchiMate support description of complete and lawful business process models. BWW model consists of constructs present in the real world that must be represented in information system.

The aim of this paper is to propose an approach towards creating complete and lawful business process models by linking BPMN models with ArchiMate models to add active and passive structure to flow aspect of BPMN and evaluate completeness and lawfulness of models using BWW model. The proposed approach requires a repository-based modelling tool that can accommodate all three modelling methods used, namely, BPMN, ArchiMate, and BWW.

The paper is structured as follows. In Section 2 related work is outlined. In Section 3 elements of BWW model are presented. In Section 4 the mapping of ArchiMate and BPMN is discussed. In Section 5 the evaluation of BPMN and ArchiMate using BWW model is discussed. In Section 6 algorithms for checking the completeness and lawfulness of business process models are discussed. Brief conclusions and future work are presented in Section 7.

2 Related Works

There exist a number of researches for linking ArchiMate and BPMN notations. The authors of [6] propose the approach of harmonizing BPMN, ArchiMate and UML notations. The authors of [7] analyse support of different kinds of active structure assignment in enterprise modelling techniques and frameworks, including ArchiMate, DODAF, and ARIS. Since these frameworks are be used in the description of an Enterprise Architecture in tandem with the detailed description of business processes, the authors also discuss the support for active structure allocation in processes modelling techniques, including XPDL, UML Activity Diagrams and BPMN in their analysis. The authors conclude that a complete integrated approach to the assignment of active structure and behaviour is yet to be incorporated into the languages and frameworks considered. However, the authors of the studies described do not propose to evaluate linked business process models and EA models for completeness and lawfulness.

The BWW model has been used in a number of studies for evaluation of modelling techniques. The authors of [5] report on the outcomes of an ontological analysis of BPMN and explore identified issues by reporting on interviews conducted with

BPMN users in Australia. As a result [5] defines few potential shortcomings in BPMN - such as existence of some ambiguous elements in its specification.

The authors of [8] examine how process modelling techniques have developed and compare modelling techniques using BWW model as a benchmark used for the analysis of grammars that purport to model, the real world, and the interactions within it. The authors of [9] propose an approach for developing a conceptual model that represents the structural, relational and behavioural elements of the computing systems based on the BWW model. The authors of [10] use of the BWW model to compare the representation capabilities of two business rule modelling languages.

This research is based on the results of related works and evaluates how the necessary elements for building an information system described by BWW model are represented by BPMN models linked with ArchiMate models.

3 BWW Model

The lack of consistent theoretical foundation for building information systems urged Wand and Weber [3] to build a set of models for the evaluation of modelling techniques. Wand and Weber have extended the systems ontology presented by Mario Bunge [4]. Wand and Weber developed a formal foundation called BWW model for modelling information systems [3] consisting of the constructs present in the real world that must be represented in information system. BWW model is a high-level ontology containing general concepts that are necessary for description of information systems [5]. Further in the text the elements of BWW model will be shown in italics. Due to the limitation of space the author has omitted the descriptions of BWW elements that can be found in [8].

The paper proposes to use BWW model as a theoretical foundation for evaluating BPMN and EA models for completeness and lawfulness. BPMN and ArchiMate models are standards with different abstraction levels, therefore gaps exist between these two standards. BPMN is used at the detailed process level, ArchiMate is used at EA level describing different layers of enterprise. BPMN and ArchiMate are complementary standards. Novelty of using BWW model as a theoretical foundation for linking BPMN and EA resides in the following:

1. Providing systems view of interlinked business processes and enterprise architecture. Interlinked ArchiMate 2.0 and BPMN 2.0 models describe elements that can be viewed and analysed as systems, e.g., application layer system, related subsystems and system environment.
2. Possibility to describe lawful states and events of systems – the evaluation of BPMN and ArchiMate using BWW shows that nor BPMN 2.0 nor ArchiMate 2.0 has the ability to describe lawful states and events of the systems at the different abstraction levels.
3. Emergent properties of systems - emergent properties describe properties possessed by a system and not by isolated elements. Emergent properties are specific properties of the system as a whole and this is added-value from BWW model.

4. Kind element of BWW model will provide the possibility to describe variations of business processes, e.g., Electronic submission process variation is Electronic submission of a journal paper or Electronic submission of a monograph.

4 Linking BPMN and ArchiMate

In an ArchiMate model, the existence of business processes is depicted [2]. It does not, however, list the flow of activities in detail [2]. The ArchiMate 2.0 specification [2] states: "During business process modelling, a business process can be expanded using a business process design language; e.g., BPMN." However the specification itself does not define the relationship at the meta-model level. The author proposes to define the linkage between BPMN and ArchiMate at the meta-models levels, by extending the behavioural elements of ArchiMate with corresponding elements from BPMN 2.0 meta-models. In this section behavioural elements of ArchiMate business, application and technology layer are mapped to corresponding BPMN elements. The principle of linking BPMN with ArchiMate resides in the following, namely, high level descriptions of enterprise behaviour are extended with corresponding BPMN models. **Table 1** describes how each element is expanded by BPMN.

Table 1. Mapping ArchiMate and BPMN.

ArchiMate Business Layer Element	**BPMN Element**
Business Process	Business Process Diagram, Pools, Lanes
Function	Task, Sub-Process
Business Interaction	Collaboration Diagram
Business Event	Event
Business Object	Data Object
Business Role	Lane
ArchiMate Application Layer Element	**BPMN Element**
Application Function	Service Task, Script Task
Data Object	Data Object
ArchiMate Technology Layer Element	**BPMN Element**
Device	Data Store
Artefact	Data Objects

5 Evaluation of BPMN and ArchiMate Using BWW Model

Business Process Model and Notation (BPMN) [11] is the de-facto standard for representing in a very expressive graphical way the processes occurring in virtually every kind of organization [12]. However BPMN has its limitations when it comes to modelling other aspects of organization such as organizational structure and roles, data, business rules, technical systems, etc. [1]. The mapping of BWW into BPMN presented in [5] is taken as a basis and extended with statements that BPMN supports the BWW notions of the *State, Property* as well as *Stable* and *Unstable States*. In BPMN the *State* of the Data Object can be captured if the Data Object is attached to a

Sequence Flow that is an input of an Activity, and it comes out of the Activity with a different *State*. *Property* of a *Thing* can be defined using BPMN Attributes of elements, *Stable* and *Unstable States* can be described using BPMN Compensation Activities and Compensation Events. The *State* of a *Thing* does not describe the overall *State* of a *System*. One solution for detecting the *State* of a *System* is to define a set of all *States* of all *Things* present in the *System*. The second solution for detecting the *State* of a *System* is to look at *System's Emergent Properties* which based on BWW are defined as properties belonging only to the *System* and not to its components [8]. One more solution is included in BPMN modelling language. BPMN allows defining multiple end states of a process. In [2] author describes the state of the process as the state of the system. Business processes do not always end normally (as intended) and very often exceptions occur. BPMN allows defining separate end events to indicate distinct end states ("normal end state" and "exceptional end state") [1]. Multiple end states of BPMN process can each correspond to *Stable* and *Unstable State* of a *System* or *Subsystem*. In BPMN the *State* of *System* can be described also with the *States* of all Data Objects that refers to a particular Pool that is considered to be a *System*. Exceptional end state should be linked with the *State* of the corresponding Data Object to reflect the *Unstable State* of a *System*. Figure 1 depicts a simple BPMN example with multiple End Events that might occur and define different end *States* of the process and system.

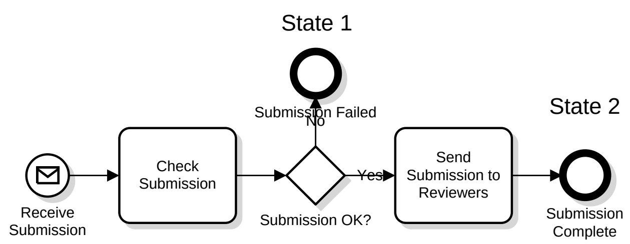


Fig. 1. Different end states of a process in BPMN.

There are 6 BWW model elements that are not supported by BPMN notation, namely, *State law, Conceivable State Space, Lawful State Space, History, Conceivable Event Space,* and *Lawful Event Space*. Since BWW model describes aspects that are important for building information systems [5], these six elements are to be taken into consideration to define a complete and consistent description of business processes.

To provide a uniform representation for diagrams that describe enterprise architectures, the ArchiMate enterprise architecture modelling language has been developed [2]. ArchiMate 2.0 language defines 3 layers of an enterprise architecture [2]:

1. Business layer offers products and services to external customers.
2. Application layer supports the business layer with application services.
3. Technology layer offers infrastructure services needed to run applications.

Table 2 shows the mapping of BWW to ArchiMate modelling language (only core elements of ArchiMate are considered. Use of extensions would provide more details concerning the *Environment* in BWW model).

Table 2. BWW elements mapped into ArchiMate elements.

BWW Elements	Corresponding ArchiMate Concepts
System	Enterprise architecture, Business layer, Application layer, Technology layer
System environment	Business layer (for Application and Technology layers), Application layer (for Technology and Business layers), Technology layer (for Business and Application layers)
System structure	Relationships
System composition	Structural concepts, Informational concepts
Level structure	Relationships between layers, Business layer, Application layer, Technology layer
Subsystem	Business layer, Application layer, Technology layer
System decomposition	Business layer, Application layer, Technology layer, Structural concepts, Informational concepts
Thing	Business actor, Business role, Business collaboration, Location, Business interface, Business object, Application component, Application collaboration, Application interface, Data object, Node, Device, System software, Infrastructure interface, Network, Communication path, Product, Contract, Artifact
Property	Meaning, Value, Representation
Class, Kind	Relationships
Event, External event, Internal event, Poorly-defined event	Business event
Transformation	Business service, Business process, Business function, Business interaction, Application function, Application interaction, Application service, Infrastructure function, Infrastructure service
Acts on, Coupling	Structural relationships
State, Conceivable state space, Lawful state space, State law, Stable state, Unstable state, History, Conceivable event space, Lawful event space, Well-defined event, Lawful transformation	***Not supported by ArchiMate***

From BWW point of view enterprise architecture is a *System* consisting of *Subsystems* – business, application and technology layers (while these sub-systems are not the only ones that can be identified in the EA). Business layer, application layer and technology layer are separate *Systems* consisting of structural and behavioural elements that are considered to be BWW *Things*. Structural and informational concepts form *System Composition* and ArchiMate Relationships between these concepts form *System Structure*. BWW element *Thing* is supported by

ArchiMate active and passive structural elements. BWW element *Property* is supported by ArchiMate element *Meaning*, since *Meaning* is related to ArchiMate Business object element and thus is as a *Property* of a *Thing*. According to BWW model *Property* maps the thing into some *Value*. According to ArchiMate Value is the relative worth, utility, or importance of a business service or product – hence it is mapped to the *Property* element of BWW model. Representation is the property of a Business Object, hence, it is mapped to the *Property* element of BWW model. BWW model elements *Class* and *Kind* are supported with ArchiMate Relationships (grouping, composite, aggregate). ArchiMate does not provide a straightforward mapping to BWW *State* element. However, if ArchiMate models are related to BPMN models, notion *State* is supported by different *States* of BPMN Data Objects and multiple End Events of the process. Since BWW model element *Transformation* is defined as a mapping from one *State* to another *State*, it is supported by all ArchiMate behavioural concepts. BWW elements *Acts* on and *Coupling* are mapped into ArchiMate structural relationships between ArchiMate concepts that are mapped into BWW *Things*. Altogether the ArchiMate modelling language does not support the description of 11 BWW model constructs comparing to 6 BWW elements missing in BPMN. Nevertheless, ArchiMate allows defining structural components of an information system at all three levels of the enterprise architecture in much greater detail than BPMN.

6 Towards Analysing Completeness and Lawfulness of Business Process Models

Business process modelling requires a meta-structure (background knowledge) that maintains the relationships between all the different models linked to business process models. When creating business process models linked with the set of enterprise models, it is necessary to achieve that all aspects of business process are stored and can be accessed and reused afterwards. Assuming that BWW model can be used as a meta-structure for analysing the completeness and lawfulness of the business process models it is necessary to identify which elements from BWW model are supported by BPMN and ArchiMate models. BWW model defines elements of the information system that are supported by BPMN and ArchiMate standards as well as a set of elements that are not supported by these standards. It indicates that complementary to BPMN and ArchiMate models it is necessary to address these missing elements in order to build the information system that conforms with a functioning system described by BWW model [3]. Mappings presented in the previous sections show that majority of BPMN and ArchiMate core elements can be mapped to BWW constructs. However, still, there exist six elements that cannot be represented using these two modelling languages, namely, *State Law (SL), Conceivable State Space (CSS), Lawful State Space (LSS), History (H), Conceivable Event Space (CES), and Lawful Event Space (LES)*. These missing BWW model elements have to be added to interlinked BPMN and ArchiMate models in order to include all BWW model's elements required for building an information system.

The proposed approach requires a repository-based modelling tool that:

1. Accommodate all three modelling methods used, namely, BPMN, ArchiMate, and BWW. Meaning that the modelling tool supports the meta-models and visual representations of BPMN, ArchiMate, and BWW.
2. Possibility to add to the BPMN, ArchiMate meta-model the missing elements from BWW model.
3. Allows defining algorithms, mechanisms, and queries to execute the completeness and lawfulness analysis on the business process models. E.g., analysing if all BWW elements are present in the business process models, analysing lawful event space – lawfulness analysis showing what are the lawful events in the business process models, analysing the lawful state space - lawfulness analysis showing what are the lawful states in the business process models, analysing whether unconceivable states and events are present in the model – lawfulness analysis showing if models are realistic.

Let's review an illustrative example. Figure 2 depicts a fragment of business process of Electronic paper submission process and ArchiMate 2.0 model that shows Business level and Application level, however does not show detailed process. The business process in the ArchiMate model called Electronic submission process is extended with BPMN business process model containing 2 lanes. Further the ArchiMate business process Receive Submission is linked with BPMN lane Editor that contains activities that Editor is responsible for. The ArchiMate business process Review Process is linked with the BPMN lane Reviewer that contains the activities the Reviewer is responsible for. The ArchiMate business role Editor is linked with the BPMN lane Editor and the business role Reviewer is linked with the lane Reviewer. Nevertheless ArchiMate model supplements BPMN model with layers and active and passive structure, still these models does not include descriptions of (the added value of BWW):

1. *State Law* - a set of all properties that are lawful to a *Submission*.
2. *Conceivable State Space* and *Lawful State Space* - to indicate what states of a data objects (e.g., *Submission*) and systems (e.g., *Application components system*) are lawful and what are conceivable. For example, *Submission* can have *Incomplete* or *Not conforming with the Template* or *Accepted* states, from which only *Accepted* is a lawful state.
3. *History* of states - for business process monitoring purposes it is necessary to maintain a log of previous states of a *Submission*, such as *Submitted*, *Reviewed*, *Accepted*, etc.
4. *Conceivable Event Space* and *Lawful Event Space* - it is necessary to indicate what events described in BPMN model are lawful, e.g., *New Submission* is a lawful event in the system, but events like *System's error* are unlawful.
5. Emergent properties of a system - one of the emergent property of Electronic submission system is faster and more efficient management of submissions.
6. Business process variations - using *Kind* element it is possible to describe different variations of the business process, e.g., Electronic submission of a monograph.

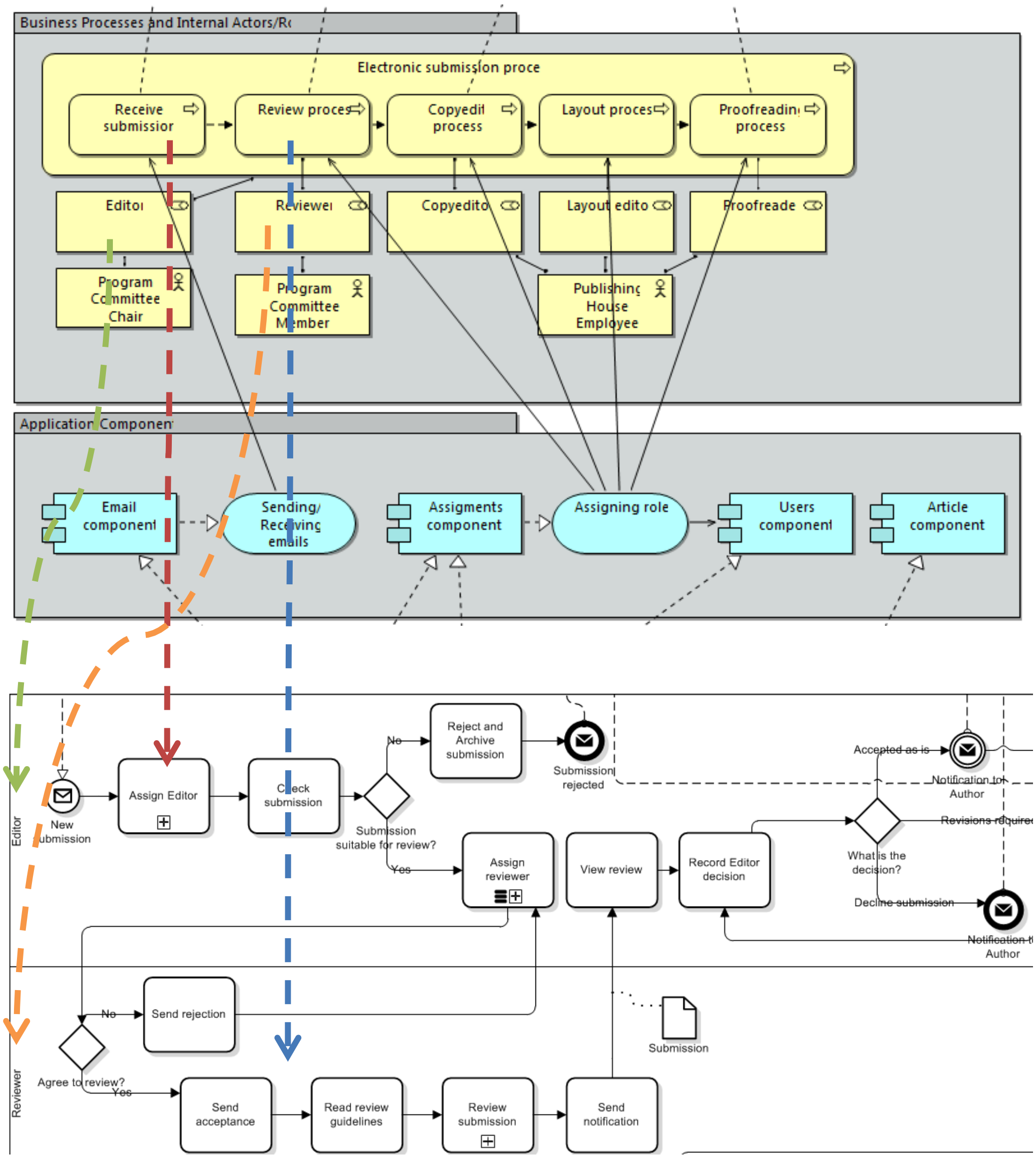


Fig. 2. Relationships between BPMN and ArchiMate.

7 Conclusions

In this paper a step towards evaluating completeness and lawfulness of business process models using BWW system's model is discussed. It was concluded that to implement the approach for completeness and lawfulness analysis the repository-based modelling tool is required. The modelling tool must allow accommodating BPMN, ArchiMate and BWW meta-models and defining algorithms, mechanisms and queries for lawfulness and completeness checking. Another essential feature is supplementing the BPMN and ArchiMate meta-models with elements from BWW model that are missing, namely, *State Law (SL), Conceivable State Space (CSS), Lawful State Space (LSS), History (H), Conceivable Event Space (CES), and Lawful Event Space (LES),* because the lack of these elements hinder lawfulness and completeness of business process models. Analysing business process models using the BWW system's model will allow analysing what necessary elements for information system developing are missing in the models. The further work will include implementing the prototype of the described repository-based modelling tool and described algorithms and queries using the ADOxx platform [13] because this platform allows creating customized modelling languages and defining algorithms, mechanisms and queries for analysing the models.

References

1. Silver, B.: BPMN Method and Style with Implementer's Guide. Cody-Cassidy Press (2011).
2. The Open Group: ArchiMate 2.0 Specification, http://goo.gl/7gC5B.
3. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information Systems Journal. 3, 217–237 (1993).
4. Bunge, M.: Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. (1979).
5. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better ? A Comparative Ontological Analysis of BPMN. Campbell, Bruce, Underwood, Jim, & Bunker, Deborah (Eds.) 16th Australasian Conference on Information Systems. (2005).
6. Berg, M. Van Den: ArchiMate , BPMN and UML : An approach to harmonizing the notations. Orbus software white paper. (2012).
7. Arpini, R.H., Almeida, J.P. a.: On the support for the assignment of active structure and behavior in enterprise modeling approaches. Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12. p. 1686. ACM Press, New York, New York, USA (2012).
8. Rosemann, M., Recker, J.: A study of the evolution of the representational capabilities of process modeling grammars. Advanced Information Systems Engineering. 447–461 (2006).
9. Goumopoulos, C., Kameas, A.: Theory and Applications of Ontology: Computer Applications. Springer Netherlands, Dordrecht (2010).
10. Muehlen, M. zur, Indulska, M., Kamp, G.: Business Process and Business Rule Modeling: A Representational Analysis. 2007 Eleventh International IEEE EDOC Conference Workshop. pp. 189–196. IEEE (2007).
11. OMG: Business Process Model and Notation 2.0, www.bpmn.org.
12. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard. Computer Standards & Interfaces. 34, 124–134 (2012).
13. BOC GROUP: ADOxx.org.

Capturing and Representing Values for Requirements of Personal Health Records

Eric-Oluf Svee[1], Maria Kvist[1,2], Sumithra Velupillai[1]

[1] Department of Computer and Systems Sciences,
Stockholm University, Kista, Sweden
[2] Department of Learning, Informatics, Management and Ethics (LIME),
Karolinska Institutet, Stockholm Sweden

{eric-sve,sumithra}@dsv.su.se
maria.kvist@karolinska.se

Abstract. Patients' access to their medical records in the form of Personal Health Records (PHRs) is a central part of the ongoing shift in health policy, where patient empowerment is in focus. A survey was conducted to gauge the stakeholder requirements of patients in regards to functionality requests in PHRs. Models from goal-oriented requirements engineering were created to express the values and preferences held by patients in regards to PHRs from this survey. The present study concludes that patient values can be extracted from survey data, allowing the incorporation of values in the common workflow of requirements engineering without extensive reworking.

Keywords: Personal health record, basic value, health care, goal-oriented requirements engineering, business/IT alignment.

1 Introduction and Purpose

To provide those goods or services which consumers desire in the method and manner which they prefer, thereby fulfilling their value proposition, it is necessary for a business to create a supporting infrastructure. Key components of such delivery mechanisms are often information systems, and as such, methods need to be developed which elicit and capture their values and preferences during the system design process, while finally presenting these to the business in such a way that they can be executed upon during the system development. The state where the goals and strategies of the business are in harmony with its IT systems is called *alignment* [1].

In this study, the business is the Swedish healthcare system, with its many-layered purposes and customers. Among these, we focus on the part of a healthcare system where the consumers are the patients, and our goal was to learn what values patients have in regards to Personal Health Records (PHRs). In the specific instance of developing a successful PHR, it is important for the business (Swedish healthcare system) to engage its consumers (patients) on a number of levels, e.g., capturing user requirements for PHR systems during development processes and user studies, as well as in the marketing process. Consumer buy-in is important for the success of any product, but in particular an individualized and deeply personal one like PHRs. Engaging a consumer's values is a crucial step towards success.

The present research utilizes results from a survey aimed at capturing patients' feature and functionality requests in a PHR system. These are then analyzed through goal-oriented requirements engineering techniques to express the values and preferences held by patients in regards to PHRs.

The paper begins with a short section to frame the general argument, and proceeds to clarify that in §2 Story. Background is provided in §3 to ground the reader in the concepts not common to enterprise modeling, specifically PHRs, business/IT alignment, and Schwartz's Value Theory. §4 presents and analyses the survey that was used for the basis of the artifact found in §5. The work concludes with a brief summary and future work in §6.

2 Story

Health records are abundant with detailed medical information including medical terminology, and are also complex in their structure. It has been shown that patients find it difficult to navigate and understand the information in their own records [2].

Electronic health record (EHR) systems are physician-oriented and do not include patient-oriented functions [3]. One problem with medical records is that they contain a lot of data which is usually kept as unstructured text in narrative form; this information overload needs to be structured and presented in a manner that patients understand. Hence, the EHR information cannot be presented directly to patients but needs to be adapted to patient requirements when exported to patient portals or PHRs. Furthermore, for the PHR to be a supporting tool for patients there is a need to identify which key functions should be implemented to support patients. Usage of PHR is highly dependent on the information offered and that functions available meet patient needs. However, few studies focused on the features that make health records comprehensible for lay audiences [see 2].

Several evaluations of the usage of patient portals have shown a decrease of patient visits and increase of online prescriptions as well as telephone and e-mail consultations [4]. These numbers of operational efficiency are presented as benefits of PHRs, as they have positive economic implications for the health care business. Also, patients report quicker access to health care by means of e-mail and telephone as positive. However, less face-to-face communication and increased online communication of sensitive nature or bad news may not be seen as a positive value for patients.

3 Background

3.1. Electronic Health Records (EHRs) and Personal Health Records (PHRs)

EHR systems were initially developed for accounting purposes and still the basic structure and vocabulary of business management is evident in the record systems. Today, EHRs are one of the most important tools for physicians and other health care professionals and are a means of communication within health care, not aimed at communication with patients. Due to the confidential nature of the content, the language in the EHRs has developed within a closed professional community and is rich in terminology, abbreviations and jargon. Many EHRs also have a structure that encourages double documentation of symptoms and events, resulting in an overload of information.

In the information age, it is quite natural that new means of communication between health care consumers and providers have evolved. Patients of today want to read the information about themselves in EHRs to follow their health care process, and want to keep their own records as PHRs. A variety of systems have been developed for this growing market, ranging from freestanding smart phone applications for e.g. vaccinations to EHR-integrated patient portals with online access. The International Standardization Organization (ISO) has defined the key features of the PHR as "it is under the control of the subject of care and that the information it contains is at least partly entered by the subject (consumer, patient)" [5]. A PHR can, as per the ISO definition, be one of the following "(a) a self-contained electronic health record (EHR), maintained and controlled by the patient/ consumer, (b) a self-contained EHR, maintained and controlled by a third party such as a web service provider, (c) a component of an integrated care EHR maintained by a health provider (e.g. general practitioner) and controlled at least partially (i.e. the PHR component as a minimum) by the patient/ consumer, or (d) a component of an integrated EHR but maintained and controlled by the patient/consumer".

Systems giving online access to (parts of) the EHR will inevitably export the problems of EHR, such as double documentation and suboptimal navigation, to the patients, if care is not taken in the design and functionalities offered. Also, to function as a means of communication, functionalities for the patients to add information and e-mail the care giver are needed. In an attempt to make EHR language more stringent and transferable between different EHR systems, international efforts have been made for a joint health care terminology, SNOMED CT [6]. However, using professional language and SNOMED terminology, which does not include layman vocabulary, will leave patients disempowered and voiceless [7].

3.2. Alignment

According to Kotler [8] consumer value plays a crucial role at the heart of all marketing activity: it is in effect a catalyst for the value exchange and refers back to the value proposition. This describes how the business will create differentiated, sustainable value [9]. This unique offering of a business demonstrates the "overall view of one of the firm's bundles of products and services that together represent a value for a specific customer segment" [10].

More recently, evolving these ideas, Kotler et al. [11] have stated that the next phase of marketing will be values driven, an evolutionary step from the original product-centric and the latter consumer-oriented types. They claim that collaborative consumers, savvy in the tools of the Internet that rapidly evolved in the past decade, and living in the age of globalization as part of a creative society, are driving companies to design their propositions around values.

Accordingly, the solutions presented in this work focus on capturing basic values and introducing them through a variety of means into the development of PHRs that support the health care system who intend to provide goods, services, and experiences to satisfy both patient and practitioner needs, based on their basic values, thus providing a core example of business-IT alignment.

3.3. Values

Value has a number of accepted meanings, with the choice of usage primarily one of context within one of two categories. Quantitative or *economic* is the type of value most commonly used in business to denote an object that can be offered by one actor to another [12] often where the worth or desirability of something is expressed as an amount of money [12]. Economic values are generally understood as an amount in goods, products, services or money, considered as a suitable equivalent for something else: the material or monetary worth of a thing [13]. These are also how companies differentiate themselves by providing a value object in a particular way [10], their *value proposition*.

In *contra poise* are values with a qualitative nature, detailing how a good, product, or service is delivered to, or perceived by, the consumer. These have been termed non-economic values [9] internal values [14], or consumer values [15] among others.

While the impact of quantitative values on IT is readily seen and acknowledged, particularly within software engineering, (e.g., value-based software engineering or VBSE [16]), qualitative values have been researched to a much lesser degree, in particular basic values. The business/IT alignment community has made several attempts such as c3 [17], e3 [18], and BMO [10], to address this deficiency, although never through the explicit use of basic values. It is through this subset of qualitative values that this research demonstrates how development of PHRs can be improved.

Basic Values. Schwartz's Value Theory (SVT) [19] adopts the definition of value as a belief that a specific mode of conduct or end-state is personally or socially preferable to its opposite. Values serve as criteria for judgment, preferences, choices, and decisions as they underlie knowledge, beliefs, and attitudes.

According to Schwartz, all the items found in earlier value theories, including religious and philosophical discussions of values, can be classified into one of the following motivationally distinct Basic Values (Table 1): Power, Universalism, Achievement, Benevolence, Hedonism, Tradition, Stimulation, Conformity, Self-determination, and Security. SVT emphasizes the profound nature of values, but at the same time offers the possibility of a consumer research approach by concretely combining these value structures with an analysis of human motivation. This integrated structure of values can be summarized with two orthogonal dimensions (Table 1).

Table 1. Schwartz's Basic Values as per their Classifying Dimensions, with examples (italicized).
Hedonism shares elements of both Openness and Self-enhancement [1] [22]

Dimension	**Basic Value**	**Dimension**	**Basic Value**
Openness to Change (independence of action, thought, and feeling, and a readiness for new experiences)	Self-determination *(Creativity, Freedom)* Stimulation *(An exciting life)* Hedonism[1] *(Pleasure)*	**Self-transcendence** (concern for the welfare/interest of others)	Universalism *(Equality, Justice)* Benevolence *(Helpfulness)* Hedonism[1] *(Pleasure)*
Self-enhancement (pursuit of self-interests)	Achievement *(Success, Ambition)* Power *(Authority, Wealth)*	**Conservation** (self-restriction, order, and resistance to change)	Conformity *(Obedience)* Tradition *(Humility, Devotion)* Security *(Social order)*

Reading from the upper left, Openness to Change (combining Self-determination and Stimulation) opposes Conservation (combining Conformity, Tradition, and Security). These dimensions reflect the conflict between an emphasis on independent thought and action and a preference for change in opposition to self-restriction, preservation of traditional practices, and protecting stability. Moving to the upper right, the dimension Self-Transcendence (combining Universalism and Benevolence) opposes Self-Enhancement (combining Power and Achievement), where in the former one finds acceptance of others as equals, coupled with a concern for their welfare, while in the latter lies the pursuit of one's own relative success and dominance over others.

The values of an individual have an effect on their behavior as consumers through their attitudes, which in turn impact on their choices within the value exchange [14, 20, 17, 21]. Additionally, it was shown that values relate to real-life choices, and may also influence behavior through different manifestations, such as habits [22]. Therefore, the use of values—in particular basic values—makes a solid foundation for which to develop complex and heavily laden systems such as PHRs.

4 Study Design

A study was conducted to generate requirements that would be expressed through goal-modeling techniques. For this, the results of an existing survey [23] were further analyzed from the perspective of value modeling for requirements elicitation.

4.1. Survey: PHR-functions preferred by patients

To elucidate patients' requests on a future PHR system, a thematic analysis of interview data from five participants was used to design an online survey.

A five-point Likert scale was used to perform a descriptive analysis of the respondents' attitude to 18 statements, categorized in five themes: 1) overview of the content, 2) help to understand the content, 3) help to understand screening results, 4) communication/interaction with healthcare and 5) additional functions. Each statement also included the option for the respondent to comment in free text.

The survey was distributed to members of five patient organizations, and it was also made available in an online article published by a Swedish newspaper. 201 respondents participated in the survey.

4.2. Value model creation

Respondent comments were processed using a textual analysis technique from requirements elicitation—SVO (Subject Verb Object)—to discover the key actors and activities, as well as the patients' values and goals. The textual analysis was performed by three researchers (one clinician and two computer scientists).

The i* framework and language was chosen to formally express the discovered requirements [24] because it assists in examining and understanding the relationships among social actors [25]. Based upon the idea that a system aims to improve the relationship that some actors have with other actors, i* was directly in line with the focus of this research: improving PHRs through an exploration of the values and relationships of the actors within the system. Additionally, i* possesses a more complete set of concepts and primitives than similar goal modeling techniques such as c3 [12], e3 [18], and BMO [26].

Both Strategic Dependency (SD) and Strategic Rationale (SR) diagrams for the patient actor were fully developed, but due to space constraints only the SR is included herein, see Figure 1.

5 Results

The typical survey respondent was a female aged 54 years 7 months who suffered from some kind of illness and had good computer skills.

The survey revealed explicit answers to patients' attitudes toward suggested possible future functions in PHRs. The Likert scale responses revealed that almost all the answers were at the level of "agree" and "strongly agree".

Foremost, functions such as overviews, fact boxes and search functions were requested both in regards to screening results and medical record content. Moreover, the respondents wanted a clear overview of their illness and medication through timelines. Explanations illustrated with pictures and videos, access to a medical dictionary and text simplification were also highly requested. Also, they wanted the possibility to add information to the PHR.

For communication with caregivers, e-mail was preferred over video calls. Chat bots were least popular, as well as possibility to view PHR content in another language.

In the analysis of the free text comments some key issues were discovered: Computer Security, Anxiety, Limited Resources, Control and Fairness. Patients expressed concern about the security of their data, not only in its transmission electronically, but also in terms of access: is a family member, acting in the role of a care provider, able to read the complete file, or can certain sections be secured? Anxiety was discussed in terms of a lack of information about medical terminology, specifically whether the records would be understandable and useful to the patients.

Anxiety and Computer Security were personal goals for the individuals and are related to Schwartz's value Security.

Limited Resources and Fairness were an expression of the amount of effort the healthcare system and care providers would need to devote to maintaining such a system; not only were patients worried about care providers expending time in writing records in laymen's terminology, but also whether they would be able to treat patients as well as answer e-mails, etc. This was an interesting outcome, possibly indicating that a high number of healthcare professionals answered the survey, as this finding was also borne out in other research on the development of PHRs. These issues relate to Schwartz's value of Universalism.

The issue of Control related to patients' requests for being able to follow their own healthcare process, e.g. by transparency in the system for seeing which tests are taken and which clinicians are involved in making decisions, by having the possibility to choose treatment type, etc. This issue is related to Schwartz's value Self-determination.

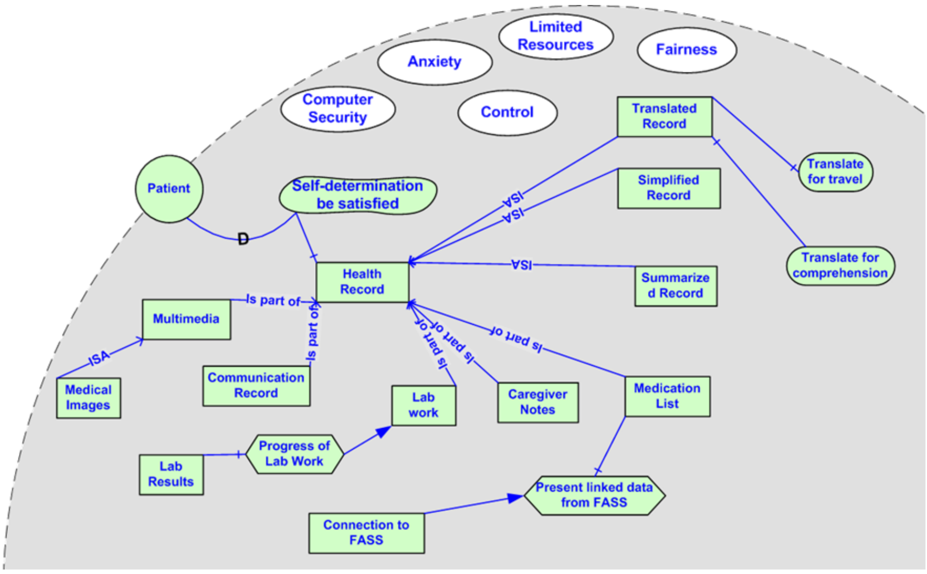


Fig. 1. SR for Patients regarding Patient Health Records

Figure 1 highlights requirements that would satisfice for the patients' softgoal "Self-determination be Satisfied". To capture, and stress the importance of, Schwartz's values, the i* constructs for a Softgoal Dependency were adapted for this SR, where the depender (Actor: Patient) depends on the dependee (Resource: HealthRecord) to perform some task that meets a softgoal (Self-determination be satisfied).

6 Lessons Learned and Future Work

Patient values can be extracted from survey data, allowing the incorporation of values in the common workflow of requirements engineering without extensive reworking. The importance of doing so should be evident from the references provided.

The textual analysis technique applied on the free text comments from the survey was useful for generating values linked to Schwartz's model, specifically Security, Universalism, and Self-determination. Our intention is to further analyze this material for identifying values in the remaining dimensions of Schwartz's model. Of course better results would have been obtained if using PVQ [21], but this study shows that it is possible to re-use survey material created for related purposes. Additionally, the i* [24] framework provided a suitable platform for modeling these values, possessing a more complete feature set than similar techniques such as c3, e3 or BMO, proved to be a sufficient choice for the goals of the study.

The current study aims to capture the patients' views, who usually want EHRs as a means of understanding and communication, while doctors usually use EHRs as a legally binding means of treatment and documentation (often even between doctors). Thus there are different requirements on medical terminology, ownership, etc. An additional study exploring the values of health care providers has been completed to explore this population. It highlights significant differences between physicians and support staff, as well as those of patients. Due to the integrated nature of PHRs, this further exploration should prove fruitful for deriving additional requirements and for supporting the contention this research makes about addressing basic values.

In summary, this *a priori* approach should not only increase user acceptance, but will consequently drive down issues such as change requests and reconfiguration. Offering patients a tailored PHR based on their values facilitates high product acceptance and can activate participation, in turn leading to empowered patients.

References

1. Vitale, M. R., Ives, B., & Beath, C. M. (1986). Linking information technology and corporate strategy: an organizational view. In Proceedings of the Seventh International Conference on Information Systems (pp. 265–276). San Diego, CA.
2. Keselman, A., Slaughter, L., Arnott-Smith, C., Kim, H., Divita, G., Browne, A., & Zeng-Treitler, Q. (2007). Towards consumer-friendly PHRs: patients' experience with reviewing their health records. In AMIA Annual Symposium Proceedings (Vol. 2007, p. 399). American Medical Informatics Association.
3. Archer, N., Fevrier-Thomas, U., Lokker, C., McKibbon, K. A., Straus, S. E. (2011). Personal health records: a scoping review. J Am Med Inform Assoc 2011;18:515-522. doi:10.1136/amiajnl-2011-000105
4. Emont S. (2011). Measuring the Impact of Patient Portals: What the literature tells us. California HealthCare Foundation.
5. ISO (International Standards Organization) (2005). TC 215 Health informatics - Electronic health record - Definition, scope, and context. ISO/TR 20514.
6. IHTSDO (International Health Terminology Standards Development Organisation). SNOMED CT, http://www.ihtsdo.org/snomed-ct/, Retrieved 30 August 2013.
7. Showell, C., Cummings, E., Turner, P. (2010). Language Games and Patient-centered eHealth. In: Seamless Care-Safe Care: The Challenges of Interoperability and Patient Safety in Health Care; Proc. 10th EFMI Special Topic Conf. 2010:vol 155: pp 55-61
8. Kotler, P., & Keller, K. L. (2006). Marketing management 12e. New Jersey.
9. Afuah, A., & Tucci, C. L. (2000). Internet business models and strategies: Text and cases. McGraw-Hill Higher Education.
10. Oxford University Press. (1971). value. Oxford English Dictionary, The Compact Edition. Oxford University Press, USA.
11. Kotler, P., Kartajaya, H., & Setiawan, I. (2010). Marketing 3.0: from products to customers to the human spirit. Wiley.
12. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., & Ilayperuma, T. (2006). On the notion of value object. In Advanced information systems engineering (pp. 321–335). Springer.
13. Ladhari, R., Pons, F., Bressolles, G., & Zins, M. (2011). Culture and personal values: How they influence perceived service quality. Journal of Business Research, 64(9), 951–957.

14. Cai, Y., & Shannon, R. (2012). Personal values and mall shopping behaviour: The mediating role of intention among Chinese consumers. International Journal of Retail & Distribution Management, 40(4), 290–318.
15. Holbrook, M.B. (1998). Consumer Value: A Framework for Analysis and Research. Routledge. London, GBR
16. Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., & Grünbacher, P. (2005). Value-based software engineering. Springer.
17. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., & Ilayperuma, T. (2007). Strategic Analysis Using Value Modeling--The c3-Value Approach. In System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on (p. 175c–175c). IEEE.
18. Gordijn, J., & Akkermans, J. M. (2003). Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements engineering, 8(2), 114–134.
19. Schwartz, S. (1992). Universals in the content and structure of values: Theory and empirical tests in 20 countries. In M. Zanna (Ed.), Advances in experimental social psychology, Vol. 25, 1-65.
20. Kahle, L. R. (1996). Social values and consumer behavior: Research from the list of values. In The psychology of values: The Ontario Symposium (Vol. 8, pp. 135–151). Lawrence Erlbaum Associates Mahwah, NJ.
21. Schwartz, S. (2005). Basic human values: Their content and structures across countries. In A. Tamayo and J.B. Porto (eds) Values and behavior in organizations, Vozes, 21-55
22. Schwartz, S., Melech, G., Lehmann, A., Burgess, S., Harris, M., & Owens, V. (2001). Extending the cross-cultural validity of the theory of basic human values with a different method of measurement. Journal of Cross-Cultural Psychology, 32, 519–542.
23. Ibrahim, Omran. (2013). Personal Health Records in Sweden: Functions preferred by patients (Master's thesis). Stockholm University, Stockholm, Sweden.
24. Yu, E. (1995), Modelling Strategic Relationships for Process Reengineering, Ph.D. dissertation, University of Toronto.
25. Yu, E., Giorgini, P., Maiden, N., & Mylopoulos, J. (Eds.). (2011). Social modeling for requirements engineering. MIT Press.
26. Osterwalder, A. (2004). The Business Model Ontology: a proposition in a design science approach. Institut d’Informatique et Organisation. Lausanne, Switzerland, University of Lausanne, Ecole des Hautes Etudes Commerciales HEC, 173.

Rule Governance, Social Coding and Social Modeling

Joost de Vries[1], Stijn Hoppenbrouwers[2]

[1] Everest b.v., j.de.vries@everest.nl
[2] HAN University of Applied Sciences, stijn.hoppenbrouwers@han.nl

Abstract. The current financial climate in the world forces organisations in government and finance to automate their operational decision making to the highest degree. The Dutch government is initiating an approach that facilitates quick, repeatable and correct implementation of new laws and thorough accountability of operational decisions that have been taken. The name of this approach can be translated as ‘rule governance’ or ‘agile execution of law’. This article proposes the term social modeling and argues that a solution to facilitate rule governance modeling would benefit from being based on social modeling.

Keywords: Rule Governance, Enterprise Modeling; Accountability.

1 Background

1.1 Developments in Dutch government

In the Netherlands, where the authors are situated, the government has stated the vision that by 2017 citizens can completely *digitally interact* with the government [1]. A typical example of this interaction would be the process where a citizen requests a permit using the self-service internet channel. Two major goals of this vision are higher quality of governmental service and higher efficiency.

Examples of parts of government that this affects are the agency responsible for taxes, the agency responsible for immigration and the agency responsible for employment matters. These agencies have in common that their business processes need to implement the often complex, detailed and changing obligations as stipulated by the law.

The Dutch tax agency has started an initiative that can be translated as ´agile execution of law´[1] to be able to implement their execution of tax law in a more timely, efficient and accountable manner.

[1] ´Wendbare Wetsuitoefening´ in Dutch

1.2 Developments in the financial sector

The financial sector is another part of society where core business processes largely concern the processing of information. The financial crisis that started in 2008 has led to an increase in governmental regulation of the financial sector. At the same time there banks and insurance firms are increasingly feeling the pressure to interact with their customers digitally and reduce the role of local offices.

This article focuses on the government but the reasoning can be similarly applied to the financial sector.

1.3 Developments in distributed collaboration

In 2011 Marc Andreessen argued in the Wall Street Journal that *information systems are replacing physical business processes* to an uncommon degree [2]. His venture capital investment firm put their money on this global trend by investing 100 million dollar in github.com; a solution for software development collaboration that '[orients] around people instead of around [source code] repositories' [3].

GitHub is arguably the prime example of what is called *social coding*: a major change in software development that has quickly become the standard way of realising *distributed collaboration*.

2 Story

2.1 Rule governance modelling at the Dutch tax agency

Many of the business processes that a government agency like the tax agency performs are essentially *decision making* processes. For instance in handling a request for subsidy the majority of the work is not transferring the money but deciding whether the giving subsidy is warranted given the stipulations of the law, in the case of a specific request.

To be able to react adequately to changes in the law, governmental agencies have started using business rules as a single point of definition for the key decision logic.

The Dutch tax agency came to the conclusion that using business rules is not enough for them [4]. What they need is *accountability*: They want to be able to support at all times the outcome of their operational decisions by a reasoned description how the decision follows from the relevant legal sources and from their agencies policy, so that they can account for how their decisions comply with all relevant laws.

What is also needed is *impact analysis:* In the event of an upcoming change in law they want to be able to pinpoint where exactly the work procedures and IT systems need to be changed. That way they should be able to effectuate new law in a short time frame and with minimal cost.

The process of implementing a new law involves a chain of analyses:

1. Careful *modelling of the law*. This is often called annotation after the physical process of highlighting the concepts involved in the legal text which forms the starting point of the analysis.
2. Modelling the relevant agency *policy*. The tax agency will decide how they can reach their goals in the best way given the law.
3. Modelling of the right *portfolio of products and services*[2] to fulfil these demands. Maybe a new service (event type) is needed or the scope of an existing one needs to be changed. This step and the next basically amount to enterprise architecture modelling.
4. Modelling how to best *implement* this change in the relevant *business processes* in terms of work procedures and relevant IT systems: data repositories[3], business rules, process activities.
5. Modelling of the internals of mainly IT components. This is amounts to modeling of system design. This step results in a model of the functioning business processes. After which operations start.

The resulting chain of models can be seen as a traceability graph from law to operational decisions. The traceability graph consists of an acyclic transitive relation that we name ´supports´. The model of the law from step 1 is the starting point of this graph and has a ´represents´ or ´models´ relation with text parts of specific legal sources.

At all steps of the modelling chain design *discussions about* the modelling decisions taken should be included in the traceability graph. These motivations of design decisions are not required for impact analysis but are essential for the compliance chain.

The compliance perspective has specific temporal requirements as well: obviously for any part of the model it should be specified what its validity range is. But also there's a requirement that it should be possible to reconstruct what the model at any point in time was. This last requirement is colloquially known as 'time travel'. Fowler [5] calls this a model with *multiple temporal dimensions*. The latter he calls 'time of record'. Snodgrass [6] calls this 'transaction time'.

To summarise; there are three modelling requirements that we want to address in this paper. First there should be *traceability graph* from legal source texts to operational execution of law. This traceability graph should support *impact analysis* of legal changes and *compliance analysis* of operational decisions. Second *discussions about modelling decisions* should be attached to the traceability graph for reasons of compliance analysis. Third it should be possible to *reconstruct what the traceability graph was* at a given moment in time.

2.2 The collaborative aspect of rule governance modelling

The whole process involves very different competences: legal analysts, civil servants responsible for policy, business architects, business process designers, IT system

[2] Produkt-Diensten Catalogus

[3] Gegevensadministraties

landscape architects and finally system designers and developers of the data stores, business rules and process activities involved. Also the latter steps involve a greater number of people. As a result it is unfeasible to have all the people involved work as one team in one room.

However, impact analysis and accountability can only function if the results of all the analysis steps form one *integral traceability graph* that describes the end-to-end links from legal texts analysis model on the one end down to the operational decisions on the other end. As a result the challenges of the required collaborative rule governance modelling have a lot in common with so-called *distributed collaborative modelling*. That is; collaborative modelling regardless of location and organizational affiliation.

2.3 Benefits of social coding

A study by Carnegie Mellon University in 2012 [7] found that the social coding solution Github allows users to understand “the activities of a large number of others regardless of location or affiliation.” And that “this transparency [has a potential] to radically improve collaboration and learning in complex knowledge-based activities.” They found that “people make a surprisingly rich set of social inferences from the networked activity information [offered by] Github.” Such as “inferring someone else’s goals and vision when they edit code, or guessing which of several similar projects has the best chance of thriving in the long term. Users combine these inferences into effective strategies for coordinating work, advancing technical skills and managing their reputation.”

They cite research that shows that collaborators in knowledge work who work in the same room are aware of each other’s activities “through overhearing, shared visual space and shared memory of discussions around artefacts.” As a result knowledge co-workers are aware of each other’s work state and expertise which helps them coordinate their activities.

GitHub is a system that lets people that cannot be together in the same room or department have the same type of awareness and mutual knowledge. “The GitHub site is unique in that it makes user identities, intern project artefacts and actions on them publicly visible across a wide community.” “The record of all action information combined with user subscription allows activity updates to flow across the site. […] Developers can ‘follow’ other developers and ‘watch’ other repositories, subscribing them to a feed of actions and communications form those developers or projects with frequent updates for active projects.”

By interviewing developers, the researchers found that people make a rich set of social inferences from this information. From recency and volume of activity developers got a sense of how ‘*live*’ or active a project was by the amount of commit events showing up in their feed. But also, for instance, inferences were made as to who had expertise in which areas.

Another type of inference people made was that “visible information about community interest in the form of watcher and fork counts for a project seemed to be and important indicator that a project was *high quality and worthwhile*." Developers would also *learn* from following so called 'rock star' developers: developers with a large number of followers that were "deemed to have special skill and knowledge about the domain."

The awareness and visibility created a "direct feedback and interaction between project owners and their user", "the owner could infer more clearly who their user base was, how they were using the project, and when they were having problems." The researchers describe this as a micro supply chain of projects depending on projects where projects *improve the quality of their support for depending projects* through better understanding of how their used.

3 Analysis and comments

3.1 Conceptual model of the data model of Github

To analyse Github it is helpful to consider its data model. The data model of Github can easily be reconstructed by looking at the extensive API [8]. The following UML diagram gives an impression of the underlying data on a conceptual level.

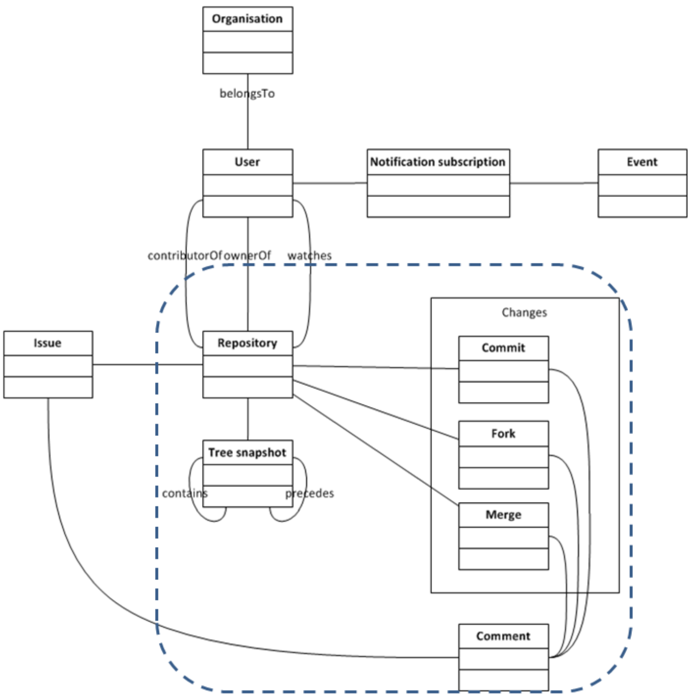


Fig. 1.Github simplified data model

The main notable aspects are that:

- Everything is an event to which a user can subscribe. An example is that a user can ‘watch’ what is happening with a repository. The resultant personal ‘feed’ of recent events largely provides the aforementioned awareness of what collaborators are doing.
- Chunks of work that need to be done can be tracked through ‘issues’, which double as feature requests and other units of work that need to be done.
- Not all comment relations are shown. Mainly, comments and discussions relate to specific (parts of) changes and issues. One could say that discussions pertain to work that needs to be done (or not) and to work that has been done and how to proceed from that.
- Everything is centred around changes and change proposals (so-called ‘pull requests’).

3.2 Github compared to other collaborative software

From the conceptual data model we observe that coding solutions like Github differ from other collaborative software in that:

1. the work is done in discrete steps and
2. specific work steps are
    a. subject to individual discussion and
    b. can individually be accepted or not or reversed at any time.

This is crucial to collaborative work processes where the work done by different collaborators needs to ‘fit’ exactly and where it is important to know exactly at what point in time which work results fulfil which tasks or targets. These requirements certainly apply for rule governance modelling.

3.3 Modelling decisions and the traceability graph

Github has an essential feature where it is very easy to submit a change to a model. Other actors have the opportunity to discuss the model in detail, suggest changes and finally accept the suggestions into the final group result. This workflow is called ‘forking’, ‘providing a pull request’ and ‘accepting and merging the pull request’ in Github parlance. From the conceptual model we learned that discussions about potential model changes are available in relationship to the changes themselves. This makes it possible to query these modelling decisions in relationship to the traceability graph and thus fulfull that requirement of agile execution of law.

3.4 Models and temporal dimensions

The Github software can be divided into the open source Git version control foundation and the commercial browser based collaboration software product built on top of it. Git is an example of Distributed Version Control (DVC) software. It is this version control core that fulfils the requirement of being able to reconstruct what the state of any model was at a given moment in time.

The part of the Github conceptual data model that is managed by Git is marked in the diagram by a blue dashed rectangle.

4 Lessons learned

We found that a social coding solution like Github offers three distinctive features that make it a suitable foundation for a solution for rule governance modelling:

- It offers support for a dependency graph of versions of models
- It offers collaborative awareness to knowledge workers who cannot be physically collocated in the same physical space at all times.
- Progress of modelling work is tracked in discrete units of work that can individually be discussed and accepted or reversed at any time
- It makes it possible to reconstruct the state of the model at previous moments in time

Environments like Github can serve as a source of inspiration for 'collaborative rule modelling environments' of the sort envisioned by the Dutch Government, and can help define requirements and patterns for the realization of such environments.

References

1. Plasterk, Ronald. Visiebrief digitale overheid 2017, http://www.rijksoverheid.nl/documenten-en-publicaties/kamerstukken/2013/05/23/visiebrief-digitale-overheid-2017.html
2. Andreessen, Marc. Why software is eating the world. *The Wall Street Journal* 20 august 2011, http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html
3. Levine, Peter. Software eats software development. http://peter.a16z.com/2012/07/09/software-eats-software-development/
4. Belastingdienst. Gestructureerd Nederlands voor Wetsanalyse. Manuscript april 2013
5. Fowler, M. (1996). *Analysis Patterns*. Addison-Wesley
6. Snodgrass, R.T. (2000). *Developing time-oriented databases in SQL*. Morgan Kaufman
7. Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM. http://www.cs.cmu.edu/~xia/resources/Documents/cscw2012_Github-paper-FinalVersion-1.pdf
8. Github API, http://developer.github.com/

The Role of Invariants in the Co-evolution of Business and Technical Service Specification of an Enterprise

Biljana Bajić-Bizumić[1], Irina Rychkova[2], and Alain Wegmann[1]

[1] LAMS, École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
biljana.bajic@epfl.ch,alain.wegmann@epfl.ch,
[2] Centre de Recherche en Informatique, Université Paris 1 Panthéon - Sorbonne,
90, rue Tolbiac, 75013 Paris, France
irina.rychkova@univ-paris1.fr

Abstract. We explore invariants as a linking mechanism between the business and technical service perspectives: From the business perspective, invariants can be used to model (business) requirements of an enterprise; from the technical perspective, invariants express the properties that must hold during the execution of a service.

We propose an approach to enterprise service design that can be described as an iterative introduction and a modification of invariants in response to the evolution of business and/or technical service specifications. We formalize the service specifications in Alloy and demonstrate how each design iteration can be simulated, visualized and validated with the Alloy analyzer tool. We illustrate our findings with the example of Order Creation service.

Keywords. Enterprise Modeling, Service Design, Service Simulation, Alloy, Declarative Specification, Model Checking, Business Rules

1 Introduction

A considerable gap between business and technical worlds (often referred to as the business/IT alignment problem [1]) represents a serious issue for implementing the "co-evolution" of business and technical specification of a service in service design and development. Therefore, we explore the invariants as a linking mechanism between business and technical service perspectives.

We propose a method for agile service specification that extends Systemic Enterprise Architecture Method (SEAM) [2]. SEAM models can be used by business and technical specialists to visually *describe* an enterprise system, its structure and services it provides. We propose a method that allows us to *simulate and validate* visual service specifications defined in SEAM. It defines five design activities (design, simulation and simulation-based testing, analysis and anomaly resolution, validation, refinement) that can be performed sequentially or iteratively, forming a *design spiral*, similarly to [4]. Within this spiral, an initial service model evolves in response to the changing business requirements and also makes these requirements evolve by revealing flaws and inconsistencies in them. This way, the partial specification is validated, verified and improved.

We illustrate our method with the example of an "Order Creation" service, specified for Générale Ressorts SA - the Swiss manufacturer of watch springs. This example is based on the consulting project we conducted with this company. The SEAM model for "Order Creation" and the transformation of this model to Alloy remains beyond the scope of this article.

This paper (a) explores the power of Alloy beyond the technical domain (b) investigates how invariants can be used as a linking mechanism between business and technical service perspectives for improved business/IT alignment.

The remainder of this paper is organized as follows: In Section 2 we explain our motivation and discuss the related works. In Section 3, we present the Alloy language and discusses the role of invariants in service design. In Section 4, we introduce our method for service design. In Section 5, we illustrate this method on the case study. In Section 6, we present our conclusions.

2 Motivation and Related Work

Since the first methods dealing with enterprise modeling (EM) that emerged in 1970s, a multitude of enterprise modeling approaches have been developed. e3Value [5] provides an ontology to conceptualize and visualize eBusiness idea and to be able to do an analysis and profitability assessment of the eBusiness model for all parties involved. The i* framework [6] focuses on modeling properties such as goals, beliefs, abilities, commitments; and on modeling strategic relationships. Enterprise Knowledge Development (EKD) [7] is a multi-model, participatory EM approach that involves a model for conceptual structures, and interlinked sub-models for goals, actors, business rules, business processes and requirements to be stated. Business Motivation Model (BMM) [8] models several concepts from goals, down to processes and technologies. The methodology that focuses more on business processes is Dynamic Essential Modeling of Organizations (DEMO) [9], which models, (re)designs and (re)engineers organizations.

SEAM [2] integrates the main principles of the well known EM methods by proposing three different types of models: SAR (business value between different stakeholders-similar to e3Value), goal-belief (goals and beliefs of the stakeholders and their relation-similar to i*), and behavior model (services and processes that implement them-similar to DEMO). In this work, we extend SEAM with the spiral design process that allows simulation and validation of SEAM models in the early stage of the design. This way, the examples of the partial specification could help the designer to realize what constraints are missing in the model. i* uses the similar approach the Formal Tropos [10] to do the model-checking of the models defined in i*. However, it focuses on the agent properties such as goals, beliefs and abilities.

Invariants have been used both in the business and technical world: to represent and check constraints [11], to model business rules [12], process invariants related to beliefs [13] etc. In requirements engineering, KAOS methodology uses invariants for object specification, domain properties specification, and indirectly for goal specification [14]. In this work, we use invariants as a a pivotal concept in improving business/IT alignment and in supporting the co-evolution of technical and business specifications.

Our method is based on Alloy, a lightweight formal specification language. The area of Alloy application is very large[1]. To the best of our knowledge, all current Alloy applications in the domain of EM target technical specialists. In this paper, we present an agile EM method where Alloy diagrams serve as a means for communicating and evaluating both business and technical design decisions. Within our approach, the role of Alloy diagrams is two-fold: They provide an instant visual feedback to a designer that suggest new constraints to be added; They represent design artifacts for validation and can drive improvements of both technical and business specifications (like UML, BPMN).

3 Foundations

3.1 Alloy

Alloy [3] is a declarative specification language for expressing complex structural constraints and behavior based on first-order logic.

The Alloy Analyzer [15] is a tool for the automated analysis of models written in the Alloy language. Given a logical formula and a data structure that defines the value domain for this formula, the Alloy Analyzer decides whether this formula is satisfiable. Mechanically, the Alloy Analyzer tries to find a model instance - a binding of the variables to values making the formula true [16].

Alloy reusable expressions (i.e., functions) and constraints (i.e., facts, predicates and assertions) [17] can be used to reason about data structures. **Fact** is a model invariant: a constraint that holds permanently. **Predicate** is a constraint that holds in a specific context or for a specific part of the model only. It can be seen as a *contextual invariant*. **Assertion** is a property that the designer believes should be implied from the model; he can check if it can be deduced from the other (permanent or contextual) constraints.

In our design process we use signatures, facts and predicates, first for partial and then for refined service specification; we use assertions in order to validate desired properties of our model.

3.2 The Role of Invariants

In computer science, an invariant is a condition that must hold during the execution of a program. Along these lines, in our design process, an invariant defines a

[1] `http://alloy.mit.edu/alloy/applications.html`

condition that must hold for all model instances that result from simulation. We define the role of invariants in our design process as follows: First, they implement the constraints required by business specification. For example, "The order can be placed for the existing parts only"; Second, they enable the designer to efficiently manage the model complexity by assuming that some of its properties always hold during an execution. For example, "To simplify the model, let's consider that the part's id provided by a customer is always correct " (i.e., exists in the database).

These roles correspond to the business and technical perspective. Therefore, in this approach we use them as a linking mechanism between these two worlds to restrict a model prohibiting some (invalid) instances identified during simulation (not necessarily covered by the explicit business specification).

4 Service Design Spiral

We introduce the five activities of our design approach, which can be performed sequentially or iteratively, forming the loops of a spiral as shown in Fig. 1a.

4.1 Model Design

We define a partial model of a service in Alloy: we specify its data structures, the initial predicate and make initial assumptions about our model defining model invariants. These invariants replace the properties required by the business specification and are used to control the model complexity.

4.2 Model Simulation

We simulate our partial model by using the Alloy Analyzer tool. Technically, a partial model written in Alloy represents a logical formula; model simulation means searching for a model instance that satisfies this formula. If it exists, it indicates that the formula is consistent (i.e., no contradictory constraints are specified). In our design process, we first check our model for consistency, and then test if it corresponds to the requirements and if there are some anomalies by studying the random set of model instances generated by Alloy Analyzer.

4.3 Model Analysis and Anomaly Resolution

There are two types of anomalies that can be observed: anomalies due to underspecification and anomalies due to overspecification.

Underspecification means that some model instances that are prohibited by the specification still appear during the simulation. In this case, we restrict the model by adding new invariants.

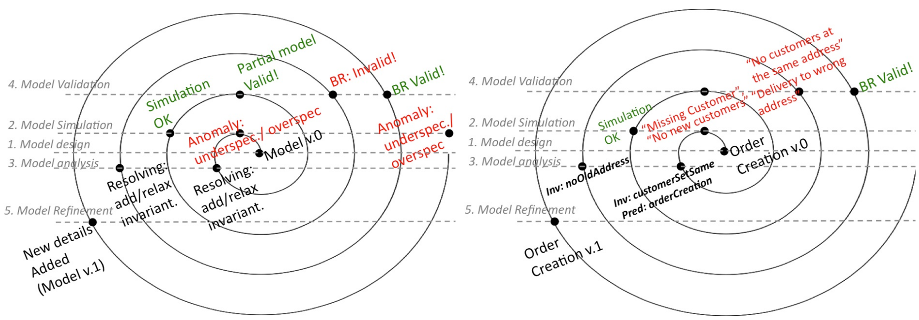
Overspecification means the opposite: some expected model instances are not observed during the simulation. The modeler then has to relax invariant, i.e. to replace an Alloy fact "X always holds" with an Alloy predicate "X holds when..." that can be activated in specific parts of the model only.

4.4 Model Validation

We make assertions about our model in order to test some desirable properties and business rules. Alloy Analyzer validates our assertion by searching for a **counterexample**: a model instance for which our assertion does not hold. If no such counterexample is found, then our assertion is valid within a given value domain. In the opposite case, the model has to be revised.

4.5 Model Refinement

In this activity, we implement new business requirements and extend our partial model. We introduce new elements in a data structure and specify new constraints. Refinement increases both the model complexity and its level of details, bringing it closer to its business specification. The complete design process is illustrated in Fig. 1a.



(a) Service design process spiral.

(b) Design process for "Order Creation" service.

Fig. 1. Spiral design process

5 "Order Creation" Design: The Case Study

In this section, we present an example of using our method for enterprise design, which involves both a technical and a business expert working together to define a complete service specification while maintaining business/IT alignment. We implement our service design process spiral step by step (Figure 1b).

5.1 Case Study: Générale Ressorts

Générale Ressorts SA is the market leader in watch barrel springs and a first-class manufacturer of tension springs, coil springs, shaped springs and industry components [18]. We illustrate our process by applying it to the design of the "Order Creation" service for Générale Ressorts SA (GR). "Order Creation" is a part of an "Order Processing"; it is followed by "Order Delivery" and "Accounting" (order-to-cash cycle).

An overview of "Order Creation" service is: "The company gets a request from a customer (*OrderRequest*-with customer *name*, *address*, *partID* and *partInfo*[2]) for manufacturing a specific watch component identified by its ID (*partID*). A company agent (*OrderEntryPerson*) identifies the customer and the part to be manufactured by entering the customer's *name* and the *partID* into the enterprise information system (*EIS*). The process terminates with a creation and confirmation of a customer order (*OrderConfirmed*) in the EIS."

We specify the following business rules for our process:

- BR1: The created order must include the complete part specification (to be used for the order fulfillment) and the complete customer details (to be used for product delivery);
- BR2: The order can be confirmed only when the customer exists in the system;
- BR3: The order can be placed for the existing parts only;
- BR4: The company has to guarantee "no faulty delivery".

5.2 Order Creation: Model Design

The data structure for the "Order Creation" service is modeled using Alloy signatures:

```
abstract sig GR {
   orderConfirmedSet: set Order,
   orderDeliveredSet: set Order,
   orderPaidSet: set Order,
   partSet: set Part,
   customerSet: set Customer
}
```

```
one sig GR_pre extends GR {
   orderRequest: one OrderRequest
}
one sig GR_post extends GR {}
```

Alloy signatures (**sig**) can be abstract or concrete, can have explicit cardinalities (e.g., only **one** *OrderRequest* object can be treated by the service at a time), and can contain one or multiple fields (as classes and attributes in object-oriented (OO) languages). We can also define additional constraints on the initial data structure with the invariants.

We express the behavior in terms of a *state transition:* we define a **pre-state** that describes the state of a system before the service has been performed and the **post-state** that describes the condition that must hold for the system upon the service termination - the service result. Note, that following the declarative modeling paradigm, we do not specify *how* the service will change the system's state.

We model the "Order Creation" service as a corresponding predicate in Alloy.

[2] We put in *italic* the names that will appear in the Alloy models.

```
1.pred orderCreation(aGR_pre:one GR_pre,aGR_post:one GR_post){
2. one aCustomer: Customer | one aPart: Part | one aOrderConfirmed: OrderConfirmed |
3.
4. aPart=findPartByPartID[aGR_pre.orderRequest.requestedPartID,aGR_pre.partSet] and
5. aCustomer= findCustomerByName[aGR_pre.orderRequest.name,aGR_pre.customerSet] and
6. aOrderConfirmed=createOrderConfirmed[aPart,aCustomer] and aGR_post.orderConfirmed=
7. aOrderConfirmed and aGR_post.orderConfirmedSet=aOrderConfirmed+aGR_pre.orderConfirmedSet}
```

This predicate shows a transition between *GR_pre* and *GR_post* states; these states are indicated as predicate parameters (line 1). In this predicate, the variables are declared (line 2), the customer and the part are found in the set (lines 4-5) and the order is created (line 6) and added to the set (line 7), as described in the case study.

5.3 Order Creation: Model Simulation and Anomaly Resolution

We attempt to simulate this model in Alloy Analyzer: to check our model for consistency and to test the random set of model instances to check for overspecification and underspecification anomalies.

Example 1. "Missing Customer" anomaly Fig. 2 illustrates an anomaly in our model behavior: In a pre-state we have *Customer0*, in a post-state we have *Customer1*. As we show exactly one execution of the service "Order Creation", we expect both the *customerSet* and the *partSet* to remain the same in pre- and post-state. However, the generated instance suggests the opposite.

NOTE: the inputs and outputs in our diagrams (e.g,, *OrderRequest* and *OrderConfirmed* in Fig. 2) are depicted with black rectangles; customer data (*Customer*, *Name*, *Address*) and part data (*Part*, *PartID*, *PartInfo*) are depicted with parallelograms and diamonds, respectively. We depict the pre-state (prior to the order creation service execution) and post-state (upon the service termination) of the GR company with "houses" and the corresponding labels: *GR_pre, GR_post*.

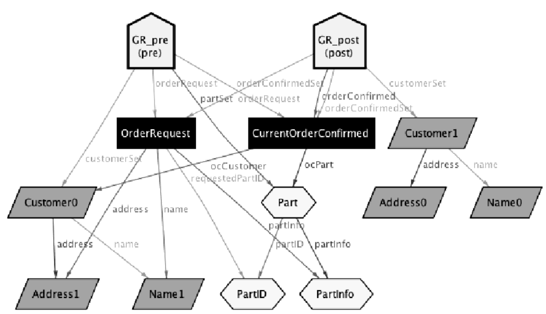


Fig. 2. Anomaly due to Underspecification: "Missing Customer"

This anomaly indicates that some constraints, which should prevent the customer set and the part set from changing during the service execution, have to be specified. Thus, it is an anomaly due to the **underspecified** model.

In fact, the declarative specification principles oblige us to explicitly state the elements that must remain "unchanged" during the state transition. Therefore, we need to add an invariant that states that the *customerSet* in post-state is the same as the *customerSet* in pre-state. The same applies to part set.

```
fact customerSetSame{ GR_post.customerSet = GR_pre.customerSet}
```

In order to validate that we have resolved the "Missing Customer" anomaly, we create an Alloy assertion that claims that *for all Order Creation executions (i.e., model instances), the customer set will remain the same in pre- and post-states of GR*.

```
customerPrePostSame: check{
 all aGR_pre:GR_pre,aGR_post:GR_post |
 orderCreation[aGR_pre, aGR_post] => aGR_post.customerSet=aGR_pre.customerSet}
```

Checking this assertion, we find no counterexamples.

```
Executing ''Check customerPrePostSame'' Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1
Symmetry=20 1014 vars. 109 primary vars. 1750 clauses. 32ms.
No counterexample found. Assertion may be valid. 12ms.
```

This confirms the assertion validity (for a given model scope). We repeat the simulation until all anomalies are resolved ("design loop" in Fig. 1).

5.4 Order Creation: Model Validation and Anomaly Resolution

We check the validity of each of the business rules from Section 5.1, using Alloy assertions. We show an example of BR4 validation ("no faulty delivery").

Example 2. "Delivery to the Wrong Address" anomaly As *OrderConfirmed* is used for delivery, to ensure "no faulty deliveries" (BR4), we check that the customer and part data in the confirmed order are exactly the same as in the requested order. The assertion "orderConfirmedCorrect" is defined to validate this BR:

```
orderConfirmedCorrect: check {
    all aGR_pre:GR_pre,aGR_post:GR_post,oReq:OrderRequest, oCurrent:CurrentOrderConfirmed |
    orderCreation[aGR_pre, aGR_post] => (oCurrent.ocCustomer.name=oReq.name and
    oCurrent.ocCustomer.address=oReq.address and
    oCurrent.ocPart.partID=oReq.requestedPartID and oCurrent.ocPart.partInfo=oReq.partInfo)}
```

When we run the assertion, we obtain the counterexamples. Fig. 3 shows an example of the incorrect delivery: the order is created on the correct customer's name, but the delivery address associated with this name does not correspond to the address provided in the *OrderRequest*. Therefore, the part can be delivered to the wrong address. The anomaly observed is due to model **underspecification**.

In order to resolve the detected anomaly, we add a new invariant "noOldAddress" that states that we cannot have a customer in the system with the name given in the requested order, but with an old/invalid address and vice versa:

```
fact noOldAddress{all c:Customer | c.address=OrderRequest.address<=>c.name=OrderRequest.name}
```

If we check now the assertion "orderConfirmedCorrect", we get the result "No counterexample found. Assertion may be valid.", meaning that this assertion holds in a given domain, and all orders will be delivered to the correct customers to the correct address.

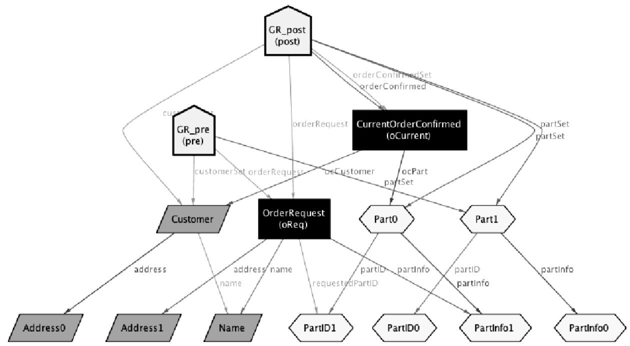


Fig. 3. Anomaly due to Underspecification: "Delivery to the Wrong Address"

We continue "debugging" the model by running the simulations, checking if we have introduced some new unwilling behavior. We repeat the process for other BRs. After validating all BRs and finding no anomalies, we conclude that the designed model meets its business requirements at a given level of details.

5.5 Order Creation: Model Refinement

At the refinement, we can add new data structures and behavior to our model. Then, we resolve all added anomalies, if any, in the "design loop". The next step is to check if the BRs still hold by repeating the "BR validation loop" until all the BRs hold. The refinement specifies a new iteration on the spiral (Fig. 1). The designer can continue refining the model until the desired level of detail is achieved. The design process we propose will ensure that, upon each iteration, the model remains consistent and has no anomalies. Refinement of "Order Creation" service will not be considered in this paper. The resulting design process of "Order Creation" (Fig. 1b) represents an instance of the spiral process illustrated in Fig. 1a.

6 Conclusion

We have presented a lightweight, interactive and visual method for service design that supports the co-evolution of technical and business service specifications of an enterprise. In particular, we have explored the power of Alloy formal method beyond the technical domain and how it can be used as a toolbox for both technical and business specialists.

The evolution of service model in Alloy can be seen as an iterative introduction and modification of logical *invariants*. Invariants represent the assumptions about business or technical properties of a modeled service and, consequently, play the role of a linking mechanism between business and technical perspectives.

This work has illustrated how Alloy can be used as a design environment for both technical and business specialists. For now, we expect that the Alloy diagrams are interpreted and analyzed by designers and business analysts. These specialists trace the observed scenarios back to the specification for its improvement. Automated interpretation and traceability between scenarios generated by Alloy and their specifications (business requirements, business rules, etc) is a subject of our future research.

References

1. Luftman, J., Papp, R., Brier, T.: Enablers and inhibitors of business-it alignment. Communications of the AIS **1**(3es) (1999) 1
2. Wegmann, A.: On the Systemic Enterprise Architecture Methodology (SEAM). In: ICEIS. (2003)
3. Jackson, D., Schechter, I., Shlyakhter, I.: ALCOA: The Alloy constraint analyzer. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland (June 2000)
4. Boehm, B.: A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes **11**(4) (August 1986)
5. Gordijn, J., Akkermans, J.: Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements engineering **8**(2) (2003) 114–134
6. Yu, E.: i* - an agent- and goal-oriented modelling framework. http://www.cs.toronto.edu/km/istar/ (page visited 2013)
7. Kirikova, M., Bubenko, J.A.: Enterprise modelling: improving the quality of requirements specifications. (1994)
8. Montilva, J., Barrios, J.: Bmm: A business modeling method for information systems development. the Clei Electronic Journal **7**(2) (2004)
9. Dietz, J.L.: Understanding and modelling business processes with demo. In: Conceptual ModelingER99. Springer (1999) 188–202
10. Fuxman, A., Pistore, M., Mylopoulos, J., Traverso, P.: Model checking early requirements specifications in tropos. In: Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on, IEEE (2001) 174–181
11. Reynolds, M.C.: Lightweight modeling of java virtual machine security constraints. In: Abstract State Machines, Alloy, B and Z. Springer (2010) 146–159
12. Kilov, H., Simmonds, I.: Business rules: from business specification to design. Springer (1998)
13. Regev, G., Bider, I., Wegmann, A.: Defining business process flexibility with the help of invariants. Software Process: Improvement and Practice **12**(1) (2007) 65–79
14. van Lamsweerde, A., Letier, E.: Handling obstacles in goal-oriented requirements engineering. Software Engineering, IEEE Transactions on **26**(10) (2000) 978–1005
15. Jackson, D.: Alloy Analyzer tool. http://alloy.mit.edu/alloy/ (2013)
16. Rychkova, I.: Formal Semantics for Refinement Verification of Enterprise Models. PhD thesis, EPFL (2008)
17. Jackson, D.: Software Abstractions- Logic, Language and Analysis. MIT Press (2011)
18. GR: Generale ressorts site. http://www.generaleressorts.com/ (2013)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly **28**(1) (2004) 75–105

A Comparative Analysis of Enterprise Modeling Approaches for Modeling Business Strategy

Constantinos Giannoulis, Iyad Zikra, Maria Bergholtz,
Jelena Zdravkovic, Janis Stirna, Paul Johannesson

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

Abstract. A gap in the alignment of business and IT lies between strategy and IS, despite the advancements of enterprise modeling. The objective of our study is to compare various enterprise modeling approaches with respect to their ability to capture and represent strategy notions. This includes identifying strategy notions from established business strategy formulations within Strategic Management, which are expressed in the Unified Business Strategy Meta-model. The interdisciplinary nature of the study constitutes a research challenge due to the significant difference on the levels of abstraction between Strategic Management and IS. To the best of our knowledge, no similar effort has been undertaken, therefore, the outcome of this study will provide the enterprise modeling community with a basis to address strategy and IS alignment linking strategic objectives and intentions to information systems.

Keywords. Business Strategy, UBSMM, Enterprise Modeling, EKD, iStar, e^3value, BMO.

1 Introduction

Organizations typically strive to attain some long-term goal (vision) with a defined purpose (mission) following a general plan. Strategic planning is the process of defining/formulating such a general plan for an organization encapsulating its intentions and actions, encompassing a certain period of time, to achieve its vision. The devised plan is commonly expressed through a business strategy, which has been broadly defined as the determination of long-term goals and courses of action using resources to achieve them, thus enabling organizations to enact the strategy [1].

Within the area of information systems (IS), Enterprise Modeling (EM) is the process for creating an integrated and negotiated model of an organization. The ensuing enterprise model helps in developing the business creating a unified and shared knowledge culture and gaining commitment from different stakeholders [2].

Business strategy coming from Strategic Management provides a business perspective of the organization, while EM offers an alternative perspective that is more Information Systems-oriented. While these two perspectives should be aligned with each other within the same organization, quite often they are not. Despite the advancements of enterprise modeling, such alignment gap between strategy and IS is still an important issue in the scope of business-IT alignment [3].

Therefore, the research question of this study is: *how do enterprise modeling approaches capture business strategy notions (explicitly or not)?* The objective of the study is to analyze a number of enterprise modeling approaches and compare their ability to capture and represent strategy notions. Such an analysis will help practitioners in selecting enterprise modeling approaches relevant to the business strategy formulations used in their organizations.

The strategy notions used for the analysis and comparisons should be derived from Strategic Management. Therefore, we are using the unified business strategy meta-model (UBSMM) as a reference model for the analysis, because it provides an integrated view of established business strategy formulations within Strategic Management (e.g. Strategy Maps and Balanced Scorecards [4], the Value Chain [5], etc.) [6], and serves as an interface of business strategy to IT, relevant to the intended analysis. This paper initiates our research and is focused on the analysis on notions of the UBSMM instantiation for Strategy Maps and Balanced Scorecards (SMBSC) [4].

Section two presents the enterprise modeling approaches analyzed. Section three presents an overview of the updated UBSMM. Section four presents the analysis. Section five discusses our findings and concludes with future directions of the work.

2 Enterprise Modeling

There exist various EM approaches that are relevant to this analysis, which can provide different views of an organization. While any EM approach can be subject to this analysis, four have been indicatively selected for our study: (i) the Enterprise Knowledge Development (EKD) approach, which provides a holistic view of an organization [2], (ii) i*, which is a requirements engineering approach focusing on social intentionality within an organization, (iii) e^3value, which is a value modeling approach focused on economic value exchanges between actors; and (iv) the Business Modeling Ontology (BMO), which provides an upper level ontology that allows describing the business model of a company accurately and in detail.

Enterprise Knowledge Development (EKD) is an EM approach that relies on six integrated “sub-models” to provide a holistic view of the organization while maintaining traceability [2]; a Goals Model (the organization’s vision and strategy), a Business Rules Model (business policies and rules), a Concepts Model (the business ontology and vocabulary), a Business Process Model (the procedural aspects of business operations), an Actors and Resources Model (organizational structure), and a Technical Component & Requirements Model (addressing IS needs).

i* is a goal modeling technique used in requirements engineering, to capture social intentionality among actors, including possible alternatives, and to operationalize stakeholders’ business goals through concrete actions and design decisions [7]. It includes two model types, the Strategic Dependency Model (SDM) where all actors are identified along with their interdependencies, and the Strategic Rationale Model (SRM), where within each actor all intentional elements are identified/ascribed.

The *e*[3]value is a business modeling ontology and approach that enables the representation and analysis of a network of enterprises exchanging resources of economic value with each other [8]. The main modeling constructs include *actors* who exchange economic resources (*value objects*), *value activities* producing these resources, and the provisioning and transfer of the resources through *services*. The *e*[3]value has been further extended with *e*[3]*forces* constructs to enable modeling of a business strategy perspective on a service offering by an enterprise and in relation to its networked value constellation, i.e. environmental forces [9].

The Business Model Ontology (BMO) provides an ontology that allows describing the business model of a firm accurately and in detail [10]. The BMO takes the perspective of a single enterprise facing a particular customer's demands and consists of nine core concepts in four categories: Product, Customer Interface, Infrastructure Management, and Financial Aspects. Key concepts are Value proposition (an overall view of a company's bundle of products and services that are of value to the customer), Target Customer (a segment of customers to which a company wants to offer value), Value Configuration (the arrangement of activities and resources necessary to create value for the customer) and Capability (the ability to execute a repeatable pattern of actions necessary in order to create value for the customer).

3 Business Strategy: UBSMM

For the scope of our work, EM approaches should be analyzed against a complete view of business strategy. This would limit the risk of our analysis to overlook strategic notions. Completeness requires all business strategy formulations are considered for the analysis, which is not feasible. Instead, based on three complementary views on business strategy where strategy shaping is driven from different perspectives [11], we aim at including all these views. The unified business strategy meta-model (UBSMM) is an integration of business strategy formulations within Strategic Management. The purpose of UBSMM is to become an interface of business strategy to IT [6] by integrating a formulation from each of these three views and allowing the progressive integration of others, both existing and emergent.

The three views [11] include: the *resource based view*, where strategy formulation is driven by the capabilities of the organization; the *industrial organization view*, where positioning the organization against competition is the main driver; and the *Schumpeterian view*, where radical innovations are in focus disrupting the environment in which the organization operates.

UBSMM includes Strategy Maps and Balanced Scorecards (SMBSC) [4], the resource-based view, and the Value Configuration (VC), based on the Value Chain [5], the industrial organization view. For our analysis, UBSMM has been extended to include Blue Ocean Strategy (BOS) [12], from the Schumpeterian view of strategy, using a conceptualization of the formulation [13]. This provides an integration of the three complementary views of business strategy, which constitutes an interface to IT.

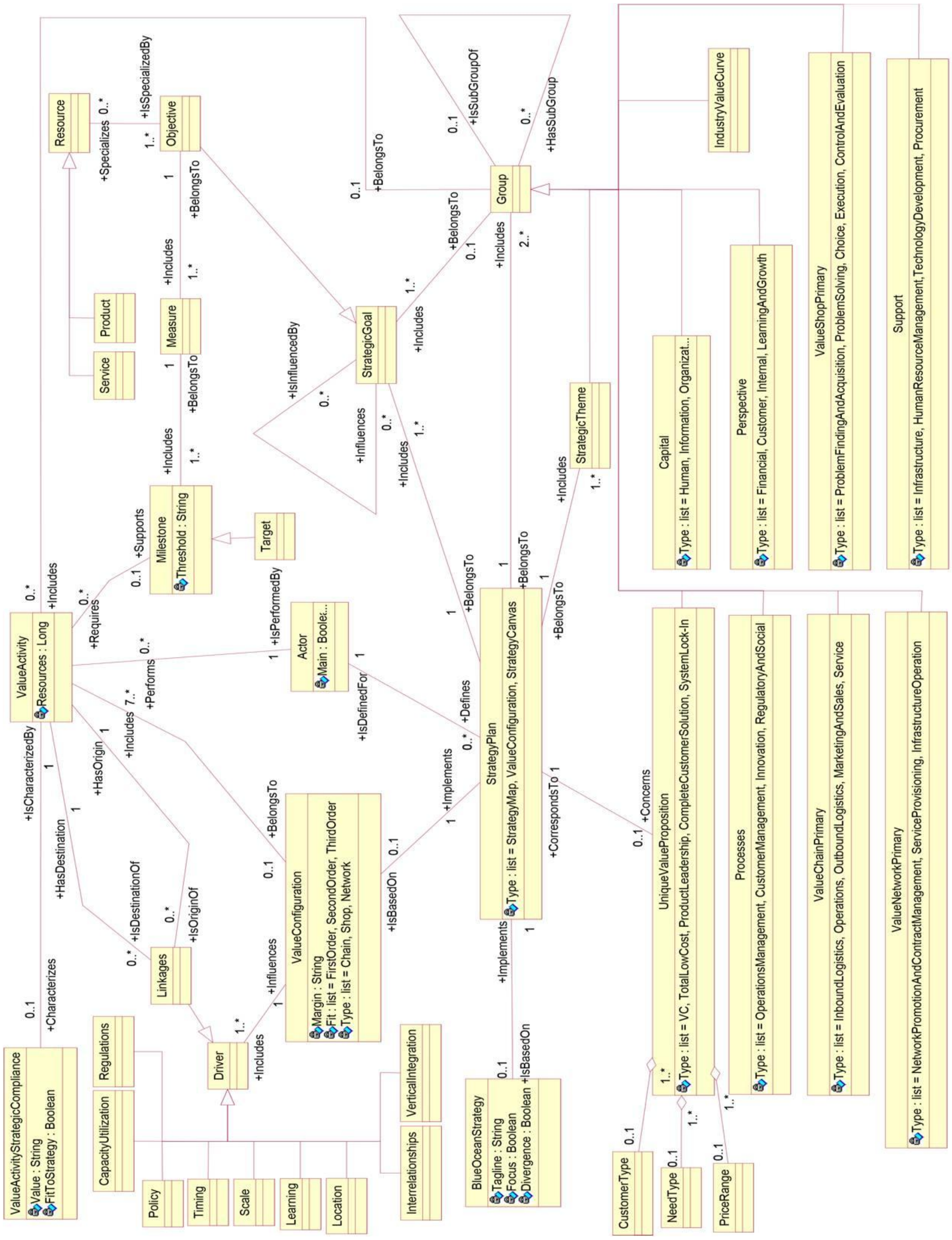


Fig. 1. The updated Unified Business Strategy Meta-model (UBSMM).

The updated UBSMM (figure 1) also includes sets of constraints that need to be applied to instantiate each of the different business strategy formulations it includes. A complete specification of UBSMM with class descriptions and constraints can be found in [14], chapter 9.

In this paper, we examine the UBMSS instantiation of Strategy Maps and Balanced Scorecards (denoted as UBSMM.SMBSC). SMBSC is a business strategy formulation serving as a mediator between the mission, core values, and the vision of an organization. A strategy map template is used to represent how an organization can create value. Starting from a mission statement and core values, a strategic vision is defined, which projects the organization's overall goal. A set of goals is then defined and grouped within the financial and customer perspectives, along with goals for all processes and all types of capital, both human and economic [4]. Goals are extended to a set of targets using measures to evaluate their achievement, and thereafter, initiatives are identified to achieve the targets. This extension of the strategy map is the balanced scorecards, which are essential for monitoring and assessing the cause-effect links between strategic goals across an organization.

From the complete UBSMM (figure 1), classes relevant to UBSMM.SMBSC are listed below (for detailed specification of class descriptions and constraints see [14], section 10.1) and are used for the analysis of EM approaches with respect to Strategy Maps and Balanced Scorecards:

- **StrategyPlan**. captures a complete strategy map.
- **StrategicTheme** captures a grouping of particular interest within a strategy map usually focusing on areas of critical importance for executives.
- **Actor** captures the organization/unit for whom the strategy map is defined**.**
- **StrategicGoal** captures the goals set across the four perspectives of a strategy map.
- **Objective** captures measurable goals that are used for building balanced scorecards
- **Group** captures all groupings and subgroupings included in a strategy map
- **Objective** captures measurable goals used for building balanced scorecards.
- **Perspective** captures the highest level of grouping in a strategy map (financial, customer, internal, learning and growth).
- **UniqueValueProposition.** captures how the actor delivers unique value (low total cost, product leadership, complete customer solution, and system lock-in). **Processes** captures groupings within the internal perspective of a strategy map.
- **Capital** captures groupings within the learning and growth perspective of a strategy map.
- **Measure** captures the way/scale to evaluate the achievement of an objective.
- **Milestone** captures any short-term or intermediate target of an objective.
- **Target** captures a final desired state, usually long-term, of an objective.
- **ValueActivity** captures an activity performed to achieve an objective.

4 Enterprise Modeling Analysis for UBSMM.SMBSC

The EM approaches included in this study are analyzed in terms of their capability to capture the notions of UBSMM.SMBSC. The findings of this analysis are presented in the following subsections and are summarized in Table 1. For each notion in UBSMM.SMBSC and each EM approach, we determine and justify if and how the EM approach supports the UBSMM.SMBSC notion. There are four possibilities:

- ***Directly Supported*** denotes that to a UBSMM class, there exists exactly one corresponding class in the EM approach with similar meaning (with respect to naming, structure, attributes).
- ***Indirectly Supported*** denotes that to a UBSMM class, there exists no corresponding class in the EM approach with similar meaning (with respect to naming, structure, attributes). However, the EM approach can still fully capture the UBSMM.SMBSC class in another way.
- ***Partially Supported*** denotes that to a UBSMM class, there exists no corresponding class in the EM approach with similar meaning (with respect to naming, structure, attributes). However, the EM approach can partially capture the UBSMM.SMBSC class in some way.
- ***Not Supported*** denotes that to a UBSMM class, there exists no corresponding class in the EM approach with similar meaning (with respect to naming, structure, attributes), and the EM approach cannot capture the UBSMM.SMBSC class in another way.

Table 1. Summary of the analysis findings of EM approaches.

UBSMM.SMBSC	**EKD**	**i***	**e^3value**	**BMO**
StrategyPlan	Partially	Partially	Partially	Partially
Actor	Directly	Directly	Directly	Partially
StrategicGoal	Directly	Directly	Not	Not
StrategicTheme	Indirectly	Partially	Not	Not
Group	Indirectly	Indirectly	Partially	Not
Perspective	Indirectly	Partially	Not	Partially
UniqueValueProposition	Partially	Partially	Indirectly	Indirectly
Processes	Indirectly	Partially	Not	Partially
Capital	Not	Partially	Not	Partially
Objective	Indirectly	Directly	Not	Not
Measure	Indirectly	Partially	Not	Not
Milestone	Not	Partially	Not	Not
Target	Not	Partially	Not	Not
ValueActivity	Indirectly	Directly	Partially	Indirectly

4.1 EKD

- **StrategyPlan: Partially Supported.** Based on the overall assessment of SMBSC with respect to EKD, there exist SMBSC notions not supported by EKD.
- **Actor: Directly Supported**. An Actor in EKD can be an Individual, an Organizational Unit, a Non-Human Resource (e.g. a machine), or a Role that can be played by instances of other types of actors [2]. Actors define Intentional Components (i.e. Business Goals and Opportunities) and are responsible for the fulfillment of Business Goals. They can also perform Processes.

- **StrategicGoal: Directly Supported**. Business Goals in EKD describe the future state-of-affairs that the enterprise aims to achieve. They provide the underlying motivation for developing process, concept, and IS architecture models. The modeling guidelines of EKD recommend goal operationalization i.e. refinement of goals into sub-goals while capturing the mutual influence among goals.
- **StrategicTheme: Indirectly Supported**. EKD is composed of different models that are connected using inter-model relationships to give an overall view of the organization and its supporting IS. The purpose is to describe the organization from different perspectives. StrategicTheme is not directly expressed in EKD. However, it can be achieved by arranging parts of the models in separate schemata, each focusing on a specific theme, thus using multiple schemata.
- **Group: Indirectly Supported**. The ability to create groups of modeling components for a certain purpose is not part of the EKD meta-model. But, grouping of modeling components can be done by introducing auxiliary components.
- **Perspective: Indirectly Supported**. EKD offers various perspectives of the organization, delimited by the type of knowledge that is being modeled (static, dynamic, intentional, etc.) rather than management boundaries. Thematic perspectives in EKD are modeled by allocating a particular model schema or view to a specific perspective, which is applicable to SMBSC perspectives.
- **UniqueValueProposition: Partially Supported**. EKD captures and represents organizational design using conceptual models. The delivered value is expressed collectively by the set of models or modeling components. But EKD does not have an explicit construct for representing value.
- **Processes: Indirectly Supported**. EKD includes the business process model (BPM) to describe processes within an organization. Though specific process types are not distinguished, processes are captured on different levels of abstraction, and can then be associated to the Goals model to allow groupings of process goals.
- **Capital: Not Supported**. Even though EKD is able to capture Opportunities that can represent growth and learning potential, the current definition of this concept falls short of expressing the multi-faceted Capital in SMBSC.
- **Objective: Indirectly Supported**. Operationalization of Business Goals in EKD enables the refinement of high-level strategic goals into more specific goals that usually have specific fulfillment criteria (expressed using the SMART goal principle), thus facilitating the measurement of the extent of goal fulfillment. Measurements are not modeled directly, but Goals can be linked to Concepts (in the Concepts model) to specify measurements and monitor goal achievement.
- **Measure: Indirectly Supported**. A dedicated modeling component for measures is not part of EKD. However, it is possible to represent measurable objectives with links to business concepts that represent measures.
- **Milestone, Target: Not Supported**. EKD has no means to capture neither intermediate (Milestone) nor final (Target) checkpoints to fulfill objectives.
- **ValueActivity: Indirectly Supported**. A Process in EKD represents activities that are needed to realize a goal. Concepts (Resources) that are Consumed and Produced by a Process, as well as Events that affect the performance of the process, are captured in the business process model (BPM).

4.2 i*

- **StrategyPlan: Partially Supported.** A complete SMBSC can be captured by both the SDM and SRM in i*, which respectively capture all dependencies within an organizational/unit and intentional elements (goals, soft-goals, tasks, etc.). An SDM within the boundaries of an actor consisting of four interdependent roles. Roles and the direction of dependencies can be set to include the four perspectives in the hierarchical order of SMBSC. An SRM model can be elaborated for each role to include all subgroupings as internal roles and all intentional elements.
- **Actor: Directly Supported**. In i* actor is an active entity (refers to generically any unit to which intentional dependencies can be ascribed) that carries out actions to achieve goals. Similarly the SMBSC actor (organization/unit) for whom a strategy map is defined entails undertaking actions to achieve goals.
- **StrategicGoal: Directly Supported**. i* includes the notion of soft goal, a goal whose criteria for satisfaction are not clear-cut, meaning satisfaction is described via contribution links from other elements. Thus, strategic goals can be represented as soft goals with the causality relationships across a strategy map describing how they can be sufficiently satisfied based on other goals or objectives.
- **StrategicTheme: Partially Supported**. i* does not include any such grouping notions. However it includes the notion of vulnerability over dependencies across actors, which is expressed through a classification of dependencies into open, committed and critical. Thus, a StrategicTheme can be introduced through critical goal and soft goal dependencies across roles representing SMBSC groupings, however this is limited to dependencies across roles and not within roles.
- **Group: Indirectly Supported**. i* can express groupings by introducing roles within an actor, as roles refer to the abstract behavior for a social actor within some specialized context of domain. Groups of goals/objectives can be attributed to roles and be represented within their boundaries. Group nesting can be supported by IsPartOf associations among roles.
- **Perspective, UniqueValueProposition**, Processes, Capital: Partially Supported. i* does not include any of these grouping notions, however, as per the above partial relation between Groups and roles, these groupings can be represented through a predefined set of roles representing the SMBSC groupings. Perspectives can be defined as role boundaries and subgroupings for each perspective can be represented through a role and the IsPartOf association towards their parent grouping.
- **Objective: Directly Supported**. A goal in i* expresses the intentional desire of an actor (hard goal) and includes clear criteria for satisfaction, making it measurable. Thus, Objectives can be captured as i* goals with the specifics of how the goal is to be satisfied described through the value activities relevant to the objective.
- **Measure, Milestone, Target: Partially Supported**. i* does not include notions like measure, milestone and target. However, these can be partially captured through heuristics by i) naming the goal to include the measure, ii) introducing a task to achieve the goal though a means-end link to (in i* a goal can be decomposed to sub goals through a task and then through task decomposition into other goals), iii) introduce goals through task decomposition, one for each milestone and target. An

objective along with its measures, milestones and target can be represented by an i* goal, decomposed through means-end link to a task stating the goal's achievement, and through task decomposition to goals (one for each milestone and target).

- **ValueActivitty: Directly Supported**. i* includes the notion of task, whose specification (e.g. how to be carried out) requires further decomposition. Information such as money and time can be expressed in i* through task decomposition to resources (need to be consumed for the task to be performed).

4.3 e^{3}value

- **StrategyPlan: Partially Supported**. In e^{3}forces, an actor of the organization type, or a group of actors (constellation) has Business Strategy as a property, describing the direction and scope of the organization's configuration and the position in its environment; however, the notion is not explicitly modeled nor its relationships with the other constructs. StrategyPlan exists in e^{3}value/forces, but it is not conceptualized, just represented as a text-based property of an actor.
- **Actor: Directly Supported**. In e^{3}value, actors are organizations, or end-customers perceived from their environment as economically independent entities, capable of taking economic decisions. An actor in e^{3}value/forces includes SMBSC. Actor that performs value activities, and extends it with the customer and constellation type actors, the first consuming what is provided by the value activities (value object) and the latter joining several organization-type actors cooperating to create a value.
- **StrategicGoal: Not Supported**. no corresponding notion is included.
- **Strategic Theme: Not Supported**. no corresponding notion is included .
- **Group: Partially Supported**. e^{3}value/forces allows groupings and subgrouping with respect to actors as it can be composite, consisting of other Actors. e^{3}value/forces considers an Actor as a concrete entity (e.g. an organization) or a role, such as Retailer referring to the abstract behavior within a business domain.
- **Perspective: Not Supported**. e^{3}value/forces distinguishes intra-enterprise alignment, i.e. where business strategy, e-services, process and IT/IS are analyzed within an enterprise. However, these parameters are also considered for alignment among several enterprises to provide integrated values, through integrated processes and IS/IT [9]. Thus e^{3}value/forces supports Perspective differently than of SMBSC.
- **Unique Value Proposition: Indirectly Supported**. e^{3}value/forces define 3 Business Strategy types for value object proposition (cost-leadership, differentiation and focus). However, they are not explicitly modeled in the ontology but they are represented as a text-based type of the Business Strategy property of an Actor.
- **Processes: Not Supported**. e^{3}value/forces is service-centered, meaning that it identifies services for realizing a business strategy and delivering values; the notion of process corresponds to activities needed to provide services (not in the ontology).
- **Capital, Objective, Measure, Milestone, Target: Not Supported**. e^{3}value/forces does not include intentional elements nor elements closely related to them.
- **ValueActivity: Partially Supported**. In e^{3}value, value activity is a set of operations yielding a profit to the actor(s) who perform it. As such, it is one of the core components of the ontology. Belonging to an actor, a value activity can be related to

a business strategy of the actor, and the corresponding strategic goals (not explicitly though), but these relationships are not conceptualized in e^3value/forces.

4.4 BMO

- **StrategyPlan: Partially Supported.** Notions of the SMBSC StrategyPlan are analyzed in terms of and compared to the Business Modeling Ontology (BMO). Overall, notions of the StrategyPlan in SMBSC are only partially supported by BMO primarily due to the lack of goal-related concepts.
- **Actor: Partially Supported**. In BMO actors are viewed from an internal perspective. The class Actor in BMO represents all actors except the one from whose perspective a BMO model is constructed [15]. Therefore, the BMO Actor corresponds to the SMBSC Actor i.e. that performs a value activity (not the Actor/Organization who defines the strategy).
- **StrategicGoal, StrategicTheme: Not Supported**. Strategic theme is in SMBSC an aggregation of Strategic Goals. BMO in general does not model the desired future state of the offering organization, e.g. strategic goals or themes.
- **Group: Partially Supported**. Groupings in SMBSC refer to a sub typing or part-of data abstraction, for example there exist groups of processes such as management processes, innovation processes etc., all of which are part of a higher level group such as Process. The meta-model of BMO does not contain any similar construct, however several BMO constructs can be de-composed into other constructs in that there exists a SetOf-association between the two.
- **Perspective: Partially Supported**. For SMBSC this captures the four perspectives of the strategy map template and includes aspects of goals (Objective, Measure etc.). However, perspectives in BMO do not model exactly the same thing, as goals are missing or not explicit in BMO. Nevertheless, the financial, customer and internal perspectives are directly supported in BMO with financial aspects, customer interface, and infrastructure management respectively. While the LearningAndGrowth perspective is said to correspond to Product (interface) in BMO, and mainly the value proposition [10], this can be debatable since they seem to capture different yet related concepts.
- **UniqueValueProposition: Indirectly Supported**. In BMO, a Value Proposition represents value for one or several Target Customers, i.e. how an organization differentiates what it offers from its competitors. A Value Proposition is decomposed into a set of Offerings. Each BMO Offering describes an elementary product or service, offered (directed) towards the target customers, which indirectly corresponds to the notion of SMBSC UniqueValueProposition.
- **Processes: Partially Supported**. In BMO this refers to Infrastructure Management. In particular, Processes map onto the Value Configuration part of Infrastructure Management in BMO, which describes the arrangement of activities and recourse that are necessary in order to crate value for the customer. One difference between BMO and SMBSC in this respect is that SMBSC Processes refers to groupings of goals for processes and related concepts such as Objective, Target, and Measure etc. while BMO Infrastructure does not contain any corresponding concepts.

- **Capital: Partially Supported**. The concept may vaguely be modeled using the BMO concept of Resource, which in BMO is divided into tangible, intangible and human resources. Another candidate is BMO Capability, where a Capability describes whether or not a particular needed Value Configuration can be applied by a particular company to provide the value proposition and if the appropriate resources (i.e. services and resources) are available.
- **Objective, Measure, Milestone, Target: Not Supported**. BMO in general does not capture the desired future state of the offering organization.
- **Value Activity: Indirectly**. BMO does not explicitly have a way of representing objectives/goals though it does represent Value Configurations. Attributes and relationships of BMO Value configuration, i.e. resources, actors etc. are similar to those of SMBSC Value Activity. Moreover, the BMO notion Activity is defined as "an action a company performs to do business and achieve its goals" [10].

5 Discussion, Conclusions and Future Work

The analysis shows that there is a considerable overlap between the EM approaches examined, both with regard to strategy and also among themselves. The analysis also highlights differences; most notably in terms of how core strategy notions like goals, means, etc. are represented. In this respect, EKD and i* include many constructs that correspond to goal-related notions, while BMO and e3value almost lack explicit representations of such notions. This partially validates the observation of Osterwalder that strategy models and business models deal with the same concepts but on different business layers [10]. Business models and strategy models complement each other in that the former are implementations of the latter, e.g. a company's strategy in terms of goals, means etc. are translated into value propositions, customer relations and value networks.

Overall none of the EM approaches examined directly supports all strategy notions. This limitation is expected as strategic notions are beyond the original intended scope of EM. Nevertheless, the analysis shows that either indirectly or partially examined EM approaches are able to model strategic notions.

The results of the study can be used in practice in several ways. Practitioners can get support for selecting the most appropriate EM language for modeling strategic aspects. For example, if strategic goals are in focus, i* or EKD are strong candidates, while BMO or e3value can be useful for designing and representing value configurations. Depending on the needs, several languages can also be used together in a complementary way. The results of the study can also support the development and extension of the EM approaches. For example, for EKD, an extension towards strategy formulations will provide a more streamlined development process offering inherent traceability from strategy modeling, through the different perspectives of an organization, to platform-specific implementations of IS.

Another aspect of the analysis lays in the use of the reference model UBSMM and its extensions to business strategy formulations i.e. UBSMM.SMBSC. Business strategy formulations are traditionally natural language-based, usually accompanied by schematic representations. The ambiguity of such formulations risks making analysis of EM approaches unattainable. This difficulty can be overcome by mapping EM constructs onto UBSMM.SMBSC constructs, since the meta-model provides clear semantics for a set of strategy notions. Thus, UBSMM.SMBSC facilitates understanding of strategic notions in EM approaches and reduces their ambiguity.

Future work will consolidate the analysis by including strategy notions found in other business strategy formulations such as VC and BOS, which are all part of UBSMM. Furthermore, other EM languages can be analyzed using the same framework used in this study.

References

1. Chandler, A. D.: Strategy and Structure. Cambridge, MA: The MIT Press (1962)
2. Bubenko J.A., jr., 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. Luftman, J.N., Derksen. B.: Key issues for IT executives 2012: Doing More with Less. MIS Quarterly Executive, 11, 4, pp. 207—218 (2012)
4. Kaplan, R.S., and Norton, D.P.: Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Harvard Business School Press, Boston, MA (2004)
5. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance, Free Press, New York (1985)
6. Giannoulis, C., Zdravkovic, J., Petit, M.: Model-driven Strategic Awareness: From a Unified Business Strategy Meta-model (UBSMM) to Enterprise Architecture. In 17th International conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD2012), LNBIP 113, Springer, p. 255-269 (2012)
7. iStar (i*) wiki, http://istar.rwth-aachen.de/ (last accessed on 07-07-2013)
8. Gordijn J, Akkermans JM, van Vliet JC.: Business modeling is not process modeling, In: Conceptual modeling for e-business and the web. LNCS, Vol 1921. Springer (2000)
9. Pijpers, V., Gordijn, J. e3forces: Understanding Strategies of Networked e^3value Constellations by Analyzing Environmental Forces. In:19th International Conference on Advanced IS Engineering (CAiSE), Springer, LNCS Vol. 4495, p. 188-202 (2007)
10. Osterwalder, A.: The business model ontology: A proposition in a design science approach. Institut d'Informatique et Organisation. Lausanne, Switzerland, University of Lausanne, Ecole des Hautes Etudes Commerciales HEC 173 (2004)
11. Barney J.: Types of Competition and the Theory of Strategy: Toward an Integrative Framework. J. Academy of Management Review, vol. 32, 11, 1231--1241 (1986)
12. Chan, K. W., Mauborgne, R.: Blue Ocean Strategy. Harvard Business Review Press, Boston (2005)
13. Giannoulis, C., Zdravkovic, J.: Linking Strategic Innovation to Requirements: a look into Blue Ocean Strategy. In 5th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM2012), Short Papers, CEUR-WS, p. 118-128 (2012)
14. Giannoulis, C.: Schema Integration Process for UBSMM. Technical Report, http://constantinos.blogs.dsv.su.se/files/2013/07/TowardsUBSMM_v02.pdf (2013)
15. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P., Gordijn, P., Gregoire, B., Schmitt, M., Dubois, E., Abels, S., Hahn, A., Wangler, B., Weigand, H., Towards a Reference Model for Business Models. In: 25th International Conference on Conceptual Modelling (ER' 2006), Springer Verlag, (2006)

Integration of Business Rules and Model Driven Development

Lauma Jokste

Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia
lauma.jokste@rtu.lv

Abstract. To help bridge the gap between Information system (IS) analysts and stakeholders, a Model-Driven Development (MDD) approach is proposed. A MDD approach uses models as primary development artifacts. Models increase the abstraction level of IS development and help to improve migration between various development phases. A MDD approach provides considerable benefits in the IS development domain, nevertheless this approach contains a variety of difficulties. This paper addresses MDD and specifically one of the models used in this approach – the Business Rules Model. Business rules are usually maintained in a textual form thus complicating their usage in the MDD approach. In this paper a new Business rules metamodeling language towards Business rules adaption for MDD is provided. As an input to MDD, Enterprise Modeling (EM) is used.

Keywords: Model Driven Development, MDD, Business Rules, Enterprise modeling, Enterprise Knowledge Development, EKD.

1 Introduction

One of the basic problems in the Information System (IS) development domain and requirements specification is ambiguity between system analysts and stakeholders [1]. System analysts tend to use models to determine the expected result of system functionality, but these models might never be directly used as a component of development process. Software development processes can effectively be improved by using models not only as visual means but also as a software development component. This leads us to *Model Driven Development* (MDD).

MDD is an approach to software development that refers to the systematic use of models and model transformations in entire software development- and runtime. The main idea of MDD is to automate the process of software development by using model transformations [2].

Organizational processes can be described with Enterprise Models. *Enterprise Modeling* (EM) represents behavior, structure, business goals, processes, concepts, actors and resources of an organization. In a context of software development EM includes representation of system requirements. In the EM approach all processes and components of organization are represented by using conceptual models [3].

This paper represents a Business rules metamodeling language which improves business rules integration with MDD. For describing the components and organization of an enterprise, the EM method *Enterprise Knowledge Development* (EKD) is chosen. EKD is an approach that describes an enterprise as a network of correlated business processes which collectively realize business goals [4]. EKD is selected as a widely used in both business and public sector. EKD has proved its effectiveness by providing a framework for stating, modeling and reasoning regarding pertinent knowledge in difficult problem situations [5].

The EKD approach includes several sub-models in which each sub-model describes the enterprise from different aspects: Goals model, Business rules model, Concepts model, Business process model, Actors and resource model and Technical component and requirements model [3]. In this paper the main emphasis is put on the *Business rules model* (BRM). Business rules are usually expressed in a natural business-like format and they might be liable to frequent changes, which complicates their usage in software development and maintenance. The business rules are means to which an organization is able to control the business, realize competitive strategies, promote the organization's policy and to comply with legal and other obligations [6]. Business rules describe the policies, laws and regulations of an organization. One of the main conditions for MDD is that all models and their components should ensure interoperability between all models used in MDD.

The objective of this paper is to present a Business rules metamodeling language which can be used to integrate business rules in the MDD approach and to discuss its clarity by giving an example case in which a Business rules metamodel is created. Main emphasis is put on a condition that the business rules expressed by provided metamodeling language should be easy perceivable for business people and usable in MDD.

In the context of this paper the metamodel is assumed to be a model's model that serves for the explanation and definition of relationships between various components of the applied model itself [7].

The research taken in this paper is argumentative. Preliminary validation and demonstration of the Business rules metamodeling language is performed using an example case of a student scholarship's system administration requirements, which are based on regulations of scholarship awards.

The remainder of his paper is structured as follows: a background about the business rules and Business rules model is given in section 2. The proposed graphical metamodeling language is given in section 3. In section 4 an example case of Business rules metamodel is demonstrated. In section 5 concluding remarks are given.

2 Business rules and Business rules model

Organization's business rules are usually expressed and maintained in a natural language format. For example, scholarship regulation can contain a business rule:

Only students who have passed all their exams in the previous session on the first attempt are eligible for the Scholarship.

In order to use business rules in MDD, it is necessary to distinguish concepts, attributes, conditions and actors in the rule. A way how company's business rules are described may depend on many circumstances, for example – different lawyers can draw up the same rules and interpret them differently. In order to prevent such situations, we appoint that every rule should be transformed to a standardized natural language form before it is integrated in MDD. For this purpose business rules notation which is based on formal English – RuleSpeak is chosen. RuleSpeak was first developed in 1990 by Business Rules Solutions (BRS), LLC. It is widely used among business people [8] and even been tried for IS requirements specification [9]. It was decided to choose RuleSpeak as a base for a Business rules metamodeling language, because it has a clear definition and defines well structured business rules sentence forms, which can be adapted to Business rules metamodel and MDD.

RuleSpeak is a set of practical guidelines for expressing business rules in unambiguous and well-structured English, which helps to improve communication about business rules among business people, business analysts and IT professionals [8]. The basic concept of RuleSpeak states that every rule should include one of these two words: "must" and "only". Instead of "must", "may" can be used but only when combined with the word "only". RuleSpeak guidelines defines the best practice for business rules sentences structure, how to avoid a redundancy and express rule clearly and unambiguously interpretable. Detailed RuleSpeak guidelines can be found in [10] and [11].

A business rule which is expressed by using RuleSpeak guidelines would look as follows:

The scholarship may be assigned only if student has passed all his exams in previous session on the first attempt.

In the following Business rules model rules are expressed according to a RuleSpeak specification (see in Figure 2.). The Business rules model is developed by using an example case, the requirements of a student scholarship system administration, which is based on regulations of scholarships awards. According to the EKD method, business rules are motivated by goals, they cause business processes and are based on concepts defined in a concept model [12]. In the Business rules model each rule consists of one sentence. In the Business rules metamodel each rule should be divided into separate objects, which our provided graphical business rules metamodeling language supports.

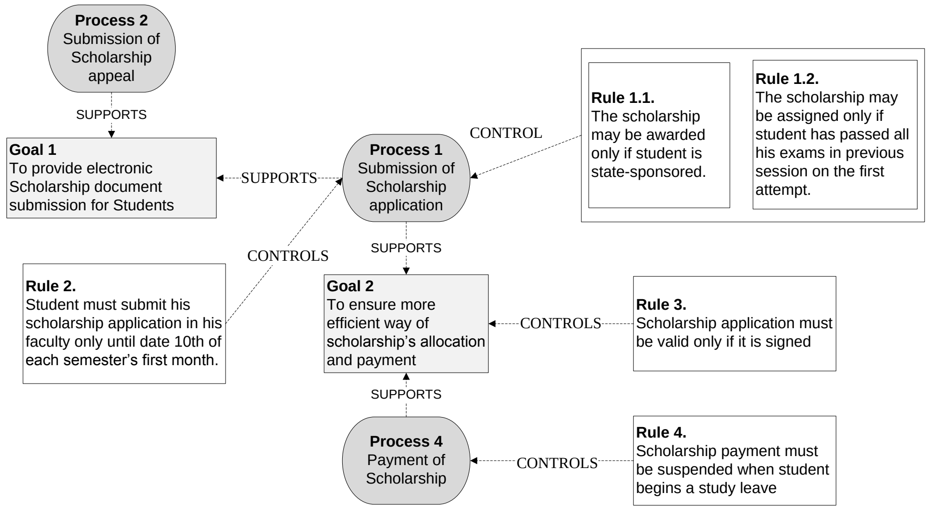


Fig. 1. The Business rules model example case based on the EKD method.

This Business rules model example case is a simplification of real life. There might be more business rules to consider than those displayed in figure 1.

3 Definition of graphical business rules metamodeling language

In this section a graphical metamodeling language is defined. Metamodeling language is based on metamodeling concepts called GOPPRR: graphs, objects, properties, ports, relationships and roles [13]. The Business rules metamodeling language consists of two categories: objects (see Tab. 1) and relationships between them (see Tab. 2).

Objects in a metamodel are connected with links. Every link contains two roles: the source and the target. The source points to an object where the link begins. The target points to an object to which the link is lined. Roles define in which direction the link can be drawn – either directions or only one specific direction. In the provided metamodeling language, roles define from which to which object a link may be drawn. Strictly defined roles and relationships are an important condition for Business rules metamodel to be compatible with other models used in MDD and to be explicitly used in the code generation. Links and roles are defined in Table 2.

Table 1. Defined metamodeling language objects

Object	Description
Rule	A modeling component contains rule name and marks a beginning point of each business rule. Every rule always starts with a *Rule* object. It is necessary for selecting certain rules and for integration with other models used in code generation.

Concept	Concepts define “things” and “phenoma” which are used in all other models [12]. *A Concept* is a modeling component which is used to characterize all concepts used in a Business rules metamodel. Every concept can be defined only once. Concept is, for example: *The Scholarship*.
Action	The RuleSpeak specification says that each business rule must contain one of the words “must” and “may” when describing an action in the business rule, hence the modeling component *Action* is separated in two sections, where the first section contains the *keyword* and the second section *the action* (verb). The *keyword* section contains either predefined values “must” or “may”, as well as form of denial: “must not” or “may not”. Action is, for example: *may be assigned*, where *may* is keyword and *be assigned* – action.
Additional word	*An* a*dditiona*l word is a word or expression, which helps to create a coherent and readable structure of the business rule. An *Additional word* contains predefined words and word combination lists with values such as *only if, only when, only, if, and, or, in, when, then* etc. *Additional words* usually do not affect business rules modeling functionality but are mostly used to make a logical interpretation of the rule.
Condition	*A Condition* is a composite object, which consists of 4 parts: 1) *Attribute*; 2) *Concept* (*Attribute source*); 3) *Operator*; 4) *Attribute value*. *The Attribute* is a text input field for an attribute name. The *Concept* field must contain one of existing concepts. The *Operator* is a value that can be expressed as a logical operator in code, for example: is equal, is until, is at least etc. The *Attribute value* is a text input field for the value of attribute. Condition is, for example: *Exam attempt is first,* where *Exam* is a concept (attribute source), *attempt* – an attribute, *is* – an operator and *first* – an attribute value.

Table 2. Relationships in graphical Business rules metamodeling language

To: → From: ↓	Rule	Concept	Action	Additional word	Condition
Rule	0	1	0	1	0
Concept	0	0	1	1	0
Action	0	1	0	1	0
Additional word	0	1	0	0	1
Condition	0	0	0	1	0

Relationships between modeling components in a Business rules metamodel are described by a connectivity matrix (See table 2). Value ‘0’ means that a relationship from one modeling component to other doesn't exist and value '1' means that a relationship between object "From" to object "To" is defined. For example, a link from *Rule* to *Concept* exists, while a link from *Concept* to *Rule* is not possible.

4 Example case

To a purpose to demonstrate provided Business rules metamodeling language, a Business rules metamodel example case is created. Coherence between the Business rules model and its metamodel are shown in figure 2.

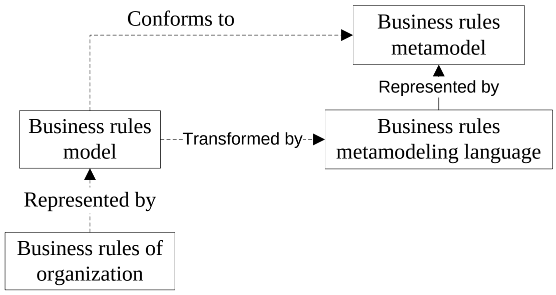


Fig. 2. Coherence between a Business rules model and its metamodel.

In the presented Business rules metamodeling language each rule is designed as an independent structure, where the same objects between different rules are not connected with visible links. Every object or object component should contain a unique identification thus ensuring that, for example, the same named concepts are actually one and the same object. This approach allows building a transparent and easily modifiable metamodel while ensuring metamodel usability in MDD.

Each rule must contain at least one concept, one action and one condition object. Figure 3 shows an example of how to modify business rules expressed in RuleSpeak form to a business rules form in metamodel, using our graphical metamodeling language.

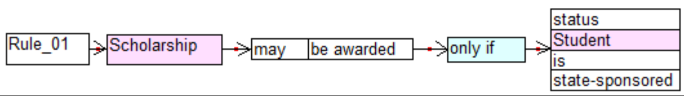
RuleSpeak form	Scholarship may be awarded only if student is state-sponsored.
Form in metamodel	

Fig. 3. Graphical Business rules metamodel representation of example no. 1.

The Business rule expressed in metamodel form contains an extra word *status* (see Fig. 3); it is compulsory that the *condition* object contains an attribute, because it is used to connect the Business rules metamodel with other models. If this rule would be generated back to a formal expression, it would look as follows:

Scholarship may be awarded only if Student status is state-sponsored.

Business rules back-generation from metamodel to their formal expression is a metamodeling tools functionality. It is easy implementable and usable for different metamodeling tools.

The operator field contains the value *is*. For graphical representation, textual operators are recommended, though in code textual operators can easily be defined as logical operators, for example: [is] = [=], [is until] = [≤], [is greater than] = [>], [is at least] = [≥], [is not] = [≠] etc.

RuleSpeak form	Scholarship payment must be suspended when student begins a study leave.

Form in metamodel

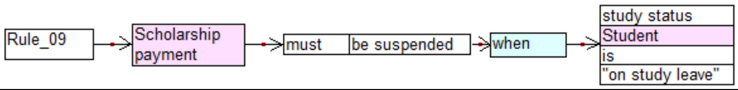


Fig. 4. Graphical Business rules metamodel representation of example no. 2.

Figure 4 demonstrates a representation of an event-action business rule. If this rule would be generated back to formal expression, it would look as follows:

Scholarship payment must be suspended when Student study status is "on study leave"

RuleSpeak form	Student must submit the scholarship application to faculty only until the 10th of each term's first month.

Form in metamodel

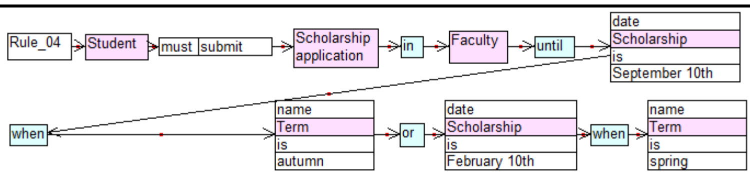


Fig. 5. graphical Business rules metamodel representation of example no. 3.

Figure 5 demonstrates a representation of a derivative business rule. A phrase "*until the 10th of each term's first month*" might be ambiguous for developers, therefore to make it unmistakably clear, the number of terms is specified with *each term's first month*. Consequently, if this rule is generated back to a formal expression, it looks like this:

Student must submit scholarship application in Faculty until Scholarship application dates is September 10^th^ when Term name is autumn or Scholarship application date is February 10^th^ when Term name is spring.

In both examples that figures 4 and 5 demonstrate, the business rule sentence form has changed, preserving an essence of the business rule in its RuleSpeak sentence format.

Business rules metamodeling language can be used in different metamodeling platforms, for example Eclipse Modeling Framework (EMF) [14], ADOxx [15], MetaEdit+ [13] and others. It was decided to choose MetaEdit+ as it has strengths in metamodeling and it provides a full functionality for defining a graphical metamodeling language [16].

A fragment of the developed Business rules metamodel in MetaEdit+ is shown in Figure 6.

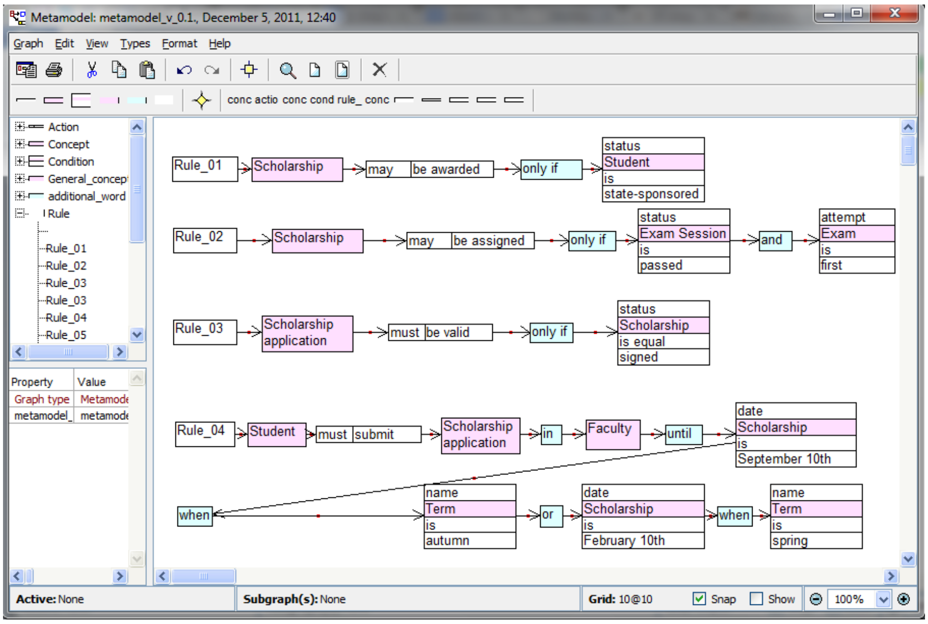


Fig. 6. Fragment of the Business rules metamodel in MetaEdit+ .

5 Concluding remarks

This paper presents a business rules metamodeling language which can be used to create Business rules metamodels for MDD. The most important challenges for business rules usage in MDD is the fact that business rules are usually expressed in a business-friendly manner and are not guided to specific instructions how the business rule should be written. The goal is to clarify the logic of a business rule and to transform it to more rigorous form thus making business rule usable in MDD.

Provided graphical business rules metamodeling language has some similarities with Decision Model developed by Halle von Barbara. Both approaches ensure well-formed, predictable, stable and maintainable expression of the business rules [17]. Unlike Decision Model, Business rules metamodeling language is tended to a logic of the separate objects in the business rule, while Decision Model are based on a business logic in general. But both – Decision Model and Business rules metamodeling language can be anchored to any other models and in the same time can be maintained independently of them.

The provided Business rules metamodeling language is experimentally validated with an example case and ensures that business rules can be transformed from their natural expression which is formalized by RuleSpeak to the Business rules metamodel and vice versa without losing interpretation of the business rule, even if the structure of the rule is slightly changed. For the time being, interpretation changes can be evaluated subjectively and there is no appropriate statistical method for measuring an interpretation of textually expressed business rules. Business rules expressed by provided Business rules metamodeling language can be further used for code generation in MDD.

However, a more detailed case study is necessary to improve that Business rules metamodeling language can ensure that any kind of business rules can be expressed by this language.

References

1. Kirikova M., Bubenko J.A. (1994), Software Requirements Acquisition through Enterprise Modeling. Paper presented at 6th Int. Conference on Software Engeneering – SEKE'94, Jurmala, Latvia.
2. Sanchez P., Ana Moreira A., Fuentes L. et al (2010) Model-driven development for early aspects, Information and Software Technology 52 p.249–273.
3. Zikra I., Stirna J., Zdravkovic J. (2011.), Analyzing the Integration between Requirements and Models in Model Driven Development, Springer Verlag, Berlin pp 342 – 356.
4. Kavakli V., Loucopoulos P. (1999), Goal-Driven Business Process Analysis Application in Electricity Deregulation. Information Systems 24, pp 187-207.
5. Gints Stale, Ivars Majors (2012), The Application of EM for Knowledge Flow Analyses and the Development of an Educational IT Ecosystem, Vol. 933. Paper presented at the 5th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling, Rostock, Germany, 7-8 Nov 2012.
6. von Halle B., Goldberg L., Zachman J. (2006), Business Rule Revalution. Running Business the right way. Happy about, California - P. 221.
7. Génov G. (2009), What is a metamodel: the OMG's metamodeling infrastructure // Modeling and metamodeling in Model Driven Development, Poland, Warsaw, 14-15 May 2009.
8. RuleSpeak official homepage http://www.rulespeak.com/en/ Accessed 26 Jun 2013.
9. Kapočius K., Danikauskas T. (2006), The Use of Business Rules for the Specification of Dynamic Aspects of IS. ISSN 1392 – 124X, Information Technology and Control, Vol.35., No 3A, pp 327-332.
10. Ronald G. Ross, Basic RuleSpeak Guidelines, Business Rules Solutions, LLC, version 2.2. (2001.-2009.).
11. Ross R. G. (2001.-2009.), RuleSpeak Sentence Forms, Business Rules Solutions, LLC, version 2.2.
12. Bubenko, J., Persson, A., Stirna, J. (1996), User Guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns, no IST 2000-28401, Sweden, ftp://ftp.dsv.su.se/users/js/ekd_user_guide.pdf Accessed 10 Jun 2013.
13. MetaEdit+ Domain-Specific Modeling environment // Metacase (2012). http://www.metacase.com/MetaEdit.html last accessed 10.06.2013.

14. Eclipse Modeling Framework Project, http://www.eclipse.org/modeling/emf/ Accessed 27 Aug 2013.
15. ADOxx, http://www.adoxx.org/live/home Accessed 27 Aug 2013.
16. Kern H. (2008) The Interchange of (Meta)Models between MetaEdit+ and Eclipse EMF Using M3-Level-Based Bridges. Paper presented at the 8th OOPSLA Workshop on Domain-Specific Modeling conference. Nashville, TN, USA, 19-20 October 2008
17. von Halle, B., Goldberg, L. (2009) The Decision Model: A Business Logic Framework Linking Business and Technology. Auerbach Publications.