CEUR Workshop Proceedings
http://ceur-ws.org
ISSN 1613-0073

Vol-344

© 2008 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. Re-publication of material from this volume requires permission by the copyright owners.

CAiSE'08 Forum

Proceedings of the Forum at the CAiSE'08 conference

http://www.lirmm.fr/caise08

Montpellier, France, June 18-20, 2008

Volume Editors

Zohra Bellahsène, LIRMM CNRS-University of Montpellier 2, France
Carson Woo, University of British Columbia, Canada
Ela Hunt, Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland

CAiSE'08 Workshops Proceedings Editors

Ela Hunt, Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland
Xavier Franch, Universitat Politècnica de Catalunya, Spain
Remi Coletta, LIRMM CNRS-University of Montpellier 2, France

Table of Contents

---

submitted by E. Hunt, 28-May-2008
published on CEUR-WS.org, XX-XX-2008

Preface

CAiSE 2008 was the 20th in the series of International Conferences on Advanced Information System Engineering. This edition continues the success of previous conferences, a success largely due to that fact that, since its first edition, this series has been evolved in parallel with the evolution of the importance of information systems in economic development. CAiSE has been able to follow, and often to anticipate, important changes that have occurred since 1978 when the first CAiSE conference was organised by Arne Sølvberg and Janis Bubenko. In all these years, modern businesses and IT systems have been facing an ever more complex environment characterized by openness, variety and change. Furthermore, enterprises are experiencing ever more variety in their business in many dimensions. In the same way, the explosion of information technologies is overwhelming with a multitude of languages, platforms, devices, standards and products. Thus enterprises need to manage an environment to monitor the interplay of changes in the business processes, in information technologies, and at the ontological level, in order to achieve a sustainable development of their information systems. Enterprises must enter the era of Sustainable Information Systems to face the important developmental challenges. During all these years, CAiSE researchers have been challenged by all these changes, and the CAiSE conferences provide a forum for presenting and debating important scientific results. In fact, CAiSE is positioned at the core of these tumultuous processes, hosting new emerging ideas, fostering innovative processes of design and evaluation, developing new information technologies adapted to information systems, creating new kinds of models, but always being subject to rigorous scientific selection. And so, the previous CAiSE conferences have largely contributed to develop a sustainable conceptual platform for information systems engineering, well suited to the era of Sustainable Information Systems. This is the main theme of this conference.

The CAiSE Forum 2008 proceedings represent a collection of 22 excellent short research papers and five demos which were presented at four poster sessions during the conference. The selection of Forum papers was very stringent, due to the very high standard of the submitted papers. Several high-quality papers initially submitted to the CAiSE conference were selected for the CAiSE Forum to stimulate open discussions of high-quality on-going research.

As editors of this volume, we would like to express our gratitude to the program board, the program committee and external reviewers for their efforts in providing very thorough evaluations of the submitted CAiSE papers. We also would like to thank Richard van de Stadt for his very effective support during the paper evaluation and proceedings preparation. Finally, many thanks to Google, Microsoft, ERCIM, UM2 and CNRS, Languedoc-Roussillon Region and The City Hall of Montpellier for their sponsorship.

May 2008

Ela Hunt and Carson Woo, CAiSE'08 Forum chairs

Zohra Bellahsene, CAiSE'08 PC chair

Xavier Franch, CAiSE'08 Workshop co-chair

Remi Coletta, CAiSE'08 local organising committee chair

CAiSE'08 Programme Committee

Agerfalk Pär, Ireland
Akoka Jacky, France
Bajec Marko, Slovenia
Benatallah Boalem, Australia
Boudjlida Nacer, France
Bouzeghoub Mokrane, France
Casati Fabio, Italy
Castano Silvana, Italy
Castro Jaelson, Brazil
Cauvet Corinne, France
Dori Dov, Israel
Dumas Marlon, Australia
Embley David, USA
Evermann Joerg, New Zealand
Falcao e Cunha Joao, Portugal
Franch Xavier, Spain
Front Agnès, France
Giorgini Paolo, Italy
Godart Claude, France
Gordijn Jaap, the Netherlands
Hacid Mohand-Said, France
Mouratidis Haris, UK
Mylopoulos John, Canada
Norrie Moira, Switzerland
Oberweis Andreas, Germany
Olivé Antoni, Spain
Parsons Jeffrey, Canada
Petit Michaël, Belgium
Pigneur Yves, Switzerland
Poels Gert, Belgium
Prakash Naveen, India
Proper Erik, the Netherlands
Ralyte Jolita, Switzerland
Regnell Björn, Sweden
Reichert Manfred, the Netherlands
Roantree Mark, Ireland
Roel Wieringa, the Netherlands
Rosemann Michael, Australia
Rossi Matti, Finland
Rossi Gustavo, Argentina
Ryan Kevin, Ireland
Saeki Motoshi, Japan

Halpin Terry, USA
Hauswirth Manfred, Ireland
Heymans Patrick, Belgium
Jarke Matthias, Germany
Jeusfeld Manfred, the Netherlands
Johannesson Paul, Sweden
Jonkers Henk, the Netherlands
Jorgensen Havard, Norway
Kaschek Roland, New Zealand
Kirkova Marite, Latvia
Krogstie John, Norway
Laleau Régine, France
Lankhorst Marc, Brazil
Leite Julio, Brazil
Lyytinen Kalle, USA
Maiden Neil, UK
McBrien Peter, UK
Mirbel Isabelle, France
Missikoff Michele, Italy
Salinesi Camille, France
Shan Tony C., USA
Siau Keng, USA
Sindre Guttorm, Norway
Snoeck Monique, Belgium
Soulé-Dupuis Chantal, France
Stirna Janis, Sweden
Taniar David, Australia
Thalheim Bernhard, Germany
Toumani Farouk, France
Tsalgatidou Aphrodite, Greece
Valduriez Patrick, France
Van Der Aalst Wil, the Netherlands
Vasilecas Olegas, Lithuania
Wand Yair, Canada
Weske Mathias, Germany
Woo Carson, Canada
Yu Eric, Canada

Method Tailoring as Negotiation

Fredrik Karlsson [1]

[1] Dept. of Informatics (ESI), Methodology Exploration Lab
Örebro University, SE-701 82 Örebro, Sweden
fredrik.karlsson@esi.oru.se

Abstract. The need for method tailoring is widely accepted in the field of information systems development methods. Today much attention has been devoted to viewing method tailoring either as (a) a highly rational process with the method engineer as the driver where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. In this paper we view method tailoring from a negotiation perspective using Actor Network Theory. Our narrative examples depict method tailoring as a more complex process than either (a) or (b) show.

Keywords: Method tailoring, Method Engineering, Method-in-Action, Actor Network Theory, Negotiation

1 Introduction

As Fitzgerald et al. [1] conclude 'it is now widely accepted that [information systems development] methods should be tailored to the actual needs of the development context.' This statement is acknowledged by, what appears to be, the two schools of information systems development methods [2]: method engineering [3] and method-in-action [4].

These two schools view method tailoring as (a) a highly rational process with the method engineer as the driver, where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. However, Riemenschneider and Hardgrave [5] show that acceptance of methods is dependent of 'the opinions of developers' coworkers and supervisors toward using the methodology.' Madsen et al. [6] conclude that far too 'little research has addressed the details how the unique and local method emerges and why it takes the form it does.' They unfold the emergent method through three different perspectives, one of them being the interactive process perspective. But, still the details of the negotiation aspect of the emergent method are not evident. Consequently, method tailoring is poorly understood as a social activity. The purpose of this paper is to exemplify method tailoring as negotiation, the interplay between humans and artifacts. For this purpose we apply Actor Network Theory (ANT) [7].

2 Actor Network Theory as Research Approach

The research presented in this paper is an interpretive investigation of method tailoring. The empirical base is an information systems development project undertaken in a public organization. A content management system was implemented and the organization's existing web site was migrated to the new platform. The author participated as one of thirteen regular team members in this project, spending 1 200 man hours during eighteen months. The project was carried out at three different locations, two locations inside the public organization and one location at a consulting firm. The actors are categorized according to the main roles they had during the project: project manager, systems administrator, implementer, requirements engineer and content manager.

Walsham [8] concludes that ANT is both a theory and a research method. ANT contains a conceptual framework to use during data collection and analysis. Our interpretation of this framework is inspired by Walsham [8] and our framework has a specific characteristic; it does not contain any a priori distinction between human and non-human actors. Both concepts are viewed as active makers of actor networks and are specializations of the actant concept. Furthermore, networks are changed through translation, an establishment of a new relation between actors. Latour [7] describes it as coexisting in a network to achieve of a common goal, for example when the system analyst and the tester agree to document the information system's external behavior using use cases. Often translation requires enrollment where an actor seeks to influence how another actor should act, for example to use a specific technique. Enrollment and translation can result in inscriptions, where interests are inscribed into written material or technical systems.

This research is based on several data sources from the systems development project: intermediate project artifacts, e-mails and project notes. Intermediate project artifacts show what has actually been documented. Furthermore, these artifacts are time stamped making it possible to analyze how they have evolved over time. That is to say, they show the result of method tailoring. The e-mails and the project notes are used to capture tailoring decisions and arguments behind these decisions. Consequently, these documents contain traces of the team members' different viewpoints about the emergent method.

3 Method Tailoring – The Analysis

3.1 The First Example

Our first examples concerns method support for requirements engineering. Much of the requirement work during this project concerned the web page templates. Simple sketches were used to capture layout and functional requirements. This work was carried out by the requirements engineer together with the content managers. However, when it came to more advanced web page templates, containing interaction

possibilities, static sketches did not provide enough information. The method's inadequacy is salient in the e-mail conversation between the requirements engineer and the implementers, concerning the design of forum templates: 'what are the options in this listbox', 'how are these [web] pages linked to each other', and 'how shall we display the thread overview.' These quotes illustrate the problems to capture and communicate the design.

In an e-mail the requirements engineer concludes 'that several options exist', but she suggests the use of either use cases or storyboards. Both her suggestions are enrollments of techniques used in other methods. In addition, through her proposal she tried to enroll the content managers and the implementers in use of one of these techniques. Two of the implementers express that they prefer storyboards to use cases: 'use cases tend to become cluttered … difficult to show how [web] pages are related.' Consequently, the two implementers seem to be concerned with the amount and the type of details that are capture if they chose to use cases. Furthermore, the implementers needed to know how web pages were related to each other in order to determine possible navigation paths and when to provide a specified functionality. According to the implementers use cases insufficient means to this end.

In an additional e-mail to the implementers, the requirements engineer referred to a discussion with some of the content managers (it is unclear with whom). She stated that the content managers preferred storyboards as well, 'since they are easier [to read].' Hence, the three actor groups have made a translation and later use of documents show an inscription in the method.

3.2 The Second Example

Our second example concerns method support for testing. The content managers, who are responsible for testing the web page templates, had just begun their work. They either gave oral reports to the implementers or documented the flaws on post-it notes. At initial stage the implementers concluded that the method lacked proper support for test reports and tracing flaws.

One of the implementers addressed this issue with the content managers via e-mail. He argued in favor of one shared artifact for documented bugs, since 'I believe we cannot keep trace of all the bugs we have found.' The content managers' answers to this e-mail can be divided as follows: (1) two actors did acknowledge the problem (2) one actor did not acknowledge this as a problem, and (3) two actors did not answer. In an e-mail reply to the content managers the implementer proposes 'a simple Excel sheet … on a shared domain.' The implementer received three positive replies to this enrollment. In one reply we find '… we have to discuss the layout [of this document].' The person who did not acknowledge the need for a formal test report document did not answer the implementer's e-mail.

The implementers discussed the need for a formal test report document with the project manager, arguing that they were not able to manage the change requests with the current way of working. Hence, they enrolled him in the method tailoring process. At this meeting the implementers presented a document template, which was later e-mailed to the content managers. The e-mail conversation shows a translation between

the implementers and the project manager, where the latter act as delegate: 'I believe this [document layout] looks good.'

At a later meeting a decision was taken to use this document template. This decision was explicitly supported by the implementers as well as two of the content managers. Consequently, an inscription in the method was made. When analyzing the use of the template we identified that two of the content managers disagreed with the decision and the inscription. They continued to report bugs via post-it notes, e-mail and orally. The other actors did not express any problems with the new modifications. Hence, they aligned with the method evolvement.

4 Conclusion

In this paper we have depicted method tailoring from a negotiation perspective, using Actor Network Theory. Most of the existing literature view method tailoring as either (a) a highly rational process with the method engineer as the driver where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. Our narratives show that method users are not passive information providers during method tailoring. But these narratives also show that method tailoring is not about individual choices either. Accordingly, this is clearly an interesting venue for further research, investigating negotiation patterns in method tailoring and how they can be used in construction of approaches and tools.

References

1. Fitzgerald, B., Russo, N.L.,O'Kane, T.: Software Development Method Tailoring at Motorola. Communications of the ACM 46(4), 65-70 (2003)
2. Ågerfalk, P.J.,Fitzgerald, B.: Exploring the Concept of Method Rationale: A Conceptual Tool for Method Tailoring. In: Siau, K. (ed.) Advanced Topics in Database Research, pp. 63-78. PA: Idea Group, Hershey (2006)
3. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
4. Fitzgerald, B., Russo, N.L.,Stolterman, E.: Information Systems Development - Methods in Action. McGraw-Hill, London (2002)
5. Riemenschneider, C.K., Hardgrave, B.C.,Davis, F.D.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Transactions on Software Engineering 28(12), 1135–1145 (2002)
6. Madsen, S., Kautz, K.,Vidgen, R.: A framework for understanding how a unique and local IS development method emerges in practice. European Journal of Information Systems 15(2), 225-238 (2006)
7. Latour, B.: Reassembling the social: an introduction to actor-network-theory. Clarendon lectures in management studies Oxford University Press, Oxford (2007)
8. Walsham, G.: Actor-Network Theory and IS Research: Current Status and Future Prospects. In: The IFIP TC8 WG 8.2 international conference on Information systems and qualitative research, Chapman & Hall, New York (1997)

Supporting Agile Development with Participative Enterprise Modeling

Janis Stirna[1], Marite Kirikova[2]

[1] Jönköping University, PO Box 1026, SE-551 11, Jönköping, Sweden
janis.stirna@jth.hj.se

[2] Riga Technical University, 1 Kalku Street, Riga, LV 1658, Latvia
marite.kirikova@cs.rtu.lv

Abstract. Agile Modeling provides a set of best practices of "light-weight" modeling to support the modeling process on a macro level within the agile development teams. The objective of this paper is to analyze the potential of using Enterprise Modeling in agile development projects to address some of the existing challenges of agile projects.

Keywords. Enterprise modeling, agile modeling, agile development

1 Introduction

The Information System (IS) development community has been trying out and adopting various agile development approaches such as, e.g., eXtreme Programming (XP) [1] and SCRUM [2]. One of the strengths of agile approaches is their flexibility and ability of dealing with change efficiently. Their philosophy is development of only those artifacts that are directly related to the software product. Agile approaches typically do not prescribe which methods, languages, and tools are to be used. Instead, the emphasis is on choosing the simplest, most effective and, therefore, the most cost effective ones. To support the modeling process on a macro level within the agile development teams Agile Modeling (AM) [3] was developed. AM provides best practices of "light-weight" modeling and suggests active stakeholder involvement.

However, gathering requirements in agile methods is targeted exclusively to software development needs. The relationship between knowledge of enterprise stakeholders and software artifacts is tacit and contributes only to the software development process, not to the enterprise knowledge development on a larger scale. This phenomenon does no permit to utilize all possible benefits of requirement gathering. Enterprise Modeling (EM) [4] on the other hand has proven to be a practicable instrument for creating an integrated and negotiated model describing different aspects of an enterprise. [5] show that EM can be used for two main types of objectives – (1) developing the business, e.g. developing business vision, strategies, redesigning the way the business operates, developing the supporting information systems, or (2) ensuring the quality of the business, e.g. sharing the knowledge about the business, its vision, the way it operates, or ensuring the acceptance of business decisions through committing the stake-holders to the decisions made.

In this paper we analyze the potential of using EM, taking the EKD (Enterprise Knowledge Development) approach as an example, in agile development projects. The research approach is conceptual and argumentative based on findings of a number of qualitative research studies [5, 6, 7, 8]. The rest of the paper is structured as follows. Sections 2 and 3 summarize EM and agile development respectively. Section 4 integrates the AM and EM while section 6 presents concluding remarks.

2 Background to EKD

EKD is an EM method for developing, acquiring, and communicating early, enterprise knowledge, such as strategies, goals, or requirements, by a structured, iterative, working and modeling approach [4]. The Enterprise Model consists of set of structured, goal/problem driven models to be used for structuring and representing organizational knowledge – the modeling product. In a full enterprise model, the relationships between Enterprise Model components (see table 1) play an essential role because they allow tracing decisions. The EKD modeling process guides the knowledge acquisition, analysis, and representation. Knowledge acquisition and modeling needs to be participatory to consolidate multiple stakeholder views.

	Goals Model (GM)	**Business Rules Model (BRM)**	**Concepts Model (CM)**	**Business Process Model (BPM)**	**Actors and Resources Model (ARM)**	**Technical Component & Requirements Model(TCRM)**
Focus	Vision and strategy	Policies and rules	Business ontology	Business operations	Organizational structure	Information system needs
Issues	What does the organi-zation want to achieve or to avoid and why?	What are the business rules, how do they support organiza-tion's goals?	What are the things and "phenomena" addressed in other sub-models?	What are the business processes? How do they handle information and material?	Who are responsible for goals and process? How are the actors interrelated?	What are the business requirements to the IS? How are they related to other models?
Com-po-nents	Goal, prob-lem, external constraint, opportunity	Business rule	Concept, Attribute	Process, external proc., information set, material set	Actor, role, organizational unit, individual	IS goal, IS problem, IS requirement, IS component

Table 1: **Overview of the sub-models of the EKD method**

3 Background to Agile Development Approaches

Agile approaches all share the same values, e.g. to be flexible and to be able to deal with changes during the course of a software project. They provide a set of principles, techniques and best practices for iterative and incremental IS development, but they do not explicitly tackle modeling in terms of how to model and what. To address this, AM proposes to integrate more traditional modeling approaches with the ideas of agile development. See [6] for additional evidence about the suitability of AM in practice.

Agile approaches primarily focus on the development of a software system. The underlying assumption of agile projects is that the customer knows what kind of system is needed, what are its features, and who will use it and how. In practice,

however, IS development is a part of some business development or change project. Moreover, the developers and the stakeholders may have to face a number of hard organizational problems requiring structured exploration of various business and IS development alternatives. A common situation is to set up business goals of the new IS system before the development project is commissioned. If this is done without involving the developers of the new IS, it immediately puts them at a disadvantage because not all information is properly communicated to them. The agile approaches try to address this challenge by active stakeholder involvement and having a customer representative on the development site. However, these practices might not be enough because vital knowledge has to be re-acquired. Agile teams are also not used to requiring the business people to think systemically about their business and to connect their business needs to the new IS. Mush of this knowledge is scarce and tacit – it lies in the heads of a few stakeholders. There might also be different opinions, especially concerning the future plans of the organization, which, therefore, need to be consolidated, made explicit, and transferred to the agile team. While XP suggests "customer on site", in reality, however, only a few stakeholder types can be kept on site. High level managers such as CEOs or CFOs need to be engaged differently.

4. Using EM in Agile Development Projects

This section analyzes the potential of using EM, taking EKD as an example, in agile development projects. We discuss the objectives and the compatibility of Agile Model Driven Development (AMDD) [9] and EKD.

In both AM and EM the group work is used to achieve consensus, understanding and commitment concerning the scope and requirements of the development project. Furthermore, [8] shows close correspondence between artefacts of AM and EM.

AMDD stage	EM support
Iteration 0: Envisioning	
Initial Requirements modeling (identify high level scope and an initial requirements stack)	An EM seminar with all key stakeholders to establish the business goals of the system, to explore the business requirements and to set the overall strategy of the project. The intangible benefit is the consensus about these issues.
Initial Architecture Modeling	An EM seminar to identify IS architecture on a crude level.
Iteration 1-n	
Iteration Modeling: Thinking Through What You'll Do This Iteration	EM to elaborate detailed issues concerning the iteration. E.g. elaboration of the business process that needs to be supported.
Model storming (work through specific issues, just in time (JIT) modeling, stakeholders actively participate)	Short EM events in the development team to resolve specific modeling issue that they have involving stakeholder representatives that are available on site. Involving other stakeholders would have to be planned in advance.
Executable Specification via Test Driven Development	EM supports this by explicit linking of business goals, rules, and requirements which can serve as measurable constraints.

Table 2. **Combining AMDD with the EKD modeling process**

Agile projects using EM should focus on the business vision and business requirements. This can be done by integrating prototyping approaches with business analysis to explore alternatives of supporting business goals and processes by IS components and features. Using EM to capture the business knowledge pertinent to

the IS development project is not the same as BRUF (big requirements up front), which contradicts with the principle of iterative and incremental development and is argued against by many practitioners. AMDD is a framework for iterative and incremental modeling. Table 2 takes the EKD modeling process as basis and shows how EM is able to contribute to AMDD.

Concerning the specifics of integrating the agile way of working with EM we would like to propose the following recommendations: elaborate multiple perspectives iteratively, involve different stakeholder types, link other models and designs with the Enterprise Model, and use simple tools to support Agile EM process. See [8] for a more extensive discussion on these issues.

5 Concluding Remarks

EM has a potential to be useful in agile development projects. The proposed integration of AM and EM has been partly applied at two IS development projects at Riga Technical University, namely "Professional Orientation Information Base in Computer Science and Information Technology" and "Development of the Prototype for the Support of Inter-Institutional Flow of Knowledge". The EKD process helped consensus building between different stakeholders while the resulting models established project's "backbone" of knowledge. Initial experiences suggest that not only explicit artifacts of EM and AM influence project's agility, but also the growth of participants' tacit knowledge is to be taken into consideration. Deeper analysis of the tacit knowledge dimension in integrated AM and EM activities is a goal for future research aimed at development of methods for IS engineering of agile enterprises.

References

1. Beck, K. (2004). Extreme programming explained: Embrace change. Addison-Wesley.
2. Schwaber, K., & Beedle, M. (2002). Agile software development with SCRUM. Prentice Hall.
3. Ambler, S. (2002). Agile modeling: Effective practices for extreme programming and the unified process (1st ed.). John Whiley & Sons Inc.
4. Bubenko, J. A. Jr., Persson, A., & Stirna, J. (2001). User guide of the knowledge management approach using enterprise knowledge patterns, deliverable D3, IST project "Hypermedia and Pattern Based Knowledge Management for Smart Organisations", Royal Institute of Technology, http://www.dsv.su.se/~js/ekd_user_guide.html.
5. Persson A., & Stirna, J. (2001), Why Enterprise Modelling? – An Explorative Study Into Current Practice, in proc. of CAiSE'01, Springer, ISBN 3-540-42215-3
6. Jönsson, M. (2004). Agile modeling in Sweden – from practices to principles, MSc thesis, Stockholm University and Royal Institute of Technology, Stockholm, Sweden
7. Lagerquist, I., Lindmark, M., Stirna, J., & Nyfjord, J. (2006). Adoption of agile development in practice: A qualitative inquiry. In Industrial proc. of EuroSPI'06, Finland.
8. Stirna, J., Kirikova M., (2008) How to Support Agile Development Projects with Enterprise Modelling, Information Systems Engineering - from Data Analysis to Process Networks, Johannesson P. and Söderström E. (eds.), IGI Publishing, ISBN: 978-1-59904-567-2
9. Ambler, S. (2007). Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development, 11.27.2007, http://www.agilemodeling.com/essays/amdd.htm

i* Diagnoses: A Quality Process for Building i* Models

Antonio de Padua A. Oliveira [1, 2], Julio Cesar S. P. Leite [2],
Luiz Marcio Cysneiros[3], Carlos Jose P. Lucena[2]

[1] Universidade do Estado do Rio de Janeiro – UERJ
Rua São Francisco Xavier, 524 - 6 andar - Maracanã - Rio de Janeiro, Brazil

[2] Pontificia Universidade Catolica do Rio de Janeiro – PUC-Rio
Departamento de Informatica, Rua Marques de Sao Vicente 225 – Rio de Janeiro, Brazil

[3]York University – School of Information Technology
4700 Keele St. – Toronto, Canada

{padua, julio, lucena}@inf.puc-rio.br - cysneiro@yorku.ca

Abstract. Modeling with i* is not a trivial task. Our work describes i* Diagnoses Framework, a quality oriented process to analyze i* models. Our process is similar to some of the reading techniques of inspection methods and bears some similarity with the inquiry based requirement analysis approach. Our process focuses on defect prevention considering both the efficiency and effectiveness of Multi-Agent System development.

Keywords: early requirements, MAS, software development.

1 Introduction

There seems to be a consensus that dealing with intentionality at early stages of software projects is a reasonable idea. i* Framework [9] models have been receiving greater attention from several researchers [1], [2] as an infrastructure to deal with intentionality. Although i* has been cited and used in different research projects, most of their users agree that i* models are complex artifacts [7]. Although comprised of few elements, the semantics involved in using them can make i* models prone to errors [7].

The majority of the work has been focused on i* modeling and how to use this information on later stages of software production. Our goal is to focus on analyzing i* models proposing a quality assurance process to produce better i* models. Process quality focuses on defect prevention rather than looking for defects on test phase. We propose an analysis technique to enhance the quality of i* models.

We illustrate our proposal using "The Expert Committee System" (EC System) exemplar [3], a system to support the organization of a conference program.

2 The i* Canonical Structures

Figure 1 (right) shows the basic structure of an SRconstruct, which is formed by a goal (the goal's name is the SRconstruct's name) (as being the end) and at least by one task (as being the means to achieve the end). Therefore, all components (and subcomponents) needed by tasks (subtasks, resources, softgoals, and goals) should appear in the structure. Despite the fact that the goal is only one part of the

SRconstruct, we identify each SRconstruct by the name of the goal that it fulfills. That is because there is only one goal (as being the END) in each SRconstruct.

Figure 1 (left) shows that one actor (CHAIR) and another actor (REVIEWER) can have multiple dependencies in each SDsituation Situations of dependency that occur in the organizational environment and the central idea of SDsituations is: "each dependency link (goal, softgoal, task or resource) that involves actors is not isolated"; it is part of one well defined situation of collaboration called one "strategic dependency situation" or one SDsituation [6].

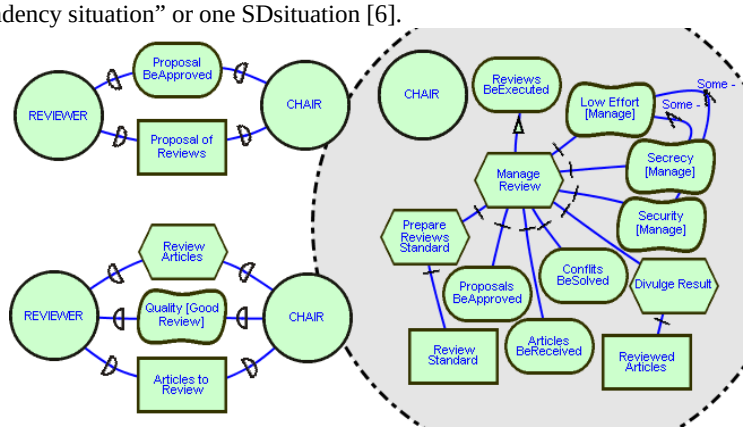


Figure 1 – Examples: left, two SDsituations, and right, one SRconstruct.

3 i* Diagnoses Strategy

The i* Diagnoses examine each canonical structure (SDsituations and SRconstructs) of a given model in order to bring questions that challenge the model consistency and completeness. The main idea is to focus on parts of an i* model and from these parts conduct an inquiry into the given construct.

Our process comprises 3 main sub-processes: IDENTIFY CONSTRUCTS, APPLY (INQUIRY) FRAMEWORK, and INTEGRATE QUESTIONS. The strategy is applied both to the Strategic Dependency Diagrams and to the Strategic Rationale Diagrams of i*.

The activity IDENTIFY CONSTRUCTS consists of breaking down the i* diagrams into constructs (SDsituations and SRconstructs). The activity APPLY (INQUIRY) FRAMEWORK consists of applying the inquiry framework to each construct. The activity INTEGRATE QUESTIONS consists of merging the questions to analyze them. The aim of each diagnose framework is to turn coupled SDsituations and SRconstructs inside out looking for problems, faults, deficiencies and potential improvements. "Diagnoses are important to deeply understand the problem before looking for the solution."

4 SDsituation & SRconstruct Diagnoses

Given the basic structures of SDsituations and SRconstructs, general questions are proposed to each element. In real cases these "hot-spots" or "place-holders" would be replaced in Templates 3.1 and 3.2 by the actual names used in the model.

Template 3.1 - **SDSITUATION**

SDSITUATION: "**SDsituation's name**"

I. INTER QUESTIONS
   1. *Who* else could collaborate with "***depender***" to have "***SDsituation goal's name***"? *How much can he collaborate?*
   2. *Why* does "***dependee***" collaborate with "***depender***" to have "***SDsituation goal's name***"?
   3. *What* SDsituations come before "***SDsituation's name***"?
   4. *What kind* of problems with previous SDsituations can be identified to have "***SDsituation goal's name***"?
   5. *What if* "***dependee***" cannot collaborate on the "***SDsituation's name***"?

II. INTRA QUESTIONS
   6. *What* are the problems inside "***SDsituation's name***"? What kinds of problems (accuracy, deficiencies, ambiguities, or omissions) are identified as having "***SDsituation goal's name***"?
   7. *What* details are needed by "***depender***"?
      a) Case: resource dependency - *What* are "***resource's name***" problems of availability? (Time, accuracy) *When? How? How much?*
      b) Case: goal dependency – *What* are "***goal's name***" problems to be achieved by "***dependee***"? (Time, ability) *When? How? How much?*
      c) Case: softgoal dependency – *What* are "***softgoal's name***" problems to be satisficed by "dependee"? (Capability) Is there "***softgoal's name***" at the end of "***SDsituation's name***"? *Why*? *Who* is demanding the softgoal?
      d) Case: task dependency - Has "***dependee***" received the directions of *how to* perform "***task's name***"? Can the "***dependee***" still perform it? (Time, ability)
   8. *What* dependency has the main duty of having "***SDsituation goal's name***"? Why?

Template 3.2 - **SRCONSTRUCT**

SRCONSTRUCT: "SRconstruct goal's name"

I. INTER QUESTIONS
   1. *Who* else has the "***endGoal's name***" achieved?
   2. *What* are the alternatives that the "***endGoal's name***" has achieved? *Why*?
   3. *What* are the elements of dependency of dependees?
   4. *What* kinds of problems (accuracy, deficiencies, ambiguities, or omissions) can be foreseen? How much? *What if* resources are unavailable? *Who* is to blame? *How to* avoid such problems?
   5. *What if* "***endGoal´s name***" is shared with another actor?
   6. What other construct depends on this goal? Why? How much?

II. INTRA QUESTIONS (for each meanTask)
   7. What are the problems with the task "***meanTask's name***"? *Why*?
   8. For the task "***meanTask's name***" *what* are the components needed to achieve "***endGoal's name***"?
      a) Case: resource – *What* are "***resource's name***" problems of availability? (Time, accuracy). *When? How?*
      b) Case: subGoal – *What* are "***subGoal's name***" problems to be achieved by "dependee/actor"? (Time, ability). *When? How?*
      c) Case: softgoal – *What* are "***softgoal's name***" problems to be satisficed by "***dependee/actor***"? (Capability) Is there "***softgoal's name***" at the end of

"***SRconstruct goal's name***"? *Why*? *What* are the contribution links to and from this ***"sofgoal´s name"***?

d) Case: subTask - Can "***dependee/actor***" perform "***task's name***"? (Time, ability)

9. Is there any softgoal details omitted, not fully operational or without operationalization? What kind? Why? How? How much?
10. Is there any resource missing? What kind? What if a resource is not available?

5 Conclusion

The first benefit of using i* canonic structures (SDsituations and SRconstructs) is managing complexity. Using SDsituations and SRconstructs, i* models can be divided into small pieces avoiding common misuses that appear in i* models [4] and also improving the stakeholders' understanding.

Our strategy provides a verification based analysis for i* models so as to assure better quality models overall. The verification analysis is performed on composing the constructs with well known general questions, the 5w2h framework [5] and with the ideas of Potts, Takahashi and Anton [8].

According to Moody [10], although software quality proposals have been concentrated at the end of the process, empirical works demonstrate that the majority of defects occur during the requirements phase.

We plan to continue the work in this direction as we will frame our diagnoses approach as a reading strategy for the inspection of i* models. By performing more analysis using the proposed i* diagnoses, we hope to improve the quality of the questions as they are today. We also foresee a possible automation, by generating the set of questions, given a set of i* models. Moreover, we plan to evaluate how this work may scale up to larger models.

References

1. Castro, J.; Kolp, M.; Mylopoulos, J. "Towards Requirements-Driven Information Systems Engineering: The Tropos Project." In: The 13th international conference on advanced information systems engineering, Oxford: Elsevier Science Ltd, v.27, n.6. p. 365-389 - 2002.
2. Cysneiros, L.M. and Yu, E.; Requirements Engineering for Large-Scale Multi-Agent Systems Book chapter in Software Engineering for Large-Scale Multi-Agent Systems – Research Issues and Practical Applications. A. Garcia, C. Lucena, F. Zambonelli, A. Omicini and J. Castro (eds.) LNCS 2603, Springer Verlag. 2003. (Revised and extended version of [SELMAS02]).
3. Deloach, S. et al.; Multiagent Systems Engineering. International. In: Journal of Software Engineering and Knowledge Engineering, 11(3): 231—258 - 2001.
4. Estrada, H; Martínez, A; Pastor, O; Mylopoulos, J.; An Experimental Evaluation of the i* Framework in a Model-based Software Generation Environment; E. Dubois, K. Pohl (Eds.); CAISE 2006, LNCS 4001, pp.513-527, 2006. Springer-Verlag, Berlin Heidelberg, ISSN: 0302-9743; ISBN: 3-540-34652-X, 978-3-540-34652-4.
5. Leite, J.C.S.P, Yu Y.; Liu Y.; Yu E.S.K., Mylopoulos, J.: "Quality-Based Software Reuse" CAiSE-05, LNCS 3520, pp.535-550, 2005, Springer-Verlag .
6. Oliveira, A. Padua A.; Cysneiros, L. M.; "Defining Strategic Dependency Situations in Requirements Elicitation" The IX Workshop on Requirements Engineering; Rio de Janeiro, Brazil - July/2006.
7. Pastor, Oscar; Estrada, Hugo; Martínez, Alicia; The Strengths and Weaknesses of the i* Framework: an experimental evaluation i*, its Applications, Variations and Extensions. Eric Yu et al. (eds.) MIT Press (accepted for publication in 2006)
8. Potts, Colin; Takahashi, Kenji; Antón, Annie I.; "Inquiry-Based Requirements Analysis", IEEE Software, Volume 11, Issue 2 (March 1994), Pages: 21 - 32
9. Yu, E. Modelling Strategic Relationships for Process Reengineering. PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada - 1995.
10. Moody, D.L. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, Data & Knowledge Engineering, 55 (2005) 243-276.

Aligning Goal Models and Business Models – extended abstract

Birger Andersson[1], Maria Bergholtz[1], Ananda Edirisuriya[1], Tharaka Ilayperuma[1], Prasad Jayaweera[2], Paul Johannesson[1], Jelena Zdravkovic[1]

[1]Department of Computer and Systems Sciences
Stockholm University and Royal Institute of Technology
{ba, maria, si-ana, si-tsi, pajo, jzc}@dsv.su.se

[2]Department of Computer Science
University of Ruhuna, Matara, Sri Lanka
prasad@ruh.ac.lk

1 Introduction and related work

In this paper we investigate the relation between the notions of goal models and the notions of business models. We will argue that aligning goal models and business models amounts to formulating goals in business model notions. We acknowledge that not all kinds of business goals are possible to formulate but we argue that a sufficient amount of them are to make this work worthwhile and the results useful. The results may be used, for instance, when aligning organizations with their IT resources.

For illustration purposes we will use the framework and terminology of the Business Motivation Model (BMM) [3] to capture goals and use the framework and terminology of e3value [5] for business modelling. We illustrate how the connection between goal models and business models can be exploited by proposing and outlining a method for model alignment. The method amounts to decomposing goals to the level of means and expressing the means using business modelling notions. The method approach is to use templates for means formulation to accomplish the alignment. The main benefits of the method lie in its simplicity and uniformity in goals formulations.

Business Models. There exist a number of approaches, languages, and ontologies for business models in the literature, e.g., [1], [4]. For the purpose of this paper we will make use of a comprehensive and well established business model ontology, the *e*^{3}*value* [5]. The basic concepts in *e*^{3}*value* are actor, market segment, value object, value port, value interface, value activity and value exchange.

Figure 1 is an *e*^{3}*value* model of a real world business case that is used as a running example. It models the various value exchanges between a provider of Massively Multiplayer Online Games (MMOG), its customers and a business associate, an Internet Service Provider (ISP). Actors are shown by rectangles, value activities by rounded rectangles, value ports by triangles, value interfaces by oblong rectangles enclosing directed value ports, and value exchanges as lines between value ports with the names of value objects as labels. In this business model there are two actors and a market segment involved – the Game Provider, the ISP and the Customer. The Game

Provider is responsible for producing the game content, selling, and distributing its software on CDs to the customers. In order to play the game, the customers need internet access, which they get from the ISP. They also need access to the game server, which they get from the Game Provider

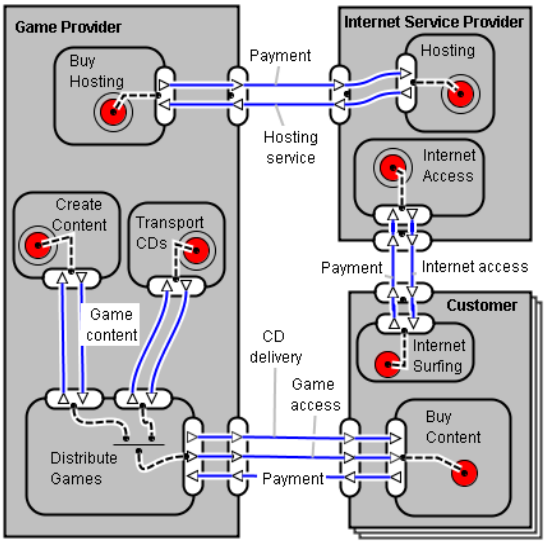


Fig. 1. *e*^{3}*value* model for the MMOG case

Goal Models. Goal models are used to capture and make explicit the goals of an enterprise. They direct the enterprise toward concrete actions, and as a consequence, the elicited actions are firmly based on a business motivation. A goal is defined as a desirable state the enterprise wants to reach. We use the BMM [3], as the technique focuses on the states an enterprise (i.e. the *principal actor*) wishes to achieve, as well as on the actions that will enable the achievement of those states. The technique relies on the use of three major concepts – Ends, Means, and Influencers. An End is something the enterprise seeks to accomplish, without any indication of how it will be achieved. A Means represents any capability or instrument that may be used to achieve Ends. An Influencer is anything that may impact the achievement of means (and thereby goals). In Figure 2, we illustrate the basic BMM elements and their relations using a small excerpt of a goal model for the MMOG case.

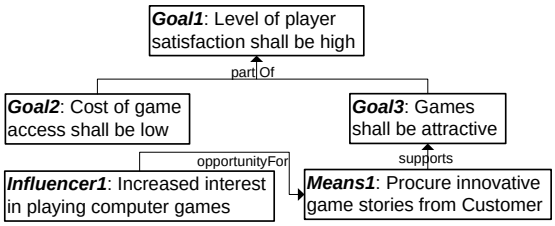


Fig. 2. Excerpt of a goal model for the MMOG case

2 Bridging Goal Models and Business models

A common problem in goal modelling is that goals are difficult to formulate, that is, the formulations of goals and means often become loose and highly abstract. In the following we propose that this can be amended by formulating them according to a

template structure. Each template has two parts, one compulsory and one optional, which is written within square brackets. The compulsory part contains the most important piece of information, while the optional part provides complementary information about the consequences of the compulsory part. A goal modeller may choose to fill in the optional part in order to provide complete information, but in many cases it is preferable to leave it out in order to make the goal model less complex. However, the business modeller has to complete the optional part before she is working towards the to-be business model. Below follows an example of use of one template. A wider set of templates and rules for applying them can be found in [2].

Example of a template: Value Object Procuring Means Templates
All mission statements that deal with value object procuring from suppliers could be captured with this template category.

1. procure $ValueObject_1$ from $Actor_1$ [use $ValueObject_1$ in $ValueActivity_1$ | offer $ValueObject_1$ to $Actor_2$ AND provide $ValueObject_1$ to $Actor_1$]

The compulsory part in this template is related to the procurement of a value object by the principal actor from another actor. The optional part describes the possible effects of the procurement of the value object. The value object procured may be used as an input to produce a certain value object or it may be offered directly to the principal actor's customers.

Method Overview. We will now discuss how business models should be aligned with goal models. For that purpose, we propose a method that takes as input a business model and a goal model and produces a new business model conforming to the goal model. In other words, a to-be business model is constructed using an as-is business model and a goal model as inputs. The main instrument used in the method is the means templates. Using this method the goal modeller first needs to construct the goal model expressed in terms of business model notions, which is accomplished by formulating the means according to the aforementioned means templates. The method can be summarized as follows:

1. The goal modeller constructs a goal model using the means templates
2. For each means the business modeller
   - complements the means by filling in the optional parts of its template when needed
   - modifies the business model based on the completed means template

Application of the Method. For each means in the goal model (step 1) select the means template and if needed complement the means with the optional part of the template, and (step 2) use the business model components (e.g. Value Objects, Value Exchanges, etc.) in the template to construct the to-be business model.

The following example shows the result of applying the method to a part of the business model of Figure 1.

Means 1: Procure Innovative Game Stories from Customer

Select template 1 (see above) and complement with the optional part.

procure *Innovative Game Stories* (value object) from *Customer* (actor) [use *Innovative Game Stories* (value object) in *Create Content* (value activity) AND provide *Payment* (value object) to *Customer* (actor)]

This means will lead to the addition of a new value exchange and a new interface for procuring Innovative Game Stories from the Customer. It will also add a new value exchange related to the Payment from Game Provider to Customer. Those exchanges will then be connected to the existing value activity Create Content that uses these Innovative Game Stories to produce Game Contents. See ②, in Figure 3.

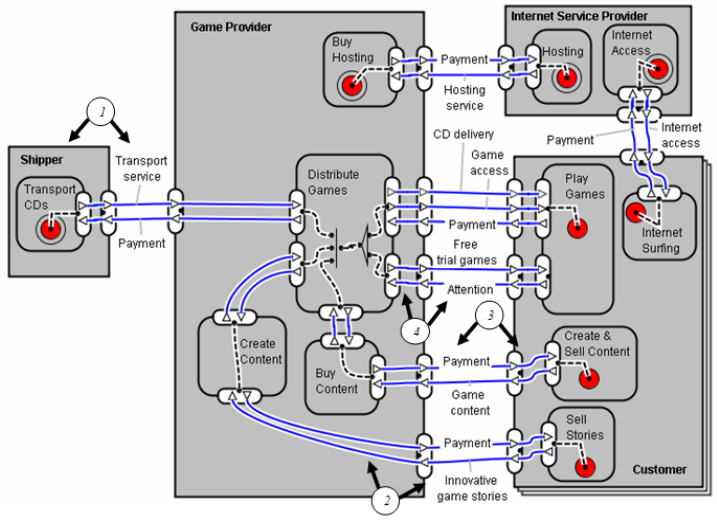


Fig. 3. Extended *e³value* model for MMOG case

3 Summary

This paper has argued that for an enterprise to be sustainable its operational processes should be aligned to its strategic goals. We have focused on a part of the complex issue of business and IT alignment by addressing the problems of aligning business models with goal models and a method for this was proposed. The method approach offers a number of benefits: clear and uniform goal model formulation, well founded business model design, and traceability between models.

References

1. Andersson B., et al.: Towards a Reference Ontology for Business Models, *25th International Conference on Conceptua Modeling(ER-2006)*, Arizona, USA
2. Andersson B., et al: Enterprise Sustainability through the Alignment of Goal Models and Business Models. Technical Report. Available at www.dsv.su.se/~ba/alignment08.pdf.
3. Business Motivation Model release 1.3., The Business Rules Group. 2007., Available 071205 at http://www.businessrulesgroup.org/second_paper/BRG-BMM.pdf
4. Dietz J.L.G.: Enterprise Ontology – theory and methodology, Springer-Verlag, Heidelberg, Berlin, New York (2005)
5. Gordijn J., Akkermans J.M. and Vliet J.C. van.: Business Modeling is not Process Modeling, *Conceptual Modeling for E-Business and the Web*, LNCS 1921, Springer-Verlag:40-51 (2000)

Using Multicriteria Decision-Making to Take into Account the Situation in System Engineering

Elena Kornyshova, Rébecca Deneckère, Camille Salinesi

CRI, University Paris 1 - Panthéon Sorbonne, 90, rue de Tolbiac,
75013 Paris, France
{elena.kornyshova,rebecca.deneckere,camille.salinesi}@univ-paris1.fr

Abstract. All Software Engineering (SE) processes include steps where several alternatives call for decisions. However, in many cases, the choice is intuitive and thereafter hazardous with unpredictable consequences. On the other side, the operational research domain has produced many methods that could be adequately used in these situations. Using these methods should facilitate the decision making activity by considering specific SE situations. However, no work has been done to understand how, when, or which of these methods could be used in SE. This paper describes how multicriteria methods could be applied to consider the situation in the SE.

Keywords: Multicriteria method, Decision making, Software engineering, Situation

1 Introduction

Information system (IS) conception, development, implementation, and every other process in Software engineering (SE) includes steps where several alternatives are considered and a decision must be made. Existing SE methodologies sometimes offer a way to guide decisions, for instance, in the requirements engineering [1], in the method engineering [2], or in other contexts. SE-related decisions result from the need to satisfy practical constraints such as quality, cost or time [3]. However, this field can be characterized by poor understanding and describing decision problems, a lack of transparency, of considering decision consequences and stakeholders' interests [3]. Therefore, we believe that an advanced decision aid is needed in the SE context.

On the other hand, the operational research area has developed numerous decision-making (DM) methods, for instance, multicriteria (MC) methods (a large overview of MC methods is presented in [4]). However, bibliographic researches show that few attempts have been conducted to systematically guide the selection of DM methods [5] and that none was developed to deal with in the IS engineering context.

In this paper, we study the application of MC methods in order to take into account specific SE situations. The paper is organized as follows. The next section provides an overview of MC methods application for considering situations in the SE. Related works and our research perspectives are discussed in the concluding section.

2 Multicriteria Method Application in the SE Context

The manner to consider the specific situation in SE using MC methods is threefold: (i) by structuring specific DM situation, (ii) by considering DM situation specificity, and (iii) by application of MC method adapted to this concrete situation.

2.1 DM Problem Definition

Many decisions are made in the field of IS. Despite their importance, these decisions are most often ill-formulated. They are characterized by poor understanding and describing decision problems, misunderstanding of decision consequences, and by a lack of transparency. To solve these problems, we investigate the main notions of DM and introduce two DM levels that help structuring the problem of DM in the SE.

B. Roy defines three basic concepts that play a fundamental role in analysing and structuring decisions [6]: alternatives (potential actions), criteria family, and decision problem. The decision *problem* [6] can be defined by the result expected from a DM. When the result is a subset of potential alternatives (most often one alternative) then it is a *choice problem.* When the result represents the potential alternatives' affectation to some predefined clusters, then it is a *classification problem.* When the result consists in potential alternatives ordered collection then it is a *ranking problem*. The concept of *alternative* designates the decision object. Any decision involves at least two alternatives that must be identified. A *criterion* can be any type of information that enables the alternatives evaluation and comparison. There are many different kinds of criteria: intrinsic characteristics of artefacts or processes, stakeholders' opinion, potential consequences and impacts of alternatives etc.

From a DM perspective, we propose considering two decision types: (i) the actual decision that aims at solving a SE problem and (ii) the decision on selection of a DM method that matches the situation in the former decision. These two types of decisions are respectively represented in Fig.1 within the levels 1 and 2. At level 1, an engineering decision leads to the choice, ranking, or classification of given alternatives with respect to various criteria defined in the situation. At level 2, a decision is made on different methods that enable to deal with the first level decisions. In this case, the MC methods are the alternatives; and the solution is selecting MC method that shall be used to make the actual level 1 decision.

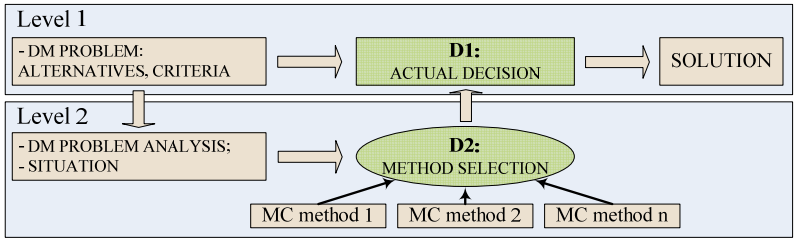


Fig. 1. Two levels of DM.

2.2 DM Situation Specification

The DM situation can be specified accordingly to the characteristics of DM problem (problem, alternatives, and criteria) and to the specific conditions of MC method application (usage). These characteristics and possible values are shown in figure 2.

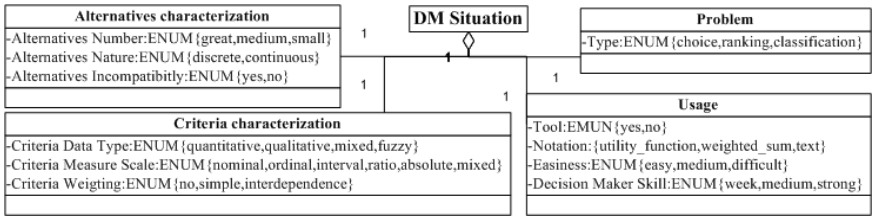


Fig. 2. DM situation specifying.

Several strategies may be applied to specify the characteristics values of alternatives, criteria, and problem (for instance, retaining the problem type, calculating alternatives number, retaining criteria measure scale, and so on). By instantiating these characteristics according to a given need, the engineer takes into account the specific situation. Additional information may also be required to specify the MC method usage in the concrete situation: the tool is required or not, the nature of the notation, the method easiness, and the level of engineer skills required for applying the method.

2.3 Multicriteria Method Selection

The selection of an appropriate MC method is carried out by its *interface*, which does not require focusing on the method content. The interface represents situations in which a given MC method can be used and corresponds to the characteristics described above.

The engineer specifies the values of these characteristics in a given situation. On this basis, a MC method could be chosen by different strategies. In this paper, we foresee the following possibilities: by MC search or by weighting.

First, a MC method may be selected by *MC search*. This means that the engineer applies a request to MC methods with identified values for obtaining one or several MC methods corresponding to the situation at hand. If it drives to the selection of several MC methods, it is possible to choose one of them by *weighting*. Using this approach, weights must be given to the characteristics. These weights indicate their relative importance in the situation at hand. Then, "0" or "1" are given to candidate MC methods according to each characteristic (in function of their correspondence to the situation). The method having the highest weighted sum of values is then chosen.

3 Related Works and Concluding Remarks

DM is a crucial problem. A poor choice may drive to a loss of time, money, and poor alignment to the situation. Our purpose is to spread MC methods in the SE. These methods would allow considering specific situation, better involving stakeholders, and increasing their confidence in the final decisions in SE.

In SE, the issue of DM was already explored with respect to requirements engineering [1,7], to method engineering [2,8], and more generally, to systems engineering [3]. Ruhe emphasized the importance of DM in SE along the whole life cycle [3]. Several examples of MC methods application can also be mentioned: AHP for prioritizing requirements [7]. Saeki uses weighting method to deal with software metrics [2]. The application of two MC methods (outranking and weighting) is illustrated in the field of method engineering [8]. The examples of selecting an appropriated MC method for business process prioritization are presented in [9,10]. Our proposal differentiates by focusing on MC decision aiding and MC methods selection corresponding to the situation.

A few proposals have been made before to help selecting an appropriate MC method. [5] presents a state of the art of existing approaches on the MC methods selection. In the SE field, [9,10] suggest constructing an analysis grid used for selecting a MC method according to the specificity of a given situation.

In the near future, our research perspectives involve: (i) improving the DM methods signatures to better select the MC methods; (ii) developing a tool supporting our approach; (iii) defining the MC methods as fragments for their integrating into existing SE methodologies; and (iv) evaluate our proposal by extensive case studies.

References

1. Karlsson, J., Ryan, K.: A Cost–Value Approach for Prioritizing Requirements, IEEE Software (1997)
2. Saeki, M.: Embedding Metrics into Information Systems Development Methods: An Application of Method Engineering Technique. CAISE'03, Austria (2003)
3. Ruhe, G.: Software Engineering Decision Support – Methodology and Applications. In: Innovations in Decision Support Systems, 3 (2003)
4. Multiple Criteria Decision Analysis – State of the Art Survey, Editors: J. Figueira, S. Greco, M. Ehrgott, Springer (2005)
5. Kornyshova, E., Salinesi, C.: Selecting MCDM Techniques: State of the Art, International Journal of Information Technology and Intelligent Computing (IT&IC), 2 (2008)
6. Roy, B.: Paradigms and challenges, Book chapter, In Multiple Criteria Decision Analysis - State of the Art Survey, Springer. editor(s) J. Figueira, S. Greco, M. Ehrgott, (2005)
7. Maiden, N.A.M., Pavan, P., Gizikis, A., Clause, O., Kim, H., Zhu X: Integrating Decision-Making Techniques into Requirements Engineering, REFSQ'02, Germany (2002)
8. Kornyshova, E., Deneckère, R., Salinesi, C.: Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach, ME'07, Switzerland (2007)
9. Kornyshova, E., Salinesi, C.: Business Process Priorisation with Multicriteria Methods: Case of Business Process Reengineering, ICEIS'07, Funchal, Portugal (2007)
10. Salinesi, C., Kornyshova, E.: Choosing a Prioritization Method – Case of IS Security Improvement. In Forum Proceedings of CAISE'06, Luxembourg (2006)

Towards an Ontology-enabled Approach for Modeling the Process of Conformity Checking in Construction

Anastasiya Yurchyshyna[1, 2] Catherine Faron-Zucker[1], Nhan Le Thanh[1], Alain Zarli[2]

[1] I3S, Université de Nice Sophia-Antipolis, CNRS, 930 route des Colles, BP 145, 06903 Sophia Antipolis, France, {Catherine.Faron-Zucker, Nhan.Le-Thanh}@unice.fr
[2] CSTB, 290 route des Lucioles, BP 209, 06904 Sophia Antipolis, France, {anastasiya.yurchyshyna, alain.zarli}@cstb.fr

Abstract. This paper presents an ontological method aimed at semi-automatic checking the conformity of a construction project represented by RDF graph against a set of construction norms formalized as SPARQL queries. The reasoning is modeled by the matching of RDF representations of construction projects to SPARQL conformity queries. We integrate meta-knowledge relative to the checking process by annotating the conformity queries themselves and organize them according to their annotations. The queries annotations also help to guide the information/knowledge extraction and reasoning process and explain the results of the validation process, especially in case of failure.

Keywords: Conformity checking, knowledge extraction in construction, organization of the base of conformity queries, Semantic Web in Construction.

1 Introduction

The execution of construction products is nowadays characterised by complex rules and regulations. However, their current representations are still mostly paper-based (e.g. texts with diagrams, tables) and require a human interpretation [7].

Construction projects (e.g. public buildings) are commonly represented by the Industry Foundation Classes (IFC) model, an object oriented data model for Building Information Modelling. There is a standard XML representation for the IFC model (ifcXML[1]), which is, however, insufficient to describe the complexity of the building information flow: the IFC model is semantically richer than any XML language.

Our research aims at the development of a conformity-checking model based on semi-formal representations of technical norms: we study how to represent and organise them for the specific task of effective conformity checking. Our checking model is based on the matching of norm representations with those of construction projects. Its efficiency is explained by the *ontological representation* of regulation

[1] http://www.iai-international.org/IFCXML/

knowledge and the conformity-oriented *annotation of norms* with meta-knowledge improving the checking process and the explaination of its results.

2 Knowledge Representation Oriented Conformity Checking

The first phase of our knowledge acquisition method aims at *acquiring formal representations of technical construction norms* relative to the accessibility of disabled persons. We use the CD REEF, the electronic encyclopaedia of construction texts and regulations, to extract a base of accessibility constraints, which we formalise as SPARQL queries in terms of the IFC model. This is a manual process (the knowledge extraction from texts is out of the scope of our research) conducted in collaboration with construction experts (mainly from CSTB) who help to explicit the domain knowledge. As a result, we are provided with a base of SPARQL queries expressing *non conformity* constraints: e.g. "*The minimum width of a door is 90 cm*" is formalized by:

```
select ?door display xml where
{ ?door rdf:type ifc:IfcDoor
OPTIONAL { ?door ifc:overallWidth ?width
FILTER ( xsd:integer(?width) >= 90)}
FILTER (! bound( ?width) )}
```

The second phase aims at the *semi-automatic acquisition of an ontology oriented conformity checking.* This conformity-checking ontology is developed on the basis of the concepts occurring in the acquired conformity queries. *Primitive* IFC concepts are extracted from the ifcXML schema - solely those occurring in the conformity queries; they are organized into an OWL Lite ontology based on the schema structure. The conformity queries also make use of some non-IFC concepts. To integrate them in the ontology, the intervention of a domain expert is necessary whose task is to define these concepts with primitive IFC concepts. These definitions are represented by RDF graphs (e.g. GroundFloor is a subclass of IfcBuildingStorey defined as an IfcBuildingStorey situated on the level of entering into a building: the value of property pset_BuildingStoreyCommon_EntranceLevel is TRUE).

The third phase of our method consists in *the annotation of the conformity queries themselves for effective checking.* We associate them supplementary information, which is helpful in the conformity checking process: e.g. information on the regulation corpus from which queries are extracted. We automatically extract RDF annotations of conformity queries from the CD REEF, which contains information relative to regulations (in addition to the regulation itself): (i) characteristics of the regulation: type of regulation text (e.g. Construction Code), level of application (e.g. national); (ii) application domain (e.g. accessibility); (iii) destination of a building (e.g. public administration building). The acquired RDF annotations are later manually enriched by *domain* knowledge: (i) subject (e.g. entrance door); (ii) construction common knowledge (*obvious* for domain experts: e.g. a hotel is a public building, not a private house), etc.

The last phase is dedicated to the *acquisition of a construction project representation oriented conformity checking*. Such representations are developed on

the basis of the initial IFC representation and guided by the acquired conformity-checking ontology. First, we develop an XSLT stylesheet that filters the ifcXML description of a construction project, transforms only the data relative to the conformity checking ontology and finally builds an RDF graph representing the project. This RDF representation may be further enriched with some non-IFC concepts defined in the conformity-checking ontology (in second phase of our method) in case their definitions appear as subgraph as the RDF graph representing the project.

4 Conformity Checking Model

We adopt an ontological approach and the semantic web technologies [2] to develop our reasoning model [7]. It is based on graph-based formalisms for knowledge representation, which have declarative semantics, are logically founded, allow the structured representation of knowledge and describe it at the different levels (e.g. ontological and asserted knowledge). The basic reasoning operation for a query-answer system is graph projection, formally defined as a labelled homomorphism between graphs [3]. The *reasoning thus consists in graph homomorphisms* [1] [5] and modelling of the checking process is close to the process of validation of knowledge bases [6]. The elementary reasoning mechanism of our model is the matching of a construction project representation with representations of conformity queries. We check the *negative* constraint (e.g. "*the width of the door is less than 90cm*"): if such matching is found for some elements, these elements cause the *non-conformity* of the project.

Conformity queries are automatically classified and organized into a query base by parsing their RDF annotations. The classification is done according to (i) external information characterizing the query (e.g. regulation text); (ii) specialization-generalization relations, which could be found in the graph patterns of queries.

By organizing the queries, we define the optimal scheduling of matching procedures as *a set of explicit expert rules*. The expert reasoning is represented by the *query scheduling*: (i) according to priorities holding between *classes* of queries (e.g. queries extracted from *acts* are prior to *circular* ones); (ii) according to knowledge specification: inside the same query class, queries representing more specialised knowledge are treated in priority (e.g. an *entrance door* query is prior to a *door* query, because if a construction project is non conform to the first one, it will be automatically non conform to the second one); (iii) according to query annotations: priority is given to the queries with most specific annotations.

The *results of the checking process* (validation/non-validation, explanation of non-validation, no answer) are *analysed* to generate *a structured conformity report* grouping conformity queries by classes. It is automatically generated on the basis of annotations of classified queries. The conformity report lists queries that have failed (i) because of non-matching; ii) queries which graph pattern is more general in comparison to the ones previously that failed, (iii) queries which annotation representing the condition of its application is more general in comparison to the annotation of another failing query. Another possible reason of failure of the project

validation is that the representation of the construction project does not contain sufficient information for matching. In case of such incomplete representations, it is useful to precise the lacking elements (the sub patterns of the query which can not be matched), so that a user could know the reason of non-verifiability and/or complete the representation of the project.

5 Conclusion and Perspectives

We have presented the ontology-enabled model for the conformity checking process of a construction project against conformity norms, based on matching of an RDF representation of a project to a SPARQL conformity query. Conformity queries are annotated and organized to improve the checking process and help in the interpretation of checking results in terms of conformity in construction.

For validation of our conformity-checking approach, we develop the C3R[2] system, which relies on the CORESE [4] semantic engine that answers SPARQL queries asked against an RDF/OWL Lite knowledge base.

Ongoing works focus on the incremental development of the C3R prototype and its evaluation by domain experts.

References

1. Baget J-F. RDF Entailment as a Graph Homomorphism, in Proc. of the 4th conference on international semantic web conference (ISWC'2005), Galway (EI), LNCS 3729, Springer Verlag, pp 82-96, 2005
2. Berners-Lee T. Reflections on Web Architecture. Conceptual Graphs and the Semantic Web, 2001, available at http://www.w3.org/DesignIssues/CG.html
3. Chein M., Mugnier M-L. Conceptual Graphs : Fundamental Notions, Revue d'Intelligence Artificielle, volume 6-4, pages 365-406, 1992
4. Corby O., Dieng R., Faron-Zucker C. Querying the Semantic Web with Corese Search Engine. Proc. Prestigious Applications of Intelligent Systems PAIS, ECAI, Valencia (2004)
5. Croitoru, M., Compatangelo, E.: A combinatorial approach to conceptual graph projection checking. In: Proc. of the 24th Int'l Conf. of the British Computer Society's Specialist Group on Art'l Intell., AI'2004, Springer-Verlag (2004)
6. Dibie-Barthélemy. J., Haemmerlé O. et Salvat E. Validation de graphes conceptuels. In Actes des 4èmes journées Extraction et Gestion des Connaissances, EGC'2004, Clermont-Ferrand, 2004, RNTI-E2, Cépaduès, pp. 135-146
7. Faron-Zucker C., Yurchyshyna A., Le Thanh N. & Lima C. Une approche ontologique pour automatiser le contrôle de conformité dans le domaine du bâtiment. In Actes des 8èmes journées Extraction et Gestion des Connaissances, EGC'2008, RNTI-E11, Cépaduès, p. 115-120. Sophia Antipolis, France (2008).

[2] Conformity Checking in Construction with the help of Reasoning

Probabilistic Metamodel Merging

Robert Lagerström[1], Moustafa Chenine[1], Pontus Johnson[1], and Ulrik Franke[1]

[1] The Royal Institute of Technology, Industrial information and control systems, Osquldas v.12, 10044 Stockholm, Sweden
{robertl, moustafac, pj101, ulrikf}@ics.kth.se

Abstract. This paper proposes the use Bayesian networks for the automatic merging of metamodels. The proposed Bayesian networks calculate the probability that a merge of two metamodel elements is suitable, thus suggesting what to merge.

Keywords: Metamodel merging, Bayesian networks.

1 Introduction

In recent years, both researchers and practitioners have discovered the emerging possibilities of using metamodels and ontologies. As a consequence, a large quantity of metamodels and ontologies now exists within all sorts of different applications. The distributed environment of metamodel and ontology development has led to large overlaps within and between metamodels. Since these diverse metamodels often describe similar aspects of systems, developers and users would gain large benefits if metamodels could easily be merged and aligned with each other. Therefore, integration and merging of metamodels and ontologies has received an increasing interest lately [1][2].

This paper proposes an approach to metamodel merging where a probabilistic inference engine is employed to evaluate candidate metamodel concepts suitable for merging. The proposed approach uses Bayesian networks to assess the probability that a merge of two elements is suitable. The Bayesian network evaluates a merge based on various syntactic and semantic characteristics of the candidate concepts, such as the similarity of names and associations to other concepts.

The concepts described in this paper all pertain to a certain type of metamodels called abstract models. Abstract models have previously been proposed as a notation for describing and analyzing enterprise systems [3]. The models represent the architectures of these systems as well as other expert knowledge and empirical observations that can be instantiated and used for analysis. An abstract model contains classes and class associations, augmented with attributes and attribute associations. A UML description of abstract models can be seen in Fig. 1.

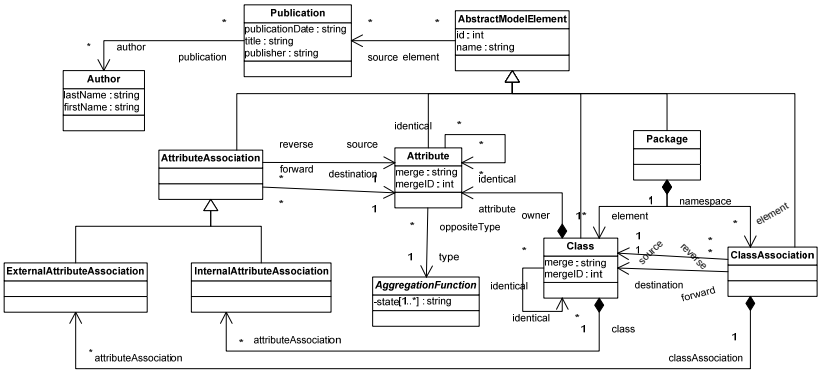


Fig. 1. UML description of abstract models.

2 Merging Metamodels using Bayesian Networks

A Bayesian network, $B=(G, P)$, is a representation of a joint probability distribution, where $G=(V, E)$ is a directed acyclic graph consisting of vertices, V, and edges, E. The vertices denote a domain of random variables $X_1,\ldots, X_n$, also called chance nodes. Each chance node, X_i, may take on a value x_i from the finite domain $Val(X_i)$. The edges denote causal dependencies between the nodes, i.e. how the nodes relate to each other. The second component, P, of the network B, describes a conditional probability distribution for each chance node, $P(X_i)$, given its parents $Pa(X_i)$ in G. More comprehensive treatment on Bayesian networks can be found in e.g. Jensen [4].

When merging abstract models, there are two concerns. Firstly, do any of the classes in the source models represent the same concepts? Secondly, when two classes have been merged, do any of the attributes in the merged class represent the same concepts? If these two concerns are correctly handled, then all associations separately holding in the source models will also be correctly transferred into the target model. Therefore, two Bayesian networks were developed; one describing class merges and one describing attribute merges, c.f. Fig. 2.

The nodes in the class merge network have the following scales: Class Merge = {Yes, No}, Class Similarity Association = {Yes, No}, Class Names = {Identical, Similar, Dissimilar}, Class References = {SamePublication, SameAuthor, DifferentAuthors}, Class Attributes = {All, Some, None}, and Class Associations = {All, Some, None}.

The nodes in the attribute merge network have the following scales: Attribute Merge = {Yes, No}, Attribute Similarity Association = {Yes, No}, Attribute Names = {Identical, Similar, Dissimilar}, Attribute References = {SamePublication, SameAuthor, DifferentAuthors}, and Attribute Associations = {All, Some, None}.

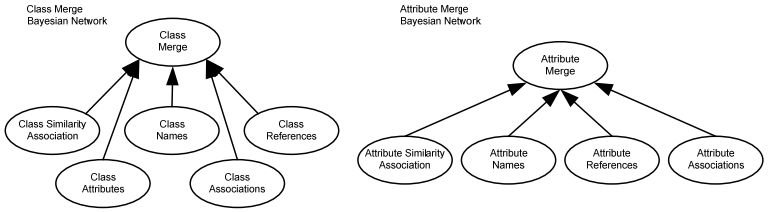


Fig. 2. Bayesian networks representing class and attribute merge.

To illustrate the application of the Class Merge network c.f. the Bayesian network screenshot from GeNIe [5] in Fig. 3.

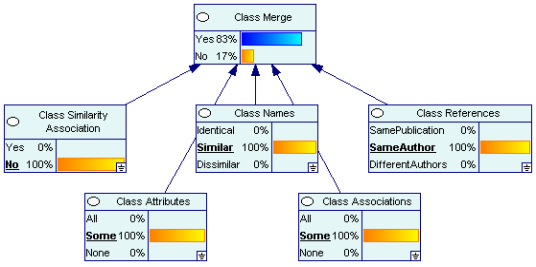


Fig. 3. The Bayesian network for class merges with example values.

Assume that a package, i.e. a set of abstract models, contains the abstract models presented in Fig. 4.

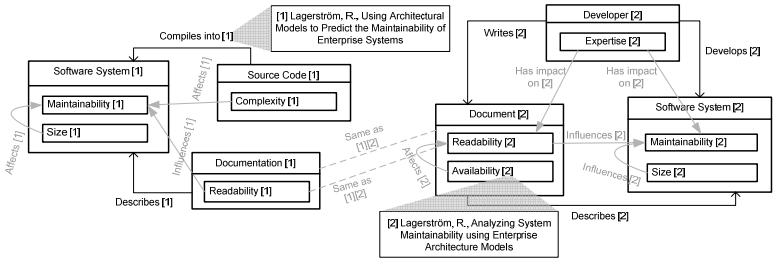


Fig. 4. Abstract models to be tested for possible merges.

The method starts by comparing the pairs of classes from the models presented in Fig. 4. Then, the classes with the highest probability are merged. In this example, the class pair *Documentation* and *Document* received the probability P = 100 %, and are therefore merged.

The next step is to compare all pairs of attributes in the merged class *Documentation*. Then, the attributes with the highest probability are merged. In this example, the pair *Readability* and *Readability* received the probability P = 100 %, and are therefore merged.

It all starts over from the beginning by comparing all pairs of classes and iterates until no classes receive probabilities over a predefined merging threshold. The resulting model in this example is presented in Fig. 5.

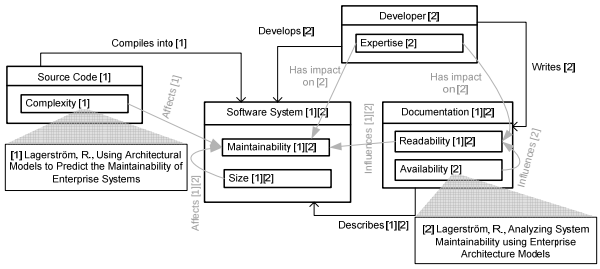


Fig. 5. The resulting abstract model after employing the proposed Bayesian networks.

4 Conclusions

This paper addresses the issue of metamodel merging, using the probabilistic framework of Bayesian networks. It was shown that Bayesian networks can be used to guide the merging of metamodels, by considering some key features of the classes and attributes at hand: basically their names, references, and associations. With this information it is possible to discern the probability that the concepts are sufficiently similar to be merged.

References

1. Noy, N., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. Internationl Journal of Human-Computer Studies, Vol. 59, No 6, p. 983-1024 (2003)
2. Doan, A., Madhaven, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In the proceedings of the 11th International Conference on World Wide Web (2002)
3. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A Tool for Enterprise Architecture Analysis, In Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC), Vol. 11, Annapolis USA (2007)
4. Jensen, F.: Bayesian Networks and Decision Graphs. Springer-Verlag (2001)
5. GeNIe: Decision System Laboratories, About GeNIe and SMILE, University of Pittsburgh, http://genie.sis.pitt.edu/about.html (2007)

An Hybrid Design Solution For Spacecraft Simulators

Vítor Rodrigues[1,2], João Correia Lopes[3,4], Ana Moreira[5]

[1] ESOC/ESA, D-64293 Darmstadt, Germany
victor.rodrigues@esa.int
[2] Oristeba — Space Services
http://www.oristeba.com
[3] INESC Porto, 4200-465 Porto, Portugal
[4] Faculdade de Engenharia da Universidade do Porto, 4200-465 Porto, Portugal
jlopes@fe.up.pt
[5] Dept. Informática, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
anm@di.fct.unl.pt

Abstract. The European Space Agency (ESA) has created the Simulation Model Portability 2 (*SMP2*) standard with the purpose to provide a design solution for the project of Spacecraft Simulators. One element of the *SMP2* standard is the metamodel Simulation Model Definition Language (*SMDL*). The design artefacts of a Spacecraft Simulator consist in descriptions of the business logic shared by a set of *SMP2* models. This paper reports results from a study that considers the hypothesis to complement the *model-driven* design approach of the *SMP2* standard with *test-driven* design techniques. The high-level abstractions of Spacecraft Simulators are used to carry out *Model-Driven Development* processes, while reusable pieces of software that can to be used by many *SMP2* models are designed and developed following *Test-Driven-Development*. The tool capable to establish the dependencies between the source code produced by the two methodologies and mission specific source code is the *GNU Build System*.

Keywords: simulation model portability, model-driven development, test-driven development, GNU build system, hybrid design techniques

1 Introduction

The design of Spacecraft Simulators is based upon a component model specified by the the *SMDL* modeling language which focuses primarily on interface reuse [1]. *SMP2* models are described in *SMP2* design artefacts that are transformed into C++ *skeletons* into which the behaviour implementations must be *added*. Although model-driven design is ideal for developing software in multiple computing platforms, multiple implementations of the same interface is left out of its scope [2]. Our objective is to *deconstruct* the purely top-down strategy of model-driven development using the *SMP2* standard onto a bottom-up development process of a *SMP2 Framework* [3]. On the other hand, reusable behaviour

implementations of a Spacecraft Simulator are developed on top of a distinct software framework called *Infrastructure Framework* developed with test-driven techniques. The hybrid design solution aggregates the principles of both software development approaches.

2 Background

Design prototyping in object orientation is the activity of using "objects" that represent abstract entities to define design models. From this premise follows that the planning of the coding activity is done by *describing* the objects and the business logic they share. Although the design models are written before any programming language specification, we can establish a direct correspondence between the symbology of a modeling language and the symbology of a programming language, thus making the design models a cross-platform specification from which model-driven development departures.

A different approach to software design is the specification of code functionalities through test code. Using only test code it is possible to design a piece of software before entering the stage of source code development. Similarly to design models, test code is a design artefact, but it does not follow a pre-defined semantic scheme as per design models. Nonetheless, test-driven development is a restrained process, carried out in closed loop through source code *refactoring* which makes it appropriated for developing software systems that are continually evolving.

This *GNU Build System* provides modelling languages to specify dependencies between *source packages* [4]. *Cross-platform* processes of creation of makefiles are integrated within the development of *SMP2* components, making possible the attempt of several configurations involving the *SMP2* models and the reusable libraries, which derive from disparate development lines.

3 Specification

Our premise is that there are parts of the spacecraft functionality that do not depend on the business logic configuration, because they are generic, context independent and, therefore, reusable. Therefore, the implementation of a Spacecraft Simulator, whether it is accomplished from scratch or as an update to an existing solution, may be decoupled from the design models.

An hybrid system would be possible to reconfigure by simply adjusting the variable parts that in it coexist. The articulation of both frameworks, that is, the integration of the reusable software *inside* the *SMP2* models, is accomplished by the *GNU Build System* (see Figure 1). The creation of a *SMP2* model is done indirectly. Template methods of the *SMP2 Framework* become hook methods on the specialized classes and the mechanism of object inheritance is used to plug-in software components into the framework's *hot spots* [5].

To build a functional *SMP2* software component it is necessary to link the code *skeletons* and the infrastructure libraries with "glue" code designated by

mission specific, which must reach the reusable behaviour provided by the infrastructure libraries and make the necessary adaptations to, in its own behalf, provide to the other *SMP2* components the behaviour which is specified in the *SMP2* logic structure. This activity constitutes the tuning of the system and the *GNU Build System* guarantees that the system is kept in a consistent state [6].

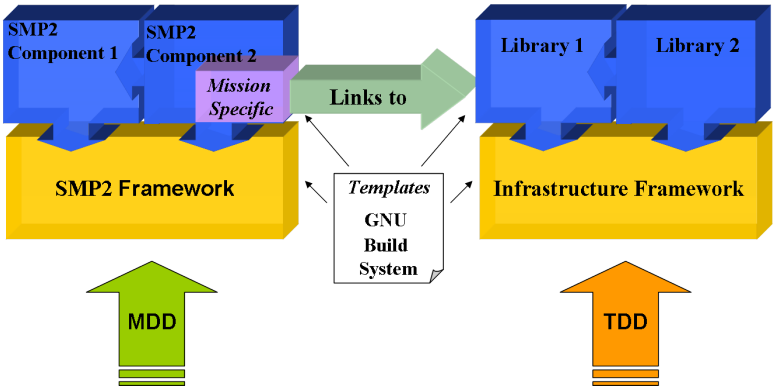


Fig. 1. The Development Lines of MDD and TDD

Two phases of decoupling are foreseen: the first decouples interface descriptions from code *skeletons*, and the second decouples code *skeletons* from behaviour implementations. The transformation between design models and code *skeletons* is automated by the model-driven development environment, but the linking to the behaviour implementations is not offered by the *SMP2* standard.

The access to the infrastructure functionalities can be automated by adopting the 'Adapter' and 'Abstract Factory' *GOF* [7] design patterns, reducing the amount of manual writing and software analysis: if the translation between the public *SMP2* interfaces and the private interfaces of the infrastructure libraries can be specified using the *SMP2* support for metadata, then this translation is foreseen as yet another transformation inside the *MDD* environment.

The hybrid system is developed by an iterative process. In each iteration it is possible to refine the *SMP2* interface signatures or the behaviour provided by the infrastructure libraries. The granularity of the *SMP2* models is decided upon metrics analysis and the granularity of the *Infrastructure Framework* is determined by the number of different contexts where a given library can be reused. If these two operations converge and if relation between the *SMP2* models and the combination of infrastructure libraries is of the type *adapter:adaptee*, then the additional coding of mission specific code will increasingly tend to zero [6].

With a pure model-driven design approach the initial prototype *becomes* the simulator after completing the coding task. This enforces the validation of the design in the earlier stages of the project life cycle [8]. On the contrary, the risk

of changing the business logic of an hybrid system during development is reduced since the core tasks of coding are done on top of the *Infrastructure Framework*. In such a decoupled system, the *SMP2* code *skeletons* can be regarded as *test* code, but there has to be no commitment to an initial high-level design.

4 Conclusion

A single design approach is hardly a one-size-fits-all solution. An hybrid system is more pluralistic because it provides the software engineers with a large set of "building blocks", which are designed to be useful in different application contexts and used to build software systems without a fully pre-fabricated structure. In this line of thought, the design of Spacecraft Simulators supported in the two distinct development methodologies of *MDD* and *TDD* widens the covering of the software requirements and produces a more complete project specification. The advantage of an hybrid solution is the opportunity to circumscribe the technological *push* of the *SMP2* standard and work exclusively on the design models and evaluate the impact that the modelling breakdown imposes. The *GNU Build System* is the enabling technology for an hybrid design solution, bringing flexibility to the integration of source code derived from the *MDD* and *TDD* development lines.

References

1. ESA: Smp 2.0 Handbook. Technical report, EGOS-SIM-GEN-TN-0099 (October 2004) Issue 1 Revision 0.
2. Brown, A.: An introduction to Model Driven Architecture, Part I (2004) Online resource: `http://www-128.ibm.com/developerworks/rational/library/3100.html`, last accessed on 10-March-2007.
3. Johnson, R.E.: Components, Frameworks, Patterns. In: ACM SIGSOFT Symposium on Software Reusability. (1997) 10–17
4. Vaughan, G.V., Elliston, B., Tromey, T., Taylor, I.L.: GNU Autoconf, Automake, and Libtool. Sams; 1st edition (2000)
5. Pree, W.: Meta Patterns — A Means for Capturing the Essentials of Reusable Object-Oriented Design. Lecture Notes in Computer Science **821** (1994) 150+
6. Rodrigues, V.: On the Specification of Spacecraft Simulators using Object-Oriented Methodologies. Master's thesis, University of Oporto, Departament of Electrical and Computer Engineering (2007)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (2005)
8. Ambler, S.W.: The Object Primer. Cambridge University Press (2004)

Measuring the Functional Size of Conceptual Models in an MDA Environment[1]

Beatriz Marín[1], Nelly Condori-Fernández[1], Oscar Pastor[1] and Alain Abran[2]

[1] Department of Information Systems and Computation,
Technical University of Valencia,
Camino de Vera s/n, 46022 Valencia, Spain
{bmarin, nelly, opastor}@dsic.upv.es

[2] Department of Software Engineering & Information Technology,
École de technologie supérieure - Université du Québec,
1100 Notre-Dame Ouest, Montréal (Québec), Canada H3C 1K3
abran.alain@etsmtl.ca

Abstract. The measurement of the functional size of applications generated in MDA environments is a challenge for the software development industry. This paper presents the OO-Method COSMIC Function Points (OOmCFP), a measurement procedure that has been designed to measure the functional size of object-oriented applications generated from their conceptual models by means of model transformations.

Keywords: Conceptual modeling, Object orientation, Functional size measurement, COSMIC, MDA.

1 Introduction

The MDA approach separates application and business logic from the platform technology, allowing code generation by means of model transformations. This is the case of the OO-Method approach [7], which is an object-oriented method that provides the semantic formalization needed to define complete and unambiguous conceptual models, allowing the automatic generation of software products using a MDA-based technology.

The adoption of MDA-based technology has presented new challenges, such as measuring the size of the generated products. The COSMIC measurement method [5] can be used to perform this task. Currently, there are some approaches that apply COSMIC to estimate the functional size of future software applications from high-level specifications [2] [3]. Since the functionality to be measured using these proposals is not detailed enough to generate the final application, the definition of an FSM procedure that allows the measure of the correct size of applications is needed.

[1] This work has been developed with the support of MEC under the project SESAMO TIN2007-62894 and co financed by FEDER.

In practical settings, it is very important to know the correct size of the conceptual models that are built with the MDA-based approaches, since the functional size of the models must be known in order to estimate the cost of the software product that is generated automatically. This paper presents the OOmCFP proposal, which is a procedure based on COSMIC that allows the measurement of the functional size of the OO-Method conceptual models from which the applications will be generated.

The rest of the paper is organized as follows: section 2 presents the design of the OOmCFP, and section 3 presents some conclusions and suggestions for further work.

2 Design of a Measurement Procedure

According to the process model for software measurement proposed by Jacquet and Abran [6], the design phase of a measurement procedure is related to the definition of the concept to be measured and the rules to measure this concept. This relevant phase is divided into four sub-steps: the definition of the objectives, the characterization of the concept to be measured, the selection of the metamodel, and the definition of the numerical assignment rules.

In the sub-step *definition of the objectives*, the objective of OOmCFP is to design a procedure in accordance with the COSMIC functional size measurement method for measuring the functional size of software applications that are generated using an MDA approach from their conceptual models which are built with OO-Method.

In the sub-step *characterization of the concept to be measured*, we define the entity and the attribute to be measured. The input artefact used to measure the functional size of the OO-Method applications is the Conceptual Model. This model is comprised of four models (Object, Dynamic, Functional, and Presentation) that allow the generation of a fully working software application. The conceptual model of OO-Method is described in detail in [7]. Therefore, the *entity* to be measured by OOmCFP will be an OO-Method conceptual model, and the *attribute* to be measured will be the functional size, which is defined by the ISO/IEC 14143-1 standard as the size of software derived by quantifying the functional user requirements [4].

In the sub-step *selection of the metamodel*, we have selected the COSMIC standard because, in contrast to other FSM standards like IFPG FPA, NESMA FPA or MARK II FPA, it allows the functional size measurement of multi-layer applications (like OO-Method) from different viewpoints. Figure 1 shows the COSMIC metamodel according to the COSMIC measurement manual version 3.0 [1], which illustrates the information that should be represented by the software artefact to be measured.

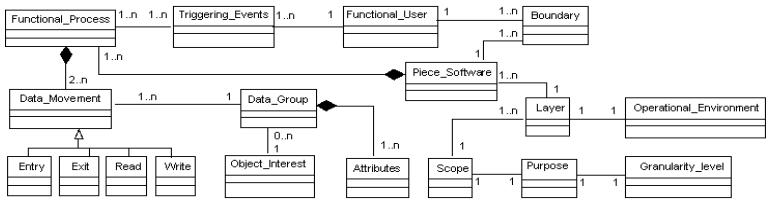


Fig. 1. Metamodel of COSMIC.

The *purpose* of the measurement in OOmCFP is to measure the functional size of the OO-Method conceptual models to estimate the cost of the applications generated by the OlivaNova Suite.

The *scope* of the measurement in OOmCFP is the OO-Method Conceptual Model from which the final software application will be built.

The *granularity level* of the measurement is low because all the details in the OO-Method conceptual model are needed to generate the applications.

The OO-Method software applications are generated according to a three-tier software architecture: the presentation tier (Client layer), the logic tier (Server layer), and database tier (Database layer) – see Figure 2. Each tier can be developed for different software environments.

As Figure 2 shows, the human user is a *functional user* of the client layer of the software and is separated from that layer by a *boundary*. In turn, the client layer of the software is a *functional user* of the server layer and is separated from that layer by a *boundary*. The server layer of the software is both a *functional user* of the client layer and a *functional user* of the database layer of the software and is separated from these layers by a *boundary*. In addition, the legacy systems are functional users of the server layer and are separated from that layer by a *boundary*.

The human user carries out the *triggering events* that occur in the real world. This user starts the *functional processes* that are direct successors of the hierarchy action tree (HAT) of the presentation model of OO-Method conceptual model. Each child represents a single functional process. The 'client user' starts the *functional processes*, which are the actions that the server layer carries out in response to the functional processes that occur in the client layer. The 'server user' starts the *functional processes*, which are the actions related to the database layer in response to the functional processes that occur in the server layer. The 'legacy user' starts the *functional processes*, which are the actions that the server layer carries out in response to the functional processes that occur in the legacy systems.

Every functional process has a set of *data movements* that can be entry data movements (E), exit data movements (X), read data movements (R) or write data movements (W). In the measurement guide[2] we have defined 69 mapping rules for the data movements that can occur in the OO-Method applications – see Figure 2.

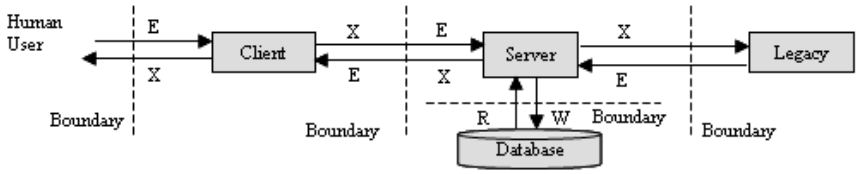


Fig. 2. Data movements that could occur in an OO-Method application.

Each single data movement moves one data group. The *data groups* are the classes of the object model of OO-Method that participate in a functional process. Every class has a set of attributes that will be the *data attributes* of a data group.

[2] http://oomethod.dsic.upv.es/labs/images/OOmCFP/guide.pdf

In the sub-step *definition of the numerical assignment rules*, we have determined the measurement rules for the data movements that can occur in an OO-Method application. One size unit, referred to as 1 *cfp*, will be assigned to each data movement. Thus, the functional size of a functional process will be the addition of the data movements that occur in that functional process. Next, the functional size of a layer will be the addition of the functional sizes of the functional processes that are contained in that layer. Last, the functional size of a generated OO-Method application will be the addition of the functional size of every layer of the application.

In terms of the validation of the OOmCFP procedure, we have verified how the measurement process works in practice using some predefined OO-Method conceptual models. In addition, we can infer that OOmCFP has been theoretically validated because the theoretical validation of COSMIC was carried out successfully in [2] using the DISTANCE framework. Moreover, an expert[3] has validated the conformity of the OOmCFP procedure with the COSMIC version 3.0.

3 Conclusions and Further Work

In this paper, we have presented OOmCFP, which is an FSM procedure based on COSMIC for object-oriented applications generated in MDA environments from their conceptual models. The design of OOmCFP was presented using a generic process model for software measurement. We consider that OOmCFP specifies the issues that must be considered for the development of a tool to automate the measurement of the functional size of applications generated in MDA environments. Further work includes empirical studies of the reproducibility and the repeatability of OOmCFP, and the creation of a tool that automatically implements OOmCFP.

References

1. Abran, A.; Desharnais, J.; Lesterhuis, A.; Londeix, B.; Meli, R.; Morris, P.; Oligny, S.; O'Neil, M.; Rollo, T.; Rule,G.; Santillo, L.; Symons, C.; Toivonen, H., *The COSMIC Functional Size Measurement Method, version 3.0* In GELOG web site www.gelog.etsmtl.ca
2. Condori-Fernández, N., *Un procedimiento de medición de tamaño funcional a partir de especificaciones de requisitos*, Doctoral thesis, Univ. Politécnica de Valencia, España, 2007.
3. Grau, G.; Franch, X., *Using the PRiM method to Evaluate Requirements Model with COSMIC-FFP* In: IWSM-MENSURA 2007, Mallorca, Spain, November 2007.
4. ISO, ISO/IEC 14143-1, *Information Technology – Software Measurement – Functional Size Measurement – Part 1: Definition of Concepts*, 1998.
5. ISO, ISO/IEC 19761, *Software Engineering – CFF – A Functional Size Measurement Method*, 2003.
6. Jacquet, J.P.; Abran, A., *From Software Metrics to Software Measurement Methods: A Process Model* In: 3rd International Standard Symposium and Forum on Software Engineering Standards, ISESS 1997, Walnut Creek, USA, 1997.
7. Pastor, O. and Molina, J. C., *Model-Driven Architecture in Practice*, Springer, 2007.

[3] We would like to thank to Charles Symons for his helpful comments.

Modeling and Executing Service Interactions using an Agent-oriented Modeling Language

Christian Hahn[1] and Ingo Zinnikus[1]

DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken
{Christian.Hahn,Ingo.Zinnikus}@dfki.de

Abstract. Modern information systems are considered as collection of independent units called services that interact with each other through the exchange of messages. This paper focuses on interactions from a more centralized or global perspective (i.e. choreography), validates the underlying approach to model interactions, and discusses how choreographies can be executed with an established agent-oriented programming language basing on the principles of model-driven development.

1 Introduction

Service-oriented architectures (SOAs) as an approach to design and implement modern information systems (ISs) aim to support business process management within an organization and across organizational borders. At this services are employed to perform tasks within these processes and processes themselves can be exposed as services. In these kinds of settings, service interactions are at the center of attention where two complementary perspectives can be distinguished.

Recently, several approaches have been proposed to describe the interaction between entities either from a local (e.g. Business Process Execution Language) or global perspective (e.g. Web Service Choreography Description Language). In this paper, we propose an agent-based approach as agent systems provide several built-in features and concepts that allow to execute SOAs in a nice manner (see [1] for more details).

A lot of effort has been undertaken to identify the most common interaction scenarios from a business perspective, which have been published as *service interaction patterns* by Barros et al. [2]. We take these patterns as a base and demonstrate how a platform independent domain specific modeling language for multiagent systems called Dsml4Mas fulfills the proposed requirements. Furthermore, we also aim at providing an agent-based model-driven methodology that allows executing choreographies.

In the remainder of this paper we discuss a selected interaction pattern and demonstrate how choreographies can be transformed to executable code by applying principles of model-driven development.

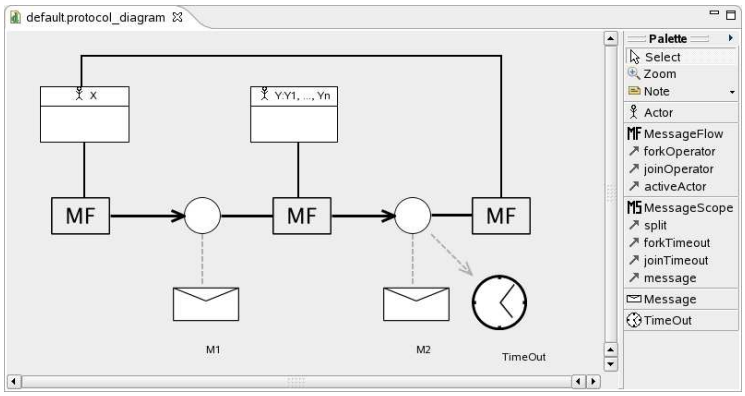


Fig. 1. Pattern 7: One-to-many send/receive modeled with the graphical editor of DSML4MAS.

2 A Platform Independent Modeling Language for Multiagent Systems

DSML4MAS defines a graphical language that could be used to define agent systems independent of any existing agent-oriented programming language (AOPL). However, model transformations can be applied to generate code with respect to selected AOPLs.

DSML4MAS is divided into several viewpoints (e.g. agent, organization, behavior, etc,), however, in this paper we mainly focus on the interaction aspect and demonstrate how to model the proposed service interaction patterns.

In general, an *Interaction* refers to a set of *Messages* and *Actors* that make use of these *Message* for the purpose of interaction. The *Actor* can again refer to a set of *Actors* as subactors, meaning that the set of instances performing the superactor is split into the several subactors. In general, the subactors are determined at design time, but filled with the particular instances that perform this kind of role at run-time. Furthermore, a *Protocol* that should be considered as a specialization of an *Interaction* refers to a set of *MessageFlows* that specify how the exchange of *Messages* is proceed.

The *MessageFlows* again refer to a set of *Actors* that are active in the current state, i.e. those instances that send the particular *Messages*. Furthermore, it specifies a join and fork operator which are both of the type *MessageScope* that defines the *Messages* and their order how these arrive. In particular, this means that *Messages* are connected via a *None, Parallel, Loop, Sequence, XOR*, or *OR* operator. Furthermore, the *MessageFlow* refers to a *TimeOut* that specifies the latest point in time a *Message* should arrive. Beside *Messages* that are sent, the *MessageFlow* may also refer to *Protocols* that are initiated at some specific point in time in the parent *Protocol* in order to execute nested protocols.

3 Modeling Service Interaction Patterns using DSML4MAS

Barros et al. [2] consolidate recurrent scenarios and abstract them in a way that provides reusable knowledge. They distinguish between four groups of patterns, however, we focus on the single-transmission patterns in which a party involved may send or receive multiple messages but as part of different interaction threads dedicated to different parties. A specific case is pattern 7: One-to-many send/receive. Here, a party X sends a request message to several other parties Y1,...,Yn, which may all be identical or logically related. Responses are expected within a given timeframe. However, some responses may not arrive within the timeframe. The interaction may complete successfully or not depending on the set of responses gathered. Fig. 1 depicts the one-to-many send/receive pattern using DSML4MAS. The parties are again modeled as *Actors*, where the atomic entities Y1,...,Yn are bound to *Actor* Y. Sending a *Message* to an *Actor* means that the particular *Message* is sent to each instance that is bound to the target *Actor*. This means that *Message* M1 is sent to each of the Y1,...,Yn in parallel. When receiving M1, each of these entities sends the corresponding answer *Message* M2 to *Actor* X. A *TimeOut* ensures that the interaction does not end up in a deadlock.

4 Model-driven Methodology to Generate Executable Code

In this section, the transformation from the local perspective to the agent-based execution platform JACK is given. JACK is a process-centric agent-based programming language that bases on principle of the belief-desire-intention theory [3]. We firstly introduce the core concepts of JACK. Due to space restrictions this is a very rough summary, however, a detailed overview regarding the JACK metamodel can be found in [4]. The most relevant concept in JACK is the concept of a *Team*, which can be either an atomic *Agent*, or a set of required *Roles* (i.e. subteams) that all together form the *Team*. A *Role* specifies which *Events* the role fillers are able to react to and send. How a *Team* actually reacts to an incoming request is specified by a set of *TeamPlans*.

The transformation to the JACK metamodel uses pattern 7 as an input model. We generate a *Team* for each *Actor* that performs a particular role (e.g. Role_X for team X) and requires *Roles* (e.g. Role_Y) to which the *Messages* in the DSML4MAS behavior model are sent (cf. Fig. 2). The *Messages* of the DSML4MAS model are mapped to *Events* in JACK (e.g. event M1 and M2). For each *Plan* in the behavior model, we instantiate a *TeamPlan* that is used by the particular *Team* (e.g. XSendM1 and XReceiveM2). The body of the *TeamPlans* is mainly generated in an one-to-one manner from the *Plans* in DSML4MAS. For instance, the XSendM1 *TeamPlan* (Fig. 2 right-hand side) also includes a parallel statement that iterates over the various role fillers and sends the event instance m1 of M1 to the role fillers y. The parallel statement ends if the *Event* has been sent to all role fillers. We refer to [4] for more detailed information regarding the model transformation from DSML4MAS to JACK.

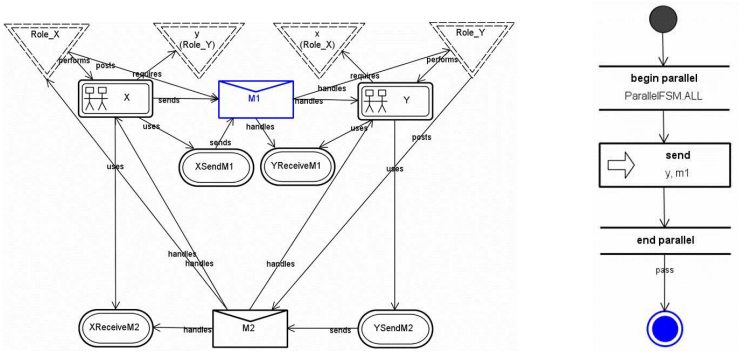


Fig. 2. The generated JACK models based on the DSML4MAS behavior for Pattern 7.

5 Conclusion

This paper discusses an agent-based approach to describe choreography-based interactions. Therefore, we proposed a modeling language for multiagent systems called DSML4MAS and demonstrated that DSML4MAS supports modeling the proposed service interaction patterns. The main result of this evaluation is that each pattern—in contrast to other proposed standards—can nicely be described.

Based on DSML4MAS, we discussed a model-driven methodology to derive code based on the choreography description. The DSML4MAS model that includes the particular generated behavior model is mapped to an agent-based programming language JACK that finally executes the choreography description.

References

1. Zinnikus, I., Hahn, C., Klein, M., Fischer, K.: An agent-based, model-driven approach for enabling interoperability in the area of multi-brand vehicle configuration. In: Proceedings of the 5th Conference on Service-Oriented Computing. Volume 4749 of Lecture Notes in Computer Science., Springer (2007) 330–341
2. Barros, A.P., Dumas, M., ter Hofstede, A.H.M.: Service interaction patterns. In van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management. Volume 3649. (2005) 302–318
3. Rao, A.S., Georgeff, M.P.: Modeling agents within a BDI-architecture. In Fikes, R., Sandewall, E., eds.: Proceedings of the 2rd International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann (1991) 473–484
4. Hahn, C.: A domain specific modeling language for multiagent systems. In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2008) (accepted).

Model Transformations powered by Rewriting Logic *

Francisco J. Lucas and Ambrosio Toval

Software Engineering Research Group
Department of Informatics and Systems
University of Murcia (Spain)
fjlucas@um.es,atoval@um.es

Abstract. This paper shows a rigorous approach based on algebraic specifications and rewriting logic which makes up for the lack of current transformation languages and offers a balanced rigour-versus-intuition framework for model transformation, focusing on the MDA-QVT standards. To illustrate this approach, an example and some formal applications of these specifications are sketched.

1 Introduction

In recent years, the profound impact of the Model Driven Architecture (MDA) proposal [1], promoted by the OMG as architecture for software development, has meant that the model transformation becomes a very active research and development direction. Within this scope, OMG also published the QVT standard [2], a language for the specification of model transformations within the MDA scope. Since transformations guide the whole software development cycle, it is crucial to offer a precise and rigorous infrastructure, in order to help to verify and guarantee correctness of them. However, current implementations of transformations languages lack this necessary mathematical underpinning.

The aim of this short paper is: on the one hand, to show how a rigorous approach based on algebraic specification and rewriting logic [3] can offer a suitable framework for model transformations; and on the other hand, to show how proving theoretical properties of transformations is possible if the transformation language or tool has a mathematical underpinning [4]. *Maude* [5] is the formal language used in this work. To illustrate this approach, the formal specifications of two metamodels, and a transformation between them, have been created.

2 Model Transformation based on a Rewriting Logic Approach

The main idea of this work is to specify metamodels through OO Maude modules, and the transformation rules as rewriting rules that rewrite source model

* Partially financed by the Spanish Ministry of Science and Technology, project DEDALO (Development of Quality Systems based on Models and Requirements) TIN2006-15175-C05-03

elements (represented as terms) into other target model elements. In this section, as well as explaining the main principles of this approach, we will illustrate it by means of an example of transformation from UML Class diagram to a RDBMS diagram (extracted from [2]).

2.1 Metamodel Formalization

In QVT, model transformations are defined in terms of metamodels, existing source and target metamodels. Metamodel elements will be specified in *Maude* by means of the object oriented modules of *Maude*. Figure 1 summarizes the elements that make up the approach. This Figure shows the OMG standards' elements, the *Maude* elements and the relationship between them.

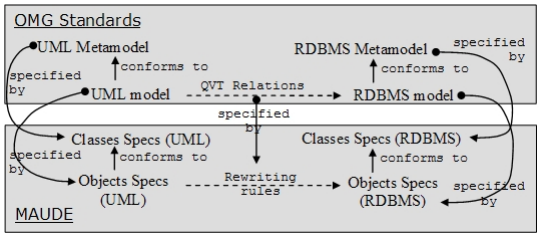


Fig. 1. Summary of the approach elements

The first metamodel specified is the so-called simple UML metamodel (taken from annex A of [2]) which represents a simplified version of the UML Class diagram. Each element of this metamodel is specified by means of a *Maude* class. Figure 2 shows an example of a simple UML diagram and how it is expressed by means of objects in *Maude*; these objects are instances of *Maude* classes that appear in the lefthand side of the figure, as it was indicated in Figure 1.

Analogously, the textual description of the simple RDBMS metamodel (taken from annex A of [2]) will be formalized in *Maude*.

2.2 QVT Relations Features in *Maude*

In this subsection, we analyze briefly the basis of QVT Relations and how to formalize them by means of the strengths offered by *Maude*.

On the one hand, a transformation is expressed in QVT Relations by means of relations between metamodel elements. A relation declares constraints that must be satisfied by the two or more metamodels (or domains) that participate in the relation. Each domain establishes a pattern that must be matched with the candidate models in order to execute the transformation, known as *object template expressions* that are directly expressed in *Maude*, since this language offers pattern-matching in the simplification of terms. Regarding the specification of QVT transformations in *Maude*, they can be specified as rewriting rules that

change and create the elements of the target model. Finally, constraints over the candidate models will be specified as conditions in the rewriting rules.

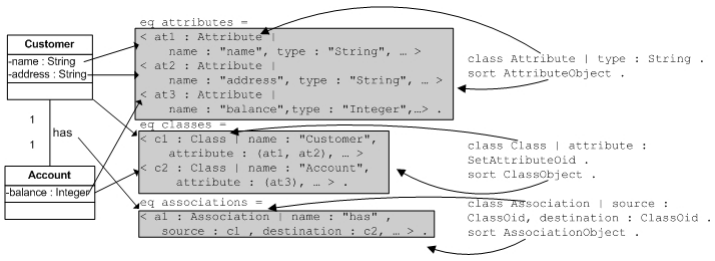


Fig. 2. Example of UML diagram and their corresponding Maude objects

2.3 Simple UML to Simple RDBMS

In this section, we will study the transformation from UML diagrams to RDBMS diagrams (taken from [2]). Basically this transformation has three main (top-level) relations: Package to Schema, Class to Table and Association to Foreign Keys. Due to lack of space only the first relation will be studied.

Package to Schema relation transforms a package of a UML diagram into a schema of a relational data base diagram. Figure 3 shows this relation expressed in QVT (a) and how it is specified in *Maude* (b). In this relation, the object template expression can be "directly" expressed in *Maude*. The pattern matching binds the variable "pn" to a specific value that is used to create a new object which represents a schema in the target model. On the other hand, since a model is represented by means of objects in *Maude*, we have to use object identifiers in the rules. In the righthand side of the rule appears both the object of the source model and the new object in the target model. Finally, once we have specified the metamodels and the QVT Relations in *Maude*, we can execute the transformation over any UML model.

3 Applications in Practice

The main advantage of the use of this approach over other non-formal transformation techniques is that some applications can be carried out only defining the rewriting rules. The application shown checks if two models are semantically equivalent. In general, various models are semantically equivalent if they have similar meaning. Being more precise, in this work we consider that two models are semantically equivalent if they hold all the equivalence relationships (defined by means of QVT Relations) which, depending on the metamodels, express the semantics equivalence of concepts as defined by the analyst. In this way, if the

```
(a) top relation PackageToSchema {  pn: String;
      checkonly domain uml p:Package {name=pn};
      enforce domain rdbms s:Schema {name=pn}; }
(b)
rl [PackageToSchema] :
< packageOid(p:String) : Package | domain : "uml", name : pn:String,...> =>
  < packageOid(p:String) : Package | domain : "uml", name : pn:String,...>
  < schemaOid(p:String) : Schema | domain : "rdbms", name : pn:String > .
(c) (search UMLdiagram =>! C:Configuration C2:Configuration
       such that (C:Configuration := RDBMSdiagram) .)
```

Fig. 3. (a) and (b) Package to Schema (*(a)* adapted from [2]); (c) Maude comand to check if two models are equivalent

semantic equivalence between two models is expressed as a relation, we can use *Maude* to infer if two particular models are equivalent using these rules.

If we define a RDBMS model that is equivalent to the one in Figure 2, Figure 3 (c) shows the *Maude* command that checks this equivalence. We ask *Maude* if it is possible to obtain the *"RDBMSdiagram"* model from the *"UMLdiagram"* model using the specified rules. This execution will find a solution since the models are equivalent.

4 Conclusions

The research presented shows the feasibility of integrating formal techniques with current software engineering standards (MDA-QVT). This approach may be particularly useful in model-driven engineering processes to develop critical or error-prone high quality systems. The metamodel specifications made in this approach offer a powerful way to verify type properties and the correctness of the models without losing the legibility and practicality of other transformation languages. Furthermore, in the formal framework proposed the transformations are represented as mathematical entities and we can take advantage of all the power of mathematical inference mechanisms. This allows us to infer information and to prove properties of the transformations.

References

1. OMG: MDA Guide Version 1.0.1, http://www.omg.org/mda. (2001)
2. OMG: MOF QVT Final Adopted Specification. Object Management Group., Retrieved from: http://www.omg.org/docs/ptc/07-07-07.pdf. (2007)
3. Meseguer, J.: Conditional rewriting logic as a unified model of concurrency. Theoretical Computer Science, 96(1):73-155 (1992)
4. Mens, T., Czarnecki, K., Van Gorp, P.: A Taxonomy of Model Transformations. Int. Workshop on Graph and Model Transformation (GraMoT). Estonia (2005)
5. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcote, C.: Maude 2.3 Manual., http://maude.csl.sri.com/. (2007)

Semi-automated Model Synchronisation in SOM

Christian Flender and Thomas Hettel

c.flender@qut.edu.au, t.hettel@qut.edu.au

Faculty of Information Technology,
Queensland University of Technology,
Brisbane, Australia.

Abstract. Model-driven engineering is at the forefront among recent attempts to information systems development. Models are gradually refined from domain specific descriptions to more concrete models closer to implementation. This is particularly relevant to the model transformation of collaborating business partners down to collaborating (web) services as they share a common interactional perspective. However, as requirements constantly evolve model layers change and so they have to be kept in sync. Model synchronisation keeps track of those changes and propagates them to other layers. This poster gives a brief introduction to model synchronisation as devised for the Semantic Object Model (SOM), a promising approach to model-driven service engineering. SOM allows for the gradual refinement of model layers through decomposition of business objects respective their interactional relationships.

1 Introduction

Model-driven development of information systems bares the indispensability of architectural frameworks. Architectures divide a complex model into several model layers and perspectives so as to reduce the amount of aspects to be considered at a point in time. Transformations do not only allow for the gradual refinement of a given layer but also hold together the whole system as a coherent architecture. Furthermore, it is the availability of Service-oriented Architecture (SOA) that increases the attractiveness of model-driven engineering as collaborative aspects gain relevance for both domain modelling (e.g. business process design) and implementation focus (e.g. compositions of web services). However, constantly changing requirements enforce changes of model layers. For instance, changes in the provision of business services (e.g. order processing) must be mirrored in terms of changes in the provision of technological services (e.g. automatic order entry). Models should be kept in sync without violating the consistency of the overall architecture. In a semi-automatic fashion, users are guided through all layers to apply dependent changes step-by-step.

The Semantic Object Model (SOM) [1] is an approach to model-driven service engineering. SOM allows for the deduction of executable process models from high-level networks of interacting business partners [3]. This poster presents the enhancement of SOM via change propagations for model synchronisation.

2 The Semantic Object Model (SOM)

The backbone of the Semantic Object Model (SOM) is an enterprise architecture as shown in Figure 1. The architecture is divided into three main layers. The enterprise plan defines the business system from an outside view in terms of its goals, objectives and strategies embedded in a broader socio-cultural context. From an inside view, the business process model implements the enterprise plan. It is specified as a system of interacting business objects which coordinate behaviour in purposefully providing and consuming services via transactions. Once a network of collaborating actors is decomposed down to a sufficient level of detail, resource assignments embody the system in terms of human actors and web-enabled software components (implementation support).

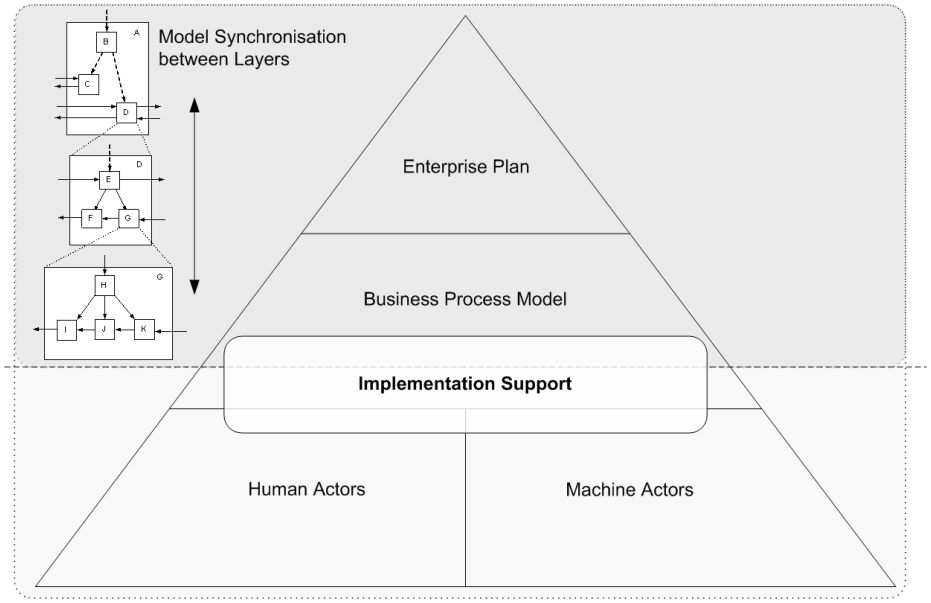


Fig. 1. The Semantic Object Model (SOM).

It is the business process model that can be further refined by decomposing interactions and objects. Interactions between objects are typed according to the coordination principles *negotiation* and *feedback-control*. Negotiation defines relationships between objects as either initiating (I), e.g. make offer, contracting (C), e.g. accept order, or enforcing (E), e.g. deliver service. Feedback-control relates objects through control transactions (R), e.g. give advice, and feedback transactions (F), e.g. state report. Model layers emerge in gradually refining networks of interacting objects. This is done by applying patterns of object-relations such as ICE, CE, RF or more complex combinations. For instance, consider the

decomposition of an object Supplier in two objects Sales and Retailer. In replacing Supplier relations between Sales and Retailer require the former to give sales advices (R) and the latter to confirm sold products (F). Following coordination principles in this manner new model layers emerge being in relationship with each other. Hence, transformations between layers constitute the trace of decomposition steps applied for their creation. However, once an architecture is modelled, requirements may change. For instance, industrial development of new products, services or markets may enforce new structures and relationships between managerial and operational actors. To avoid developing architectures anew from scratch each time requirements change, model synchronisation keeps track of changes.

3 Model Synchronisation in SOM

Consider the example of a Buyer- Supplier network. In the initial model layer as shown on the top-left side in Figure 2, both objects negotiate according to their needs. In the second model layer, Supplier was decomposed in two objects Sales

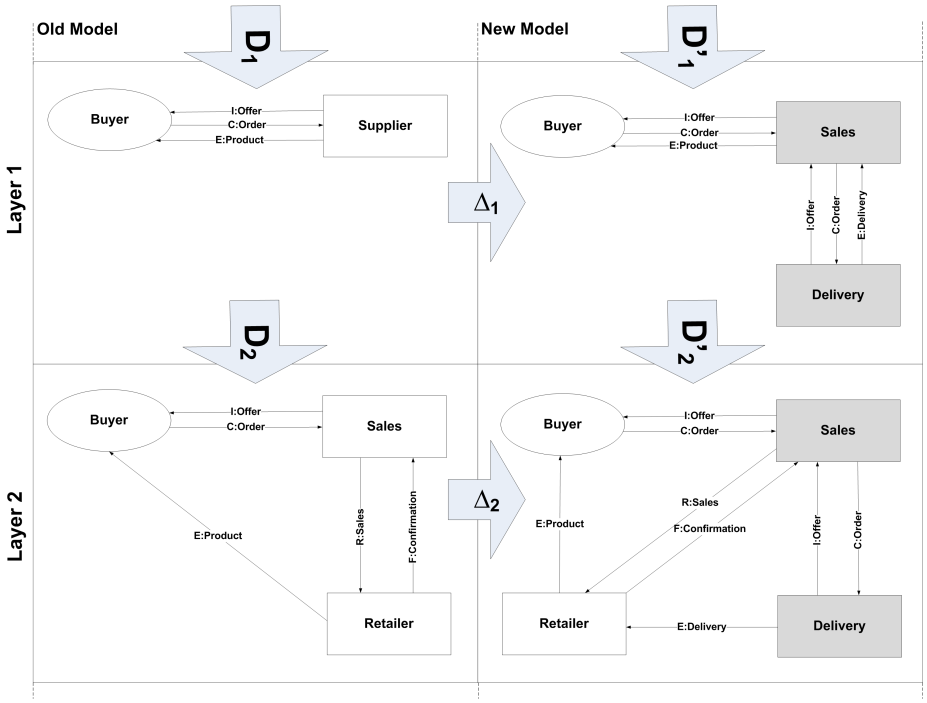


Fig. 2. Model Synchronisation in SOM.

and Retailer (see Figure 2 on the bottom-left side). The latter sells the product

to Buyer from stock according to advice from a salesman. However, in order to increase the range of products, the management decides to purchase selected items from external sources. These products must be delivered in order to resell them to buyers. On the top-right side in Figure 2 a new object Delivery is inserted according to the changed requirements. This enforces change propagations so as to keep Layer 1 and Layer 2 in sync. If one compares Layer 2 of the old model and Layer 1 of the new model, propagations result from both the decompositions which led to Layer 2 (D_2) and the the insertion of Delivery in Layer 1 (Δ_1).

We devised a complete set of change propagation algorithms keeping track of insertions, deletions and updates [2].

4 Conclusion and Future Work

The gradual refinement of model layers decreases complexity as systems are divided into manageable parts without loosing track of the whole architecture. Model-driven service engineering can unfold its full potential when architectures evolute toward sustainability. Therefore, mechanisms are needed assuring the propagation of changes on a particular layer to other layers so as to maintain consistency. SOM's transformation rules are derived from composable patterns of coordinations between objects. Having enhanced SOM's enterprise architecture with change propagations, model synchronisation introduces flexibility and accounts for evolving requirements. We are working on a SOM tool which is partly available for presentation.

References

1. O. K. Ferstl and E. J. Sinz. *Handbook on Architectures of Information Systems.*, chapter Modeling of Business Systems Using the Semantic Object Model (SOM) - A Methodological Framework. International Handbook on Information Systems. 1997.
2. C. Flender and T. Hettel. Model-driven Service Engineering with SOM. Technical report, FIT-TR-2008-02, Queensland University of Technology, Brisbane, 2008.
3. C. Flender and M. Rosemann. Service-oriented Design of an Enterprise Architecture in Home Telecare. In *18th Australasian Conference on Information Systems*, 2007.

Modelling Quality and Spatial Characteristics for Autonomous e-Service Peers

Jun Shen, Shuai Yuan
Centre for Information Systems and Technology Research
School of Information Systems and Technology
Faculty of Informatics, University of Wollongong, Australia
Wollongong 2522, Australia

jshen@uow.edu.au, shuai.yuan@hotmail.com

Abstract. In this paper, we present an autonomous and scalable WSMO-based methodology to describe quality of service (QoS) and geographic features of e-services in a peer-to-peer based environment. To fully explore the usability of service mining and categorisation, we designed an algorithm to select the most appropriate peers to improve effective service composition.

1. Introduction

It is problematic that traditional methodologies can not effectively and autonomously conduct service discovery and composition in a complex dynamic environment. Even though quite a few groups proposed numerous QoS specifications, most of them are extremely difficult to clarify the correlation between one another consistently.

In this context, we present an intelligent, autonomous and scalable ontology-based methodology to describe QoS and geographic features of Web services in a P2P-based environment. Moreover, semantic Web services selection is a process to automatically find appropriate Web services that effectively fulfil the requestor's requirements. Hence, we design and implement an algorithm to reasonably deal with the correlation between those requirement specifications, and select the most appropriate peers to foster a better service composition. In section 2 we introduce the design steps of modelling method. After comparing the related work in Section 3, our conclusions will be addressed with future work in Section 4.

2. Design and Modelling

In order to evaluate different non-functional properties of e-service peers, there are three important concepts in our design: *PreferedValueType*, *Weight*, and *Unified Value*. *PreferedValueType* has two kinds of values: "low" and "high". With regard to "*Weight*", it indicates the importance and priority of certain properties during the service composition. "*Unified Value"* indicates the each peer's overall quality with numerically indicating results.

If "*PreferedValueType*" = "high", then the property ratio (PR) of a peer's service should be calculated by:

$$PR(i, j) = \frac{nf(i, j) - nf(min)}{nf(max) - nf(min)} \quad (1)$$

"*PR(i,j)*" presents the ratio value of non-functional Property(j) of Peer(i), and "*nf*" stands for non-functional. nf(min) and nf (max) refer to the minimum and maximum value of the Property(j) among all relevant peers. On the contrary, if "*PreferedValueType*" = "low", then the ratio should be determined according to:

$$PR(i, j) = \frac{nf(max) - nf(i, j)}{nf(max) - nf(min)} \quad (2)$$

Our main aim is to scale the value ranges with the maximum and minimum values by this means. Hence, any value with different "*PreferedValueType*" can be converted into the standardised value between 0 and 1. Through this approach, every property of each peer can be compared and evaluated fairly and also quickly. Subsequently, all candidate peers' non-functional properties would be put in a matrix, looks like (for *n* properties in *m* peers):

$$Mnf = \begin{bmatrix} PR(1,1) & PR(1,2) & PR(1,3) & \dots & PR(1,n) \\ PR(2,1) & PR(2,2) & PR(2,3) & \dots & PR(2,n) \\ PR(3,1) & PR(3,2) & PR(3,3) & \dots & PR(3,n) \\ \dots & \dots & \dots & \dots & \dots \\ PR(m,1) & PR(m,2) & PR(m,3) & \dots & PR(m,n) \end{bmatrix}$$

"*Mnf*" refers to matrix of non-functional properties. For uniformity, matrix *Mnf* has to be normalised to map all real values to a relatively small range through equations (1) (2), i.e., all elements of the final matrix are real numbers in the closed interval [0, 1]. Having *Weight (W)* values assigned to each property, we apply the following equation to generate the "*Unified Values (UV)*" for each peer:

$$UV = Mnf \times W, i.e., UV(i) = \sum_{j=1}^{n} (PR(i, j) \times w(j)), i = 1..m \quad (3)$$

w(j) stands for a weight value of different property (j^{th}) for service composition. As a result, it is reasonable to indicate which peer (i^{th}) would be able to conduct a specific task more effectively, by means of achieving the highest value *UV(i)*, *i* ranges from *1 to m*. With regard to WSMO [4] extension, based on [6], we define an extensible class QoSProperty which aims to extend nonFunctionalProperties class in WSMO for P2P-based service selection [7].

```
Class nonFunctionalProperties
      ...other existing properties...
      hasQoSProperty type QoSProperty

Class QoSProperty sub-Class nonFunctionalProperties
```

```
        hasPropertyName type string
        hasPropertyValue type {int, float, long, others}
        hasPreferedValueType type {low, high}
        hasWeight type float
```

In order to effectively enhance services' quality regarding accessibility in P2P network, we herein consider basic geographic information about a would-be task-allocated peer and incorporate it into the QoS profile as an extension of previous QoS specification

```
    Class GeoProperty sub-Class QoSProperty
        hasGeoName type string
        hasGeoValue type {int, float, long, others}
        hasPreferedValueType type {low, high}
        hasWeight type float
        isEssential type boolean
```

For a peer selection process, we designed an algorithm. This algorithm aims to address the selection method with multiple peer profile specifications, and facilitate the above modelling approach. The algorithm can also be used for service/peer matchmaking, since we may set a goal for each QoSProperty if necessary. The following is the pseudo code:

```
Begin Function Mining Peers (P1, P2, … Pm)
    for i=1 to m do
        getQoSProperties(Pi);
        normalise input (Pi) using equation (1)/(2);
        then store the normalised value into array (Mnf);
    end
    getWeight() for the different properties;
    calculate the unified values by using equation (3;
    choose Pi with maximum unified value;
    return (Pi);
end function
```

3. Related Work

Functionality and non-functional properties are two essential aspects for semantic Web service. Functionality is used to measure whether this Web service meets all the functional requirements of an anticipated Web service, i.e. Web services matchmaking; while non-functional properties are qualified to evaluate the performance of the Web service. This has been viewed as a sufficient means to distinguish functionally similar Web services. For example, [3] and [1] emphasized a definition of QoS aspects and metrics. In [3], all of the possible quality requirements were introduced and divided into several categories, including runtime-related, transaction support related, configuration management and cost related, and security-related QoS. Both of them shortly present their definitions and possible determinants. Unfortunately, they failed to present a practical methodology for real applications. In [2] and [6], authors focused on the creation of QoS ontology models, which proposed QoS ontology

frameworks aiming to formally describe arbitrary QoS parameters. From their on-going work, we are aware that they did yet consider QoS-based service selection. Additionally, our approach is built by taking considerations of new intuitive correlations between various service quality measurements and also testified upon a well-founded peer-to-peer e-service workflow system, which the authors have developed in the past [5].

4. Conclusion and Future Work

In this paper, we discussed the importance of QoS and spatial specification for P2P-based service mining and selection, and presented a comprehensive analysis on non-functional properties in WSMO. We augmented WSMO description by involving QoS perspectives and geographic profiles. We also designed and implemented an effective algorithm to facilitate the peer selection. Within the near future, our service peer selection model is expected to be modernized by focusing on concrete and detailed geographic features for location-based services, and we will improve our prototype for P2P-based workflow under a dynamic circumstance more effectively.

References

1. Lee, K., Jeon, J., Lee, W., Jeong, S. and Park, S.: QoS for Web services: Requirements and Possible Approaches. W3C Working Group Note 25, 2003. Available at: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/.
2. Papaioannou, I.V., Tsesmetzis, D.T., Roussaki, I.G. and Miltiades, E.A.: QoS Ontology Language for Web-Services. In proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Volume 1, pp.18-20, IEEE Press.
3. Ran, S.: A model for Web services Discovery with QoS. ACM SIGecom Exchanges, 4(1): 1-10.
4. Roman, D., Keller, U., Lausen, H. (eds.): Web Service Modeling Ontology. Applied Ontology, 1(1): 77-106, 2005.
5. Shen, J., Yang, Y. and Yan, J.: A P2P based Service Flow System with Advanced Ontology-based Service Profiles. Advanced Engineering Informatics, 21(2): 221-229, 2007.
6. Tsesmetzis, D.T., Roussaki, I.G., Papaioannou, I.V. and Anagnostou, M.E.: QoS awareness support in Web-Service semantics, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), p.128.
7. Yuan, S. and Shen, J.: QoS Aware Service Selection in P2P-Based Business Process Frameworks. In Proceedings of the 9th IEEE Conference on E-Commerce Technology and the 4th IEEE Conference on Enterprise Computing, E-Commerce and E-Services (CEC/EEE'07), Tokyo, Japan, pp.675-682.

Timed Transition Discovery from Web Service Conversation Logs

Didier Devaurs[1], Kreshnik Musaraj[2], Fabien De Marchi[2], and Mohand-Saïd Hacid[2]

[1] University of Windsor, School of Computer Science, Windsor, Ontario, Canada
ddevaurs@uwindsor.ca

[2] Université Claude Bernard Lyon 1, LIRIS, UMR CNRS 5205, Villeurbanne, France
{kreshnik.musaraj, fabien.demarchi, mohand-said.hacid}@liris.cnrs.fr

Abstract. Despite their importance, Web service business protocols are not always published with service interfaces, which hinders automatic management. A solution is to extract them from past executions. One of the raised issues is the discovery of temporal constraints called *timed transitions*, which are not explicitly recorded. In this paper we present our approach for discovering such transitions. We define a class of patterns called *proper timeouts* which are equivalent to timed transitions, and present a polynomial algorithm for extracting these patterns.[3]

Keywords: Web service, business protocol, knowledge extraction, temporal constraint

1 Introduction

A very important ambition associated with Web services relates to loosely-coupled integration, which is already partially carried out by the fact that services use widespread standards. A good flexibility is possible only if users know how to interact with a service. This requires to associate with services elaborate descriptions (such as WSDL) enabling a good understanding of their execution semantics. However descriptions like WSDL are not sufficient for a sophisticated and automatic use of services because they provide only static properties [1]. This is what motivated authors in [1] to define a higher level model, the so-called *business protocol*, which specifies the conversations supported by a service, i.e. all valid sequences of message exchanges. It is formalized by a deterministic finite-state machine, where states represent the various service phases; transitions are triggered when the service sends or receives messages. A *timed business protocol* [2] is an enhanced version of the basic model allowing for the definition of *timed transitions*, which are not related to the emission of explicit messages but to temporal constraints (validity period, expiration date, etc); they are triggered automatically after a time interval is elapsed or after some date is reached.

[3] This work is partially funded by the ANR project Service Mozaïc (2007–2009, JCJC06_134393) and by the EU Framework 7 STREP project COMPAS (215175, FP7-ICT-2007-1).

Business protocols offer automatic reasoning mechanisms with many applications, such as correctness verification, compatibility testing, etc. However they are not often specified in real life services. Potential reasons include lack of time during implementation or uncontrolled evolution. A solution is then to infer this protocol from the *conversation logs* of a service. Direct applications are re-engineering issues, such as implementation correctness checking or service evolution. Once automated this extraction process could be applied in service discovery architectures [3] for automatic service composition or replacement.

Discovering service protocols includes many technical challenges: cleaning logs from "noise", identifying the different conversations, defining assessable models, developing refining tools for an interactive extraction, etc. The first contribution to this problem has been proposed in [4], but relates only to *untimed* business protocols. With the importance of temporal aspects in real life services it becomes crucial to extend this work to timed business protocols, which contain both explicit and timed transitions.

This paper presents our approach for extracting timed transitions from conversation logs.[4] We define a class of patterns called *proper timeouts* which reveal the presence of timed transitions in the protocol. We propose a characterization of the set of proper timeouts satisfied by the logs, which leads to a polynomial extraction algorithm. This work is an extension of [4], and both take part in *ServiceMosaic* international project (http://servicemosaic.isima.fr) which aims at developing a platform for modeling, analysing and managing Web services [6].

2 Associating Patterns with Timed Transitions

We define an **episode** as a sequence of two message names. Given an episode $\alpha = \langle m, m' \rangle$, an **occurrence** of α is a sequence of two consecutive occurrences of m and m' in the logs. The occurrence duration of an episode occurrence is the difference between the message timestamps. The minimal (respect. maximal) occurrence duration of an episode is the smallest (respect. greatest) occurrence duration of all its occurrences. The **occurrence duration interval** (ODI) of an episode is the interval which includes all its occurrence durations. The minimal (respect. maximal) occurrence duration of a set of episodes is the minimum (respect. maximum) of all the minimal (respect. maximal) occurrence durations of these episodes. The occurrence duration interval (ODI) of a set of episodes is the interval which includes all the occurrence durations of these episodes. For each message m, we denote by P_m the set of episodes whose first message is m.

Given two sets of episodes A and B, we say that A **precedes** B (denoted by $A \prec B$) if $ODI(A)$ is before $ODI(B)$.[5] We say that A and B are not comparable (denoted by $A \parallel B$) if $A \not\prec B$ and $B \not\prec A$. Given $A, B \subset P_m$, we show that: if there exists a timed transition between the state from which the transitions corresponding to the elements of A are going out, and the one from which the

[4] Technical results are presented in an extended version of this paper [5].

[5] $\prec$ is a strict order relation on sets of episodes.

transitions corresponding to the elements of B are going out, then $A \prec B$.

We define a **proper timeout** as a triplet $PT(m, A, B)$, where m is a message and $A, B \subset P_m$. We say that logs L satisfy the proper timeout $PT(m, A, B)$, which is denoted by $L \vDash PT(m, A, B)$, if:

$$\begin{cases} A \prec B \\ \forall\, \alpha \in P_m \setminus (A \cup B),\ \{\alpha\} \nparallel A \cup B \\ \forall\, Z \in \{A, B\},\ \forall\, X, Y \subset Z\ (X, Y \neq \phi),\ (X \cup Y = Z) \Rightarrow (X \not\prec Y)\ . \end{cases} \tag{1}$$

Given a message m and $A, B \subset P_m$, we show that: if there exists a timed transition in the protocol, between two states s_1 and s_2 such that the sets of transitions going out of s_1 and s_2 respectively are in bijection with A and B, then there exist $A' \subseteq A$ and $B' \subseteq B$ such that $L \vDash PT(m, A', B')$. Since each timed transition involves the satisfaction of a proper timeout, we can find all of them. However we can discover more proper timeouts than there are timed transitions, if some messages always take longer to be sent or received than messages associated with other transitions of the same state. Thus we will say that: a satisfied proper timeout reveals the presence of a *potential* timed transition.

We show that: for practical purposes, proper timeouts are the best possible representations of timed transitions. That justifies the relevance of the development of a timed transition discovery method based on the research of the proper timeouts satisfied by the logs.

3 Extracting the Proper Timeouts

The complexity of a basic "generate and test" method for extracting proper timeouts is exponential. Instead, we propose a nice characterization of the set of satisfied proper timeouts, which leads to a polynomial algorithm. This characterization, formalized by Theorem 1, states that: the proper timeouts satisfied by the logs and related to message m are exactly given by the pairs of consecutive elements of the partition of P_m satisfying (2). Thus, partitioning all sets P_m gives us all the proper timeouts satisfied by the logs.

Theorem 1. *Consider a mesage m, $i_m \in \mathbb{N}^*$, and $\{P_m^{(1)}, P_m^{(2)}, \ldots, P_m^{(i_m)}\}$ a partition of P_m. The following assertions are equivalent:*

$$\begin{cases} P_m^{(1)} \prec P_m^{(2)} \prec \ldots \prec P_m^{(i_m)} \\ \forall\, 1 \leq i \leq i_m,\ \forall\, X, Y \subset P_m^{(i)}\ (X, Y \neq \phi),\ (X \cup Y = P_m^{(i)}) \Rightarrow (X \not\prec Y)\ . \end{cases} \tag{2}$$

$$\begin{cases} \forall\, 1 \leq i < i_m,\ L \vDash PT(m, P_m^{(i)}, P_m^{(i+1)}) \\ \forall\, A, B \subset P_m,\ L \vDash PT(m, A, B) \Rightarrow \exists\, 1 \leq i < i_m,\ A = P_m^{(i)},\ B = P_m^{(i+1)}\ . \end{cases} \tag{3}$$

We propose a polynomial algorithm, called *partition*P_m, for constructing this partition in an incremental way. The input of algorithm *partition*P_m comprises

a message m, the set P_m, and the ODIs of all episodes in P_m. The output is the partition Π of P_m satisfying (2). Π is constructed by inserting one by one the elements of P_m in such a way that (2) is satisfied at each step. In order to describe the general step of the algorithm, let us consider that Π is already partly constructed. Let α be an episode of P_m not yet considered. A single pass is made over the partition to determine (i) whether the ODIs of some elements of Π overlap $ODI(\alpha)$, and (ii) between which sets of Π α is situated according to $\prec$. If there is no overlap, a new set containing α is created and inserted into the partition in compliance with $\prec$. If the overlap takes place with only one element of Π, α is simply inserted in this set. If the overlap occurs between α and several parts of Π, they are necessarily consecutive according to $\prec$; as such they are merged and α is inserted into the resulting set. As for each episode $\alpha \in P_m$ only one pass is made over the partition, the complexity is $\mathcal{O}(|P_m|^2)$.

The global method for extracting all the proper timeouts satisfied by the logs is divided in two steps. The first one is a preprocessing of the data, performed in order to obtain the set of messages, the set of episodes, and the ODIs of all episodes. A single pass is made over the logs, during which the occurrence duration of each sequence of two consecutive messages is calculated. The second step consists in constructing all sets P_m, and running algorithm $partitionP_m$ for each of them. The logs' size being far greater than the number of episodes, the first step is the most costly in term of running time. Thus the complexity of the global algorithm is $\mathcal{O}(|L|)$.

We have implemented our discovery process to test its scalability. In order to easily have a big amount of data, we have also implemented a log generator which creates conversation logs from a given business protocol by mimicking the behaviour of a service. Results of our experiments confirm the complexity results we have established formally. The final test will be to run our algorithm on real-life data in further experiments.

References

1. Benatallah, B., Casati, F., Toumani, F.: Representing, analysing and managing web service protocols. Data & Knowledge Engineering **58**(3) (2006) 327–357
2. Ponge, J., Benatallah, B., Casati, F., Toumani, F.: Fine-grained compatibility and replaceability analysis of timed web service protocols. In: ER '07. (2007) 599–614
3. Denaro, G., Pezzé, M., Tosi, D., Schilling, D.: Towards self-adaptive service-oriented architectures. In: TAV-WEB '06, Portland, Maine, USA, ACM (2006) 10–16
4. Motahari Nezhad, H.R., Saint-Paul, R., Benatallah, B., Casati, F.: Protocol discovery from imperfect service interaction logs. In: ICDE '07. (2007) 1405–1409
5. Devaurs, D., Musaraj, K., De Marchi, F., Hacid, M.S.: Timed transition discovery from web service conversation logs (extended version). Technical Report RR-LIRIS-2008-007, LIRIS UMR 5205 CNRS/Université Claude Bernard Lyon 1, Villeurbanne, France (2008) http://liris.cnrs.fr/publis/?id=3369.
6. Benatallah, B., Casati, F., Toumani, F., Ponge, J., Motahari Nezhad, H.R.: Service mosaic: A model-driven framework for web services life-cycle management. IEEE Internet Computing **10**(4) (2006) 55–63

Sustainable Information Systems: a knowledge perspective

Laura Măruşter, Niels R. Faber, and Kristian Peters

University of Groningen, Faculty of Economics and Business,
PO Box 800, 9700 AV Groningen, the Netherlands
{l.maruster,n.r.faber,k.peters}@rug.nl

Abstract. We propose a reorientation of the way the concept of sustainability is dealt with in relation to information systems, positioning the processing of knowledge at the centre of the concept. The concept of Sustainability of Knowledge (SoK), referring to processes that govern knowledge is employed to define Sustainable Information Systems (SIS). Three knowledge aspects are found to be relevant for the design of Sustainable Information Systems: adaptability, offloading and knowledge evaluation. The proposed sustainability approach is translated into requirements needed for SIS, by employing a SOA architecture.

Key words: sustainability, knowledge management, adaptation, offloading, knowledge evaluation, stakeholders, SOA

1 Introduction

Current literature acknowledges that sustainability is a broad, complex concept [1], involving environmental as well as social issues, and which requires continuous learning in order to be understood and tackled. However, the issue of sustainability is still mostly connected to ecological and environmental terms.

The discussion about sustainability and Information Systems (ISs) appears in different contexts. For instance, models and tools have been developed to assess corporate sustainability [2] and sustainability of Management Information Systems [3]. Often, the notion of sustainability of ISs stems from the broader notion of Sustainable Development and is applied in a specific domain. Also, contributions concerning sustainability and Information Systems originate from joining domains, such as sustainability and systems [4].

This article proposes a reorientation of the way the concept of sustainability is dealt with, positioning human behaviour and the processing of knowledge at the centre of the concept. The sustainability approach presented in this paper is then translated into requirements needed for designing an Sustainable Information System (SIS).

Section 2 sets out our position from a social perspective, where we use the notions of knowledge, adaptability and offloading. Section 3 discusses the requirements of an IS that conforms with our notion of sustainability, centred around knowledge aspects. In section 4 we present our conclusions and further research.

2 The social perspective on sustainability

The relation between human behaviour and issues of sustainability is conceptualised using the notion of artificial system [5]. An artificial system is defined as a system that is (i) made by humans and (ii) is operated by humans [1]. Therefore, an IS is treated as an artificial system, because (i) IS's are human-made, and (ii) IS's are operated by humans. From the definition of the artificial system and given that human actions follow from an individual's knowledge, knowledge is identified to control the artificial system.

The first notion used to approach sustainability is *knowledge*. Three related terms are placed in the sequence data - information - knowledge. Data concerns the signals that humans receive using their senses. One level higher, data is used to form information, which concerns the interpretation of data. Finally, knowledge is interpreted information, which enables humans to apply the information in reasoning, decision-making, or performing actions. Knowledge used to operate an IS needs to be updated continuously. Individuals, who control the IS, have to cope with the changes of the system to maintain an equilibrium between the system and its environment. We call this Sustainability of Knowledge (SoK), which means that all knowledge processes need to be guided to lead to the development of new knowledge. Three criteria need to be met by ISs, in order to establish an appropriate balance of *all* knowledge processes leading to SoK. An IS should (a) allow the creation of knowledge, (b) enable the critical evaluation of knowledge, and (c) ensure the effective integration and application of knowledge.

The second notion used for sustainability is *adaptation*. Adaptation means that an organization needs to ensure that its interactions with its environment fit the demands and possibilities of this environment. In some way, the organization's functions need to be aligned with in- and outputs that the environment provides or allows. Organizational functions are realized by humans in processes and tasks, thereby supported by machines and all sorts of information systems. An alignment of organizational functions implies the alteration, reorganization, and redistribution of the organizations processes and tasks.

The third notion used is *offloading*. Offloading involves burdening, harming, destroying or exploiting the economic, ecological and/or social aspects of the environment [1]. When a unbalance exists between the environment and an organisation and its supporting IS's, sustainability can be reached by attempting to achieve a reduction in offloading, by involving stakeholders. Stakeholders can be involved in the organizational sense-making, strategy-forming, and decision-making processes in order to answer the questions how they suffer from the firm's offloading and to what extend. Stakeholders are "those groups and individuals who can affect, or are affected by the achievement of an organization's purpose" [6]. Regarding IS's within organizations, two groups of stakeholders are identified: (a) stakeholders related to the business system (employees, legislators), and (b) IS stakeholders, consisting for instance IS-developers or programmers. The employees of the organization who use the IS also are considered part of the IS stakeholders.

3 Requirements for designing an SIS

Summing up, a *Sustainable Information System (SIS)* is an Information System which (i) adapts to its environment, (ii) involves relevant stakeholders, and (iii) supports the knowledge lifecyle, i.e. knowledge creation, knowledge evaluation and knowledge integration/application.

We illustrate SIS design requirements by using a platform that seems suitable to support our approach, namely Service Oriented-based Architecture [7]. Figure 1 shows our SOA-based solution, consisting of four separate layers: the Business Process layer, the Application layer, the Service layer and what we call the *Knowledge layer*. The first three layers are the standard layers of a SOA-based architecture. The rational of the additional *Knowledge layer* is to address specific knowledge aspects. Concerning *adaptability*, SIS (i) should be equipped with mechanisms that detect and deal with changes occurring between IS and its environment, (ii) the IS forms a suitable platform for dealing with changes. The theory of Complex Adaptive Systems (CAS) seems to provide a suitable framework for dealing with change [8]. CAS theory consider that systems (for instance organizations made up of human and software agents) self-organize and adapt to their changing environment. *Offloading* can be dealt with through additional services that manage stakeholder inputs. Such services can be an error-reporting service, which enables stakeholders to report errors they encounter while using the information system, or a survey service that regularly questions the stakeholders about the functionality and useability that is offered by the information system.

Figure 1 shows our SOA-based solution to support the proposed SIS approach. Concerning adaptability and offloading concerns, the SOA paradigm is an option for addressing adaptation and offloading problems, because it provides solutions for enterprise-wide loose coupling, support for service-oriented business modelling, organisational agility and layers of abstraction [7].

The architecture proposed in figure 1 provides several advantages concerning adaptation and offloading. First, it allows the decoupling between different layers: changes occurring in a certain layer are easily mastered within that layer, by means of the Orchestration Service layer. Second, by incorporating in the Knowledge layer current organizational knowledge, it allows the detection of the discrepancies between the (knowledge about) environment and information system, and enable the solution search.

4 Conclusions and further research

We propose a reorientation of the way the concept of sustainability is dealt with, positioning knowledge issues at the centre of the concept. The notion of sustainability in relation to knowledge is employed to define SIS: Sustainability of Knowledge (SoK), which refers to processes governing knowledge. Three knowledge aspects are relevant regarding SISs: adaptability, offloading and knowledge lifecycle. We translate these aspects into requirements needed for designing a SIS,

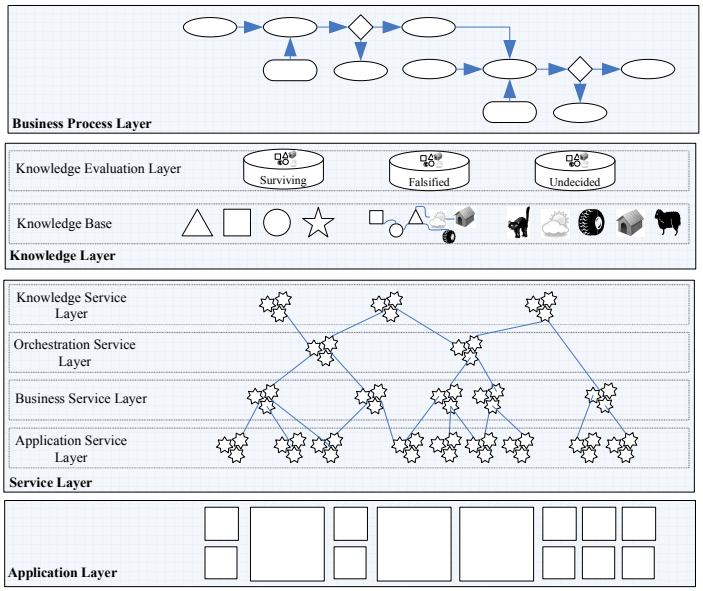


Fig. 1. SIS proposed architecture

by employing a Service Oriented-based Architecture. An additional Knowledge layer is added, consisting of a Knowledge Base and a Knowledge Evaluation layer. The function of this Knowledge layer is to support all three knowledge aspects relevant to sustainability. As further research, we aim to perform case studies in different organizations that aim to test the proposed approach.

References

1. Faber, N.R., Jorna, R.J., van Engelen, J.M.: The sustainability of "sustainability": a study into the conceptual foundations of the notion of "sustainability". Journal of Environmental Assessment Policy and Management, 7(1), 1–33 (2005)
2. Zwetsloot, G.I., van Marrewijk, M.N.: From Quality to Sustanability. J Bus Ethics, 55, 79–82 (2004)
3. Caldelli, A., Parmigiani, M.L.: Management information system - A tool for corporate sustainability. J Bus Ethics, 55, 159–171 (2004)
4. Drinan, J. In Frank A. Stowell, Ray L. Ison, e.a., ed.: Systems for Sustainability. People, Organizations and Environments. Plenum Press (1997)
5. Simon, H.A.: The sciences of the artificial. The MIT Press, Cambridge, Massachusetts (1969)
6. Amaeshi, K.M., Crane, A.: Stakeholder engagement: A mechanism for sustainable aviation. C.S.R. & E.M. 13, 245–260 (2006)
7. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR (2005)
8. Holland, J.: Hidden Order: How Adaptation Builds Complexity. Reading, MA: Perseus Books (1995)

COMA: A Tool for Collaborative Modeling

Peter Rittgen

University College of Borås, 501 90 Borås, Sweden
peter.rittgen@hb.se

Abstract. Building on earlier empirical work we have designed a prototype that supports modeling in groups. The COllaborative Modeling Architecture tool (COMA tool) coordinates UML modeling in groups in the form of a negotiated creation process. We have employed the tool in two case studies.

Keywords: Group modeling, model negotiation, collaboration support

1 Introduction

The nature of modeling as a collaborative process is widely accepted. Nevertheless, most of the tools that support modeling are single-user tools. This is even true for tools that explicitly address group modeling (e.g. Compendium [1]). Some notable exceptions such as [2] are out of date or do not address consensus building [3, 4]. Our objective is to support information synthesis and negotiation as two of the cornerstones of collaborative modeling [2]. The tool and the architecture are the result of a study of modeling behavior [5] and they have been tested in two case studies. Details on these cases are currently under review for publication.

2 Architecture of a Collaborative Modeling Support System

[2] identifies the cornerstones of collaborative modeling as information gathering, synthesis of information and negotiation. According to [6] the primary medium for information gathering is natural language and the organizational form is often that of a chauffeured session [7]. Tools for this already exist [1]. Information synthesis alone is also supported by a large amount of tools, namely by most conventional diagramming, modeling or CASE tools. But there is so far no current tool addressing the negotiation of models. The COMA tool provides this functionality while also allowing for information synthesis. For the latter we have made use of an existing UML modeling tool (UML Pad).

Distributed model negotiation means the coordination of the efforts of a number of modelers. The results from the empirical study suggest that such a system must provide the following functions: Propose, support, challenge and accept. A *proposal* is a suggestion for the revision of the current version of the model. It implies that the modeler posts the content of the local model editor to the group. In building the local

or personal version of the model the modeler can make use of bits and pieces of existing versions (i.e. group model or other proposals), or even copy a whole version and apply changes to it.

A *support* is a positive assessment of a proposal. It can be logged by any team member after reviewing the respective proposal. It can be complemented by a comment that provides a rationale for the decision and perhaps includes suggestions for minor changes.

A *challenge* is a negative assessment of a proposal. It has to be complemented by a justification for the decision as well as constructive comments regarding improvements of the proposal.

COMA offers two rules to decide on the *acceptance* of proposals: A rules of majority and a rule of seniority. When a rule of majority is used, the team operates in an unfacilitated mode where each modeler has a vote of the same weight. Acceptance only depends on the number of supports and challenges. The rule specifies the minimum number of supports required, and the maximum number of challenges allowed for a proposal to be accepted. The required number of supports should be at least two to avoid that a modeler alone (e.g. the proponent) can make the decision. A maximum number of challenges of 0 would force a unanimous decision. When a rule of seniority is applied, the team has a facilitator that makes the decision. Other group members cannot directly influence the decision, but they can do so indirectly by making suitable comments (i.e., supports and challenges). The facilitator can and should consider the supports and challenges in the decision.

3 The COMA Tool

The COMA tool is divided into three working panes (see Fig. 1). The upper one shows the current version of the group model and serves as a point of reference, e.g. for copying and pasting stable parts of the model for building a new version locally. The contents of this pane cannot be edited, hence the grey background. The lower left pane is the editor window where a user can draw the own diagram, possibly with the help of parts that have been copied from the group model or proposals by others. If the user considers the own local model finished she can save it and make it a proposal by right-clicking on the background and choosing "Propose model" from the context menu. This makes it available for others to load into their proposal panes.

The lower right pane represents the said proposal pane. Here the user can load one of the proposals made by the other group members or even the own proposal. A right-click on the background reveals a context menu that allows for logging a support or challenge for this proposal. In the same menu the user can also request a negotiation window that will pop up and display details on the status of the negotiation. These details include the lists of supports and challenges where each entry contains the name of the supporter/challenger and the rationale, i.e. the reason for the decision, and in the case of a challenge also suggestions for improvement.

If sufficient support for a proposal is available, the negotiation window can also be used to accept the proposal. This turns the proposal into the new version of the group model and starts a fresh modeling round. This implies that all the other proposals are

deleted. The proponents of the rejected proposals can resubmit them in the new round, possibly after applying some changes.

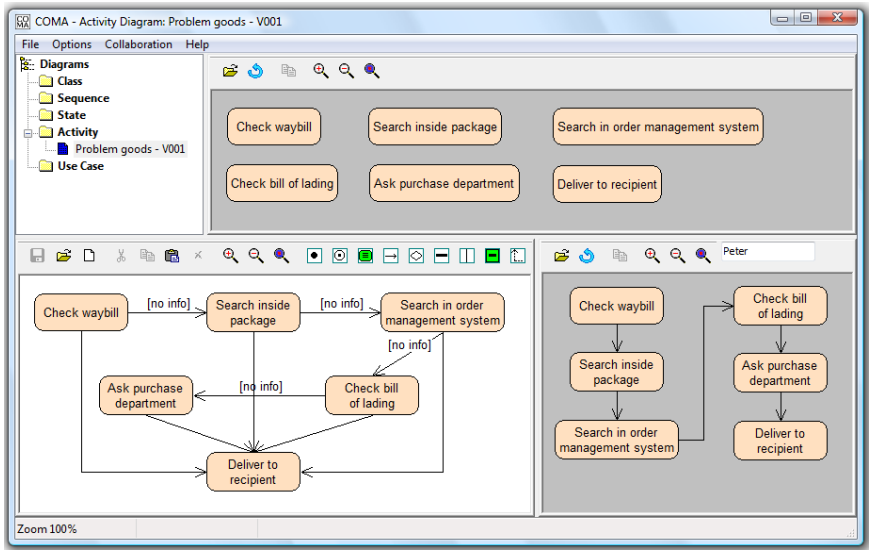


Fig. 1. Screenshot of the COMA tool

Fig. 1 shows a snapshot of the modeling process at a certain stage. This is supposed to give the reader an example of how modeling in COMA proceeds. The group was in charge of developing a model for the handling of so-called problem goods, i.e. goods with an unclear recipient. In a first step they simply wrote down all the activities that are involved thus arriving at the first version V001 (upper pane).

One member, Peter, knows from experience that the activities are performed in a certain sequence. He draws the respective diagram by copying all elements from the upper pane and simply adding the arrows and rearranging the objects. He proposes this diagram and thereby makes it accessible to the other group members who can now comment on it or also suggest their own versions.

Jenny, the group member from whom the screenshot in Fig. 1 was taken, decides to load Peter's proposal in her proposal pane (the lower right one). She takes a closer look at it and agrees with the principle sequence but she is quite sure that the search for the recipient is terminated as soon as the recipient is identified and that further steps are skipped. She draws the respective diagram in her local editor window (lower left pane) and makes a counter-proposal.

When comparing the two competing proposals the other group members decide that Jenny's proposal is more in line with the actual procedure and they log respective supports for her proposal. The new proposal was subsequently adopted by the group as version two.

Although not a business modeling language, we have chosen the UML as the basis for the COMA tool. This decision was driven by a number of factors. Firstly, the UML is a standardized language with considerable impact in the information systems

industry. Secondly, some of the diagrams, e.g. Use Case and Activity Diagrams, are often used for business process modeling as companies want to leverage the benefits of a common language for both business analysis and IT design. Another reason is the ready availability of open-source modeling tools that reduce the investments in tool development.

The tool is implemented in Visual C++ 2005 on Windows based on the UML Pad by Luigi Bignami (bignamil@tiscali.it) and with the wxWidgets GUI library (http://www.wxwidgets.org/). It is available for download at http://www.COMA.nu.

References

1. Conklin, J., Selvin, A., Buckingham Shum, S., Sierhuis, M.: Facilitated Hypertext for Collective Sensemaking: 15 Years on from gIBIS. In: Weigand, H., Goldkuhl, G., de Moor, A. (eds.): Proceedings of the 8th International Working Conference on the Language-Action Perspective on Communication Modeling (LAP'03), Tilburg, The Netherlands (2003)
2. Dean, D., Orwig, R., Lee, J., Vogel, D.: Modeling with a group modeling tool: group support, model quality, and validation. Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences. Vol.IV: Information Systems: Collaboration Technology Organizational Systems and Technology, 4-7 Jan 1994, Vol. 4. IEEE Computer Society Press, Los Alamitos, CA (1994) 214-223
3. Pereira Meire, A., Borges, M.R.S., Araújo, R.M.d.: Supporting multiple viewpoints in collaborative graphical editing. Multimedia Tools and Applications **32** (2007) 185 - 208
4. Santoro, F.M., Borges, M.R.S., Pino, J.A.: CEPE: Cooperative Editor for Processes Elicitation. Proceedings of the 33rd Hawaii International Conference on System Sciences - Volume 1. IEEE Computer Society (2000)
5. Rittgen, P.: Negotiating Models. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.): Advanced Information Systems Engineering, 19th International Conference, CAiSE 2007, Trondheim, Norway, June 2007, Proceedings. Springer, Berlin (2007) 561-573
6. Frederiks, P.J.M., Weide, T.P.v.d.: Information Modeling: the process and the required competencies of its participants. Data & Knowledge Engineering **58** (2006) 4-20
7. Dean, D.L., Orwig, R.E., Vogel, D.R.: Facilitation Methods for Collaborative Modeling Tools. Group Decision and Negotiation **9** (2000) 109-127

AutoModelGen: A Generic Data Level Implementation of ModelGen

Andrew Smith and Peter McBrien

Dept. of Computing, Imperial College London,
Exhibition Road, London SW7 2AZ

Abstract. The model management operator ModelGen translates a schema expressed in one modelling language into an equivalent schema expressed in another modelling language, and in addition produces a mapping between those two schemas. **AutoModelGen** is a generic *data level* implementation of ModelGen that meets these desiderata. Our approach is distinctive in that (i) it takes a generic approach that can be applied to any modelling language, and (ii) it does not rely on knowing the modelling language in which the source schema is expressed in.

Key words: ModelGen, Model Management, Data Transformation, Data Integration, Meta Modelling

1 Introduction

ModelGen is a model management [1] operator that translates a schema in a source **data modelling language** (**DML**), for example XML Schema, into an equivalent schema in a target DML, for example SQL, and also generates a mapping between the two schemas. To date, no implementation of ModelGen generates both a target schema and a mapping between the source and target schemas [2]. In this demonstration we present an implementation of ModelGen that automatically creates a data level mapping that describes how *instances* of the source schema should be translated [3]. Further distinguishing features of our approach are that (1) the translations are made on a **Universal Meta Model** (**UMM**) that has previously been shown to able to represent schemas from a large number of data modelling languages, and (2) the mappings created are bidirectional *i.e.* we also create a mapping from the target to the source schema.

Fig. 1 gives an overview of our approach. In step (1) the source schema S_s is transformed into an equivalent schema, S_{hdm-s} expressed in the UMM. In step (2), a series of information preserving [4] transformations are applied to S_{hdm-s} to transform it into S_{hdm-t}, that matches the structure of a schema in the target DML. In step (3) the constructs in S_{hdm-t} are transformed into their equivalents in the target DML to create S_t. We will first discuss the overall architecture of our system and then discuss the details of the two algorithms used in step (2).

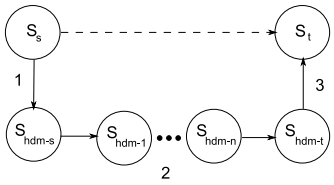


Fig. 1. Overview of the approach taken

2 Architecture

AutoModelGen is a tool that creates schemas and **both-as-view** (**BAV**) transformations [5] in the AutoMed data integration system [6]. AutoMed allows for schemas to be stored in both the native modelling language of a data source (*eg* XML or SQL/relational) and in AutoMed's UMM called the **hypergraph data model** (**HDM**) [7].

The HDM uses three modellings constructs (nodes, edges and constraints) to represent the constructs of a high level DML [8]. HDM nodes and edges have associated data values (called their **extent**). Constraints place restrictions on the data values that may appear in the extent. Each variant of a high level DML construct has a particular representation in the HDM. For example, the set of HDM constraints generated by a nullable SQL column will be different those generated by a not null column. This is important when it comes to identifying whether a group of HDM constructs matches a particular construct in the target DML.

A BAV information preserving [4] mapping is made up of a sequence of transformations called a **pathway**, where each transformation either adds, deletes or renames a single **schema object** (such as a single SQL column, SQL primary key definition, XML element, *etc*), thereby incrementally generating a new schema from an old schema. The extent of the schema object being added or deleted is defined as a query on the extents of the existing schema objects.

BAV transformations can be grouped into information preserving **composite transformations** (**CT**), that act as templates of a fragment of a pathway, describing common patterns of transformation steps. For example the CT id_node_expand is useful when the target DML has explicit keys (such as a key attribute in ER or SQL models) but the source model has implicit keys (such as in XML Schema).

3 Algorithms

AutoMatch inspects a given HDM schema S_{hdm-x} and determines which of the nodes, edges and constraints match a construct in the target DML.
AutoTransform searches for a schema in which all the HDM schema objects match the structure of the target DML by repeatedly applying CTs to the schema objects in S_{hdm-x} that **AutoMatch** identifies as not matching constructs in the

target DML. The set of possible schemas created in this way is called the **world space** [9] of the problem. It can be represented as a graph whose nodes are individual HDM schemas and whose edges are the CTs needed to get from one node in the graph to the next. To limit the number of possible CTs that have to be performed at each node of the world space graph, CTs must satisfy certain preconditions before they can be executed. In our algorithm the preconditions rely on the structure of the HDM schema, in particular the constraints, surrounding the schema object that the CT is to be applied to.

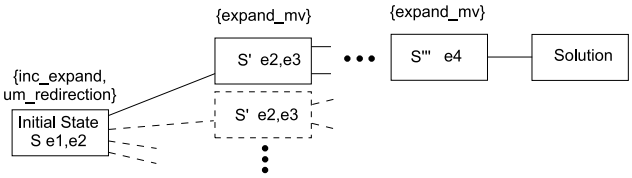


Fig. 2. An example world space graph

The world space graph for the example in the demonstration is shown in Fig. 2. Each node is labelled with a schema name ($\mathsf{S}, \mathsf{S}', \ldots$) and a list of the unidentified schema objects e1,e2,... in that schema, or the word Solution. All the constructs in a Solution node match those of the target model. Above each node is a list of CTs that meet the preconditions for the unidentified schema objects in that schema. Those CTs that meet the preconditions most closely are sorted to the top of the list and executed first. **AutoTransform** performs a depth first search on the world space graph starting from the initial state, by executing the CT at the head of the list, until a solution or a dead end is reached. If a dead end is reached the algorithm back tracks to the last node in the world space graph where an untried CT/edge combination exists, and executes the next CT in the list for that node. If all the edges on all the nodes have been tried without finding Solution then **AutoTransform** has failed.

4 Execution of the Tool

The current prototype of the tool is capable of translating between schemas represented in the XML, ER and SQL DMLs, and of materialising the data instances of a schema in one DML as instances of a second DML. In a typical execution of the tool the following steps are performed:

1. A Source schema is imported into AutoMed, and then translated into the HDM.
2. The **AutoMatch** and **AutoTransform** algorithms are run on the newly generated HDM schema, hdm, to generate a new schema hdm', where the hdm' schema is one that matches the structure of the target DML.

3. The hdm' schema is translated into a schema in target DML.
4. This target schema along with its data is then materialised.

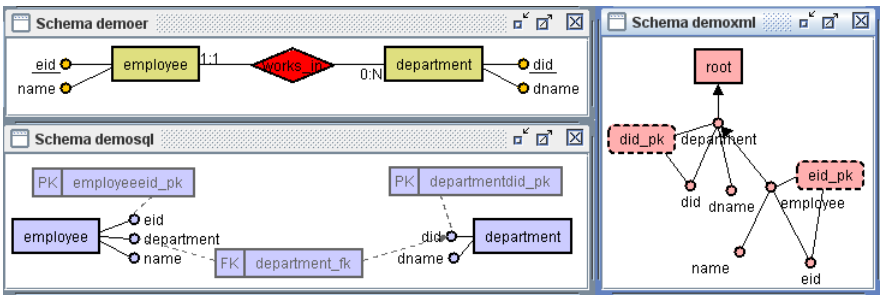


Fig. 3. Equivalent ER, SQL and XML Schemas created by **AutoModelGen**

Since the result of the tool's output is a set of schemas and BAV mappings held in AutoMed, the standard AutoMedtoolkit may be used to view results of the tool's execution. Fig. 3 shows a screen shot from the AutoMed GUI featuring an ER schema and then in an anti-clockwise direction, the SQL and XML equivalents of the schema generated automatically by **AutoModelGen**.

References

1. Bernstein, P.A., Halevy, A.Y., Pottinger, R.: A vision of management of complex models. SIGMOD Record **29**(4) (2000) 55–63
2. Bernstein, P.A., Melnik, S.: Model management 2.0: manipulating richer mappings. In: SIGMOD Conference. (2007) 1–12
3. Smith, A., McBrien, P.: A generic data level implementation of modelgen. In: BNCOD. (2008) To appear
4. Hull, R.: Relative information capacity of simple relational database schemata. SIAM J. Comput. **15**(3) (1986) 856–886
5. McBrien, P., Poulovassilis, A.: Data integration by bi-directional schema transformation rules. In: ICDE. (2003) 227–238
6. M. Boyd, S. Kittivoravitkul, C. Lazanitis, P.J. McBrien and N. Rizopoulos: AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. In: CAiSE04. Volume 3084 of LNCS., Springer Verlag (2004) 82–97
7. McBrien, P., Poulovassilis, A.: A general formal framework for schema transformation. In: Data and Knowledge Engineering. Volume 28. (1998) 47–71
8. Boyd, M., McBrien, P.: Comparing and transforming between data models via an intermediate hypergraph data model. J. Data Semantics IV (2005) 69–109
9. Weld, D.S.: An introduction to least commitment planning. AI Magazine **15**(4) (1994) 27–61

REDEPEND: Tool support for *i** modelling in large-scale industrial projects

James Lockerbie & Neil Maiden

Centre for HCI Design, City University London, UK
J.Lockerbie@soi.city.ac.uk, N.A.M.Maiden@city.ac.uk

Abstract. This paper describes our REDEPEND tool for *i** modelling and analysis, and its features designed to make it more usable and useful in industrial projects. We present usability features, pattern-based techniques that generate text requirements statements from *i** models, an extension to *i** means-end links using satisfaction arguments, and an approach to integrate *i** with an existing in-house requirements process.

1 Introduction

Whilst the *i** approach [1] has been developed and applied to case studies for some time, it has not been applied widely in industrial requirements projects. This is despite the undoubted strengths of *i**, which include a simple but formal and stable semantics, a graphical modelling notation that is simple to use, models that are amenable to computational analysis, and applicability in both agent-oriented and goal-oriented requirements methods. We believe this lack of industrial uptake is due to: (i) the lack of robust, useful and usable tools for developing and analysing *i** models; (ii) inadequate semantics to express *i** means-end links; and (iii) poor integration with in-house requirements processes. These points form the basis of one of main research objectives – to deliver a usable and useful *i** modelling tool along with new techniques to integrate the *i** approach successfully in large-scale industrial requirements projects.

The remainder of this paper presents our *i** modelling tool, REDEPEND, and a summary of our research results over the past 6 years.

2 The REDEPEND *i** Modelling Tool

The REDEPEND tool provides systems engineers with *i** modelling and analysis functions, coupled with additional functionality and reliability of Microsoft Visio. It provides a graphical palette from which systems engineers can drag-and-drop *i** concepts to develop Strategic Dependency (SD) and Rationale (SR) models. REDEPEND also provides systems engineers with simple functions to assist modelling, such as providing a verification alert if a model change violates *i** modelling constraints.

We have successfully applied *i** and REDEPEND to model requirements for major air traffic management systems [2, 3, 4]. These projects have provided valuable recommendations from analysts to enable us to improve the usability of REDEPEND, and to extend its features to better integrate *i** with in-house requirements processes. We believe that these new features and approaches are essential to the successful uptake of *i** in industrial projects. The following sections describe these features.

3 Usability Features of REDEPEND

REDEPEND has been developed to contain features that make it more usable when handling larger *i** models. For example, the user can link corresponding actors in related models to facilitate simple navigation and rapid access to different parts of a modelled system according to viewpoints – commonly between SD and SR models. This link feature also provides the user with change synchronization options to enable model consistency, e.g. changing an actor name in one model can be propagated through to another. REDEPEND also provides colour-coding and check features to highlight and shade-out model elements using layers, to partition and mark up models during analysis and review tasks. Some of these features can be seen in Figure 1(a).

4 Productivity Features of REDEPEND

To make REDEPEND more useful to requirements analysts we designed simple patterns – recurring syntactic and semantic structures in *i**– that are applied automatically to any SD model expressed in REDEPEND to generate textual requirement statements. Each pattern defines one or more desired properties (requirements) on the future system that must be satisfied for the SD model dependency to hold for the future system. As such, the SD model, which has been signed off as complete and correct, informs further discovery and specification of requirements statements. The concepts and patterns underlying this approach are described at length in [4].

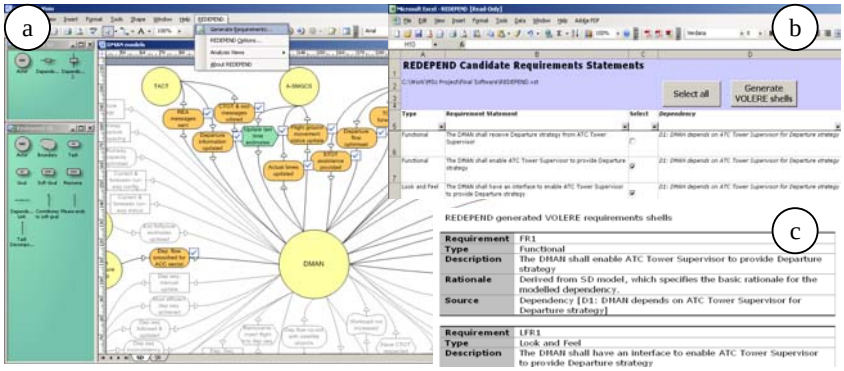


Figure 1 The three stages of requirements generation in REDEPEND

Figure 1 demonstrates how REDEPEND generates requirements from an analyst's perspective: (a) the analyst accesses the requirements generation function from the REDEPEND pull-down menu. (b) REDEPEND delivers the candidate requirements into tailored MS Excel sheets. The analyst can tick and un-tick selected requirement statements prior to generating structured VOLERE [5] shells in MS Word, as depicted in (c). Each requirement in the document is structured using and expressed with a partially complete VOLERE shell specifying: a unique id; the requirement type; the requirement description; a rationale of canned text describing how the requirement was generated; and the source dependency in the SD model.

5 Requirements Process Integration Using REDEPEND

We extended REDEPEND to support the specification of satisfaction arguments from REVEAL [6] to provide additional information to *i** means-end links by associating them with important properties of the problem domain. REDEPEND automatically generates a new satisfaction argument sheet for the selected goal or soft goal as shown in Figure 2a. The selected goal or soft goal is the default end-element, and each element that is a means to the goal or soft goal is a means-element. Means-elements from within the same actor boundary as the end-element are documented in the internal tab, with the external tab displaying means-elements from other actors. Changes made by the analyst to the satisfaction argument sheet and model are propagated automatically to both, thus keeping each model and its arguments consistent. The analyst manually completes each satisfaction argument by selecting an existing property from the database or adding a new one to it. The domain properties are stored in a global database (see Figure 2b) associated with the SR model to ensure effective reuse of properties that, we believe, can improve the specification of satisfaction arguments.

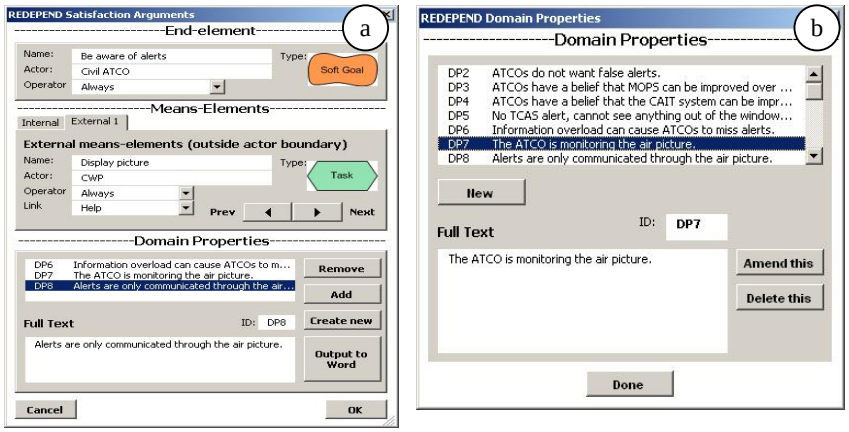


Figure 2 Satisfaction arguments (a) and domain properties management (b) in REDEPEND

We also implemented a procedure to analyse the impact of software requirements on system-wide goals and soft goals using an embedded functional requirement-SR matrix, as shown in Figure 3. An analyst copies functional requirements into the left column, then REDEPEND automatically generates the other columns with tasks and resources from the selected software actor in the SR model. The analyst then completes the matrix by adding a simple + or – to indicate whether the task or resource is enhanced or detracted by the functional requirement. The analyst can then use the tool to propagate this mapping through to goals and soft goals in the model. To aid this task REDEPEND supports 2-way navigation between elements in the SR model and the matrix. We consider such model navigation is essential to support the analysis of large systems. Full details of this work can be found in [3]

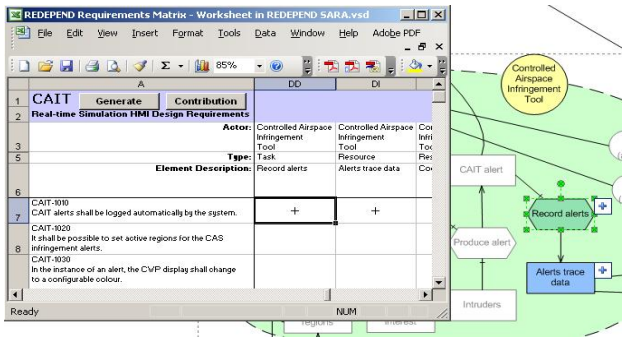


Figure 3 Mapping software requirements from the matrix to *i** model elements in REDEPEND

6 Future Research

We continue our research to develop and refine the use, and hence effectiveness, of REDEPEND through its application in industrial projects. Work is already under way in the following areas:

- Incorporating the SR model in the pattern-based requirements generation facility.
- Automatically generating a first cut SR model from a complete SD model.
- Exploration into the new functionality of Visio 2007, such as improved external data linking and presentation capabilities.
- Collapsing actor boundaries for improved model scalability.

7 References

1 Yu E. & Mylopoulos J.M., 1994, 'Understanding "Why" in Software Process Modelling, Analysis and Design', Proceedings, 16th International Conference on Software Engineering, IEEE Computer Society Press, 159-168.

2 Maiden N.A.M.,, Jones S.V., Manning S., Greenwood J. & Renou L., 2004, 'Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study', Proceedings CaiSE'2004, Springer-Verlag LNCS 3084, 368-383

3 Maiden N.A.M., Lockerbie J., Randall D., Jones S. & Bush D., 2007, 'Using Satisfaction Arguments to Enhance *i** Modelling of an Air Traffic Management System', Proceedings 15th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 49-52.

4 Ncube C., Lockerbie J. & Maiden N.A.M., 2007, 'Automatically Generating Requirements from *i** Models: A Case Study with a Complex Airport Operations System', Proceedings 13th International Working Conference, REFSQ'2007, Trondheim Norway, Springer-Verlag Lecture Notes on Computer Science LNCS 4542, 33-47.

5 Volere Requirements Resources, http://www.volere.co.uk/template.htm

6 Hammond J., Rawlings R. & Hall A., 2001, 'Will It Work?', Proceedings 5th IEEE International Symposium on Requirements Engineering, IEEE Computer Society, 102-109.

Supporting Participatory Requirement Engineering in an ERP Software Community

Harris Wu[1]

[1] Old Dominion University
Norfolk, VA, 23529
hwu@odu.edu

Abstract. The paper presents a prototype[1] aiming to widen user participation in requirement engineering for an open source ERP software company. The prototype utilizes a set of templates in a wiki system to allow community participation, and yet maintain necessary processes and models in the software development. The prototype informs developers about the ongoing work of others, in part to avoid conflicts. An XML-based infrastructure is being developed to improve interoperability with other tools such as discussion forums, issue tracking and documentation systems, to further utilize community efforts in software development.

Keywords: requirement engineering, participatory, community, collaboration.

1 Introduction

Participatory and collaborative software engineering has drawn much attention in recent years [1]. In fact, requirement engineering is participatory by its nature. Requirements engineering consists of the cohesive collection of all tasks performed by various stakeholders related to the identification, analysis, specification, and management of requirements [2]. Requirements engineering is the key bridge between users, system analysts, developers and other stakeholders of a software system. As software becomes the core of business processes, and software development becomes an integral part of the global economy, the roles of users, system analysts and other stakeholders are continuously blended.

With the increasing use of packaged software and industry acceptance of inter-organizational collaboration, the "participatory" need of requirement engineering has to be stressed. Compared to proprietary systems, off-the-shelf software packages have much larger stakeholder communities. Information sharing, reuse and web collaboration have gained industry acceptance and even become part of the industry culture, as evident from active user groups and discussion forums in many commercial software communities and the success of open source software. However, the existing collaboration tools for requirement engineering do not always scale to support larger stakeholder communities. The rapid advance of technology (such as

[1] Supported in part by the United States National Science Foundation, Award No. 0725277.

Unified Modeling Lanugage) and the deployment cost of software tools often raise the barrier of entry for participation.

This paper presents a prototype developed to support participatory requirement engineering for an open source ERP software company. The prototype aims to widen and utilize community participation. In the following sections, I first review a research roadmap for collaboration in software engineering, which will be used as a framework in subsequent discussion. Then I describe the software company and its needs for collaboration tools when trying to widen community participation in requirement engineering. I will present the prototype and preliminary results. The paper is concluded with a discussion of the next steps.

2 A Roadmap for Collaboration in Software Engineering

In a seminal roadmap [1] outlined at the recent "Future of Software Engineering" conference, tools developed specifically to support collaboration in software engineering fall into four broad categories: *model-based, process oriented, awareness support, and infrastructure*. Table 1 provides a summary of this roadmap.

Table 1. Four categories of collaboration tools for software engineering

Tools	Support
Model-based	Allow engineers to collaborate in the context of a specific representation of the software, such as a UML diagram.
Process support	Represent all or part of a software development process. Systems using explicit process representations permit software process modeling and enactment. In contrast, tools using an implicit representation of software process embed a specific tool-centric work process, such as the check-out, edit, or check-in process of most SCM tools.
Awareness	Inform developers about the ongoing work of others, in part to avoid conflicts.
Infrastructure	Improve interoperability among collaboration tools, and focuses primarily on their data and control integration

3 Requirement Engineering Challenges to an Open Source Software Company

xTuple is an Enterprise Software company, author of the open source ERP solution Postbooks (http://sourceforge.net/projects/postbooks). The company gains revenue by selling training and support for Postbooks implementations. xTuple also sells OpenMFG, a manufacturing-enhanced ERP solution with a community code model built on the same open source code base. In 2007 xTuple was a finalist in eWeek Excellence awards and received a 5 star rating from the Channel Web Network (crn.com).

xTuple receives requirements for its software through two online communities. The Postbooks project ranks in the top 10 on sourceforge.net, had over 100,000

downloads in the past six months and has about a dozen regular voluntary contributors. The commercial OpenMFG community (openmfg.com) has around twenty implementation partners and a hundred licensed customers. xTuple product support and development teams discuss requirements with users in both communities mainly through emails and discussion forum threads, often hundreds of them a day. xTuple hopes to utilize community efforts in capturing, reviewing, organizing, prioritizing, negotiating, cleansing and documenting the requirements using Web 2.0 concepts. To support requirement engineering among a large, diverse software community, xTuple has a number of needs summarized in Table 2:

Table 2. xTuple's needs for collaboration tools in requirement engineering

Tools	Needs and Challenges
Model-based	As an enterprise software company, xTuple needs diagramming tools and structured documents to support modeling of complex enterprise business processes. However, the tool should not require advanced technical knowledge or commercial software licenses.
Process support	As a commercial software company with a sizable base of paying customers, xTuple needs a rigorous process to manage how requirements are prioritized, priced, negotiated, approved and implemented. However, the cyclic and the parallel nature of its software development and the blurred role definitions (users versus developers) render most workflow tools too restrictive.
Awareness	With a large number of users involved in creating, modifying or discussing the requirements, xTuple needs to be aware, and also keep its community informed, of the constant requirement changes coming from both open source and commercial software communities. Furthermore, it needs requirement provenance, i.e. to figure out why a certain requirement was added or modified.
Infrastructure	Improve interoperability among collaboration tools, including the existing forum, bug tracker and documentation tools

4 Prototype

The prototype builds upon a number of open source tools (Figure 1) to fulfill the needs in Table 2. It also attempts to keep the current tools used by the community intact. The center component is MoinMoin, a popular wiki system. Besides wysiwyg editing, MoinMoin supports editing of diagrams, section editing and conversion of documents into/from XML. Templates for a growing set of requirement categories are being developed, including new module requests, enhancement requests, and bug fixes. The templates are further refined such as by client/server (user interface or application logic) and functional modules (accounting, customer relationship, etc). Users can create a wiki document by answering a simple questionnaire. Or, users can create a new wiki document through a link from the discussion forum (phpBB), and the discussion will be copied to the new wiki document. Various checks are performed to prevent duplication of the same requirement. Once a requirement document is created, it can be re-cast to other templates. The whole requirement

workflow including requirement creation, negotiation, pricing and approval is carried out through fleshing out of the templates by different stakeholders. Different templates result in different workflows. Awareness is supported by email notifications to stakeholders based on the template. Users can register for notifications by modifying the Notifications section of the document. Users can trace the requirements by inspecting the revisions and various metadata on the wiki document/template. The wiki system is integrated with Joomla!, a popular content management system, which provides single sign-on and access to a mysql database. XML is used for moving content in and out of MoinMoin.

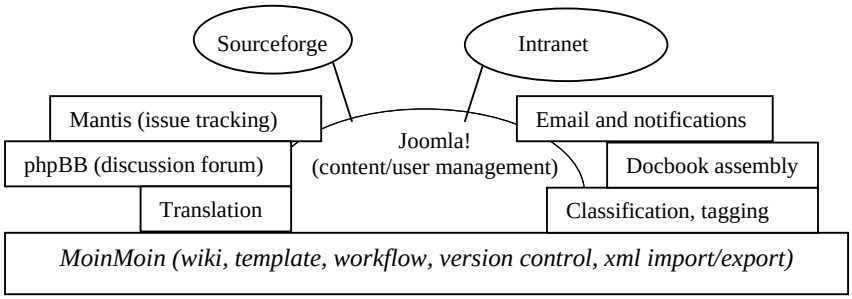


Fig. 1. Overview of the prototype.

5 Preliminary Results and Next Steps

The prototype has learned from the wiki tool in [3]. The wiki approach has gained initial organizational acceptance by xTuple, which is a considerable feat given that several key developers had doubts about using wiki to manage requirements. The effectiveness of the prototype will be assessed by examining the quality and quantity of requirements created, negotiated, cleansed or organized in the system. The prototype is being integrated with several existing tools supporting the xTuple community. The immediate enhancement tasks include classification and tagging for requirement reuse, dynamic translation (using Google), and requirement prioritization (using Mantis).

References

1. Whitehead, J. (2007). Collaboration in Software Engineering: A Roadmap. Future of Software Engineering (FOSE) 2007.
2. IT toolboxwiki, http://wiki.ittoolbox.com/index.php/Requirements_engineering.
3. Decker, B. et al. Wiki-based stakeholder participation in Requirements Engineering, IEEE Software, March/April 2007.

Role and Request Based Conceptual Modeling – A Methodology and a CASE Tool

Yair Wand, Carson Woo, and Ohad Wand

Sauder School of Business,
The University of British Columbia, Vancouver, Canada
yair.wand@ubc.ca, carson.woo@ubc.ca, ohad.wand@gmail.com

Abstract. This paper contains a brief description of the R^2M (Role and Request Modeling) method and its supporting visual modeling CASE (Computer Assisted Software Engineering) tool. R^2M is a modeling method for creating Conceptual Models of work systems using a combination of ontological and object-oriented concepts. Ontological principles serve to define the meaning of modeling constructs in terms of domain semantics, and to derive rules guiding the modeling process. The CASE tool is a graphical software tool that supports the creation of models according to the R^2M method. Guided by the principles of R^2M, the tool helps assure the semantic integrity of models, and enables management of complex models via decomposition (i.e. more details at decreasing abstraction levels). The tool can help ensure consistency between different modelers and completeness of models.

Keywords: conceptual modeling, business analysis, CASE tool

A Conceptual Model - in the context of information systems analysis - can be described in simple terms as a formal representation of the organizational domain for which an information system is being developed. The importance of Conceptual Modeling as a tool in systems analysis and requirements determination has been widely recognized. Four purposes have been identified for conceptual models: supporting an analyst's understanding of an application domain, communicating with stakeholders, communicating with implementers, and documenting system rationale for future needs.

The object-oriented approach is arguably the most common software design and implementation paradigm now in use. This is evidenced by the popularity of UML (the *Unified Modeling Language*)[1]. However, the use of object-concepts in Conceptual Modeling has not been widely adapted. A main reason is that there are no generally accepted semantics of these concepts as conceptual modeling elements.

To address the issue of assigning domain semantics to object-oriented constructs we have used ontological concepts and principles [2,3]. The ontological concepts can be used to define the meaning of object-oriented concepts and the principles can serve to suggest rules to guide ontologically-sound modeling. Specifically, we propose that objects represent active things (actors) in an application domain and object classes represent organizational roles. The dynamics of a modeled domain can then be represented in terms of state changes of individual actors and of interactions between

actors that assume certain roles. This view led us to develop a set of modeling rules which address two issues: first - the mapping of domain phenomena to a model; and second - semantic integrity constraints that can be applied to constructed models. Based on these rules, we developed a modeling procedure that assures the ontological validity of constructed models. The procedure can identify situations where the modeler needs to clarify domain aspects with stakeholders.

The modeling approach – termed *Role and Request Modeling* (R^2M) – has been implemented in a CASE Tool. This tool embeds data structures that reflect the fundamental ontological concepts and principles (that in turn guide the semantic integrity rules). As well, the tool provides checks for the adherence of constructed models to the modeling rules.

R^2M is graphic notation-independent. However, the user interface of the R^2M software (shown in Figure 1 below) uses an intuitive representation of the modeled domain. The information about the model appears in several visible panes:

- The *Role Explorer* (left side) displays all roles in the model for easy navigation.
- The *Modeling Canvas* (main portion) in which the model is created by the user.
- The *Property Details* (lower portion) where details about the role currently selected in the *Modeling Canvas* are displayed and manipulated.
- An additional pane showing errors in the model (the *Semantic Errors* pane, described below) can be visible or hidden.

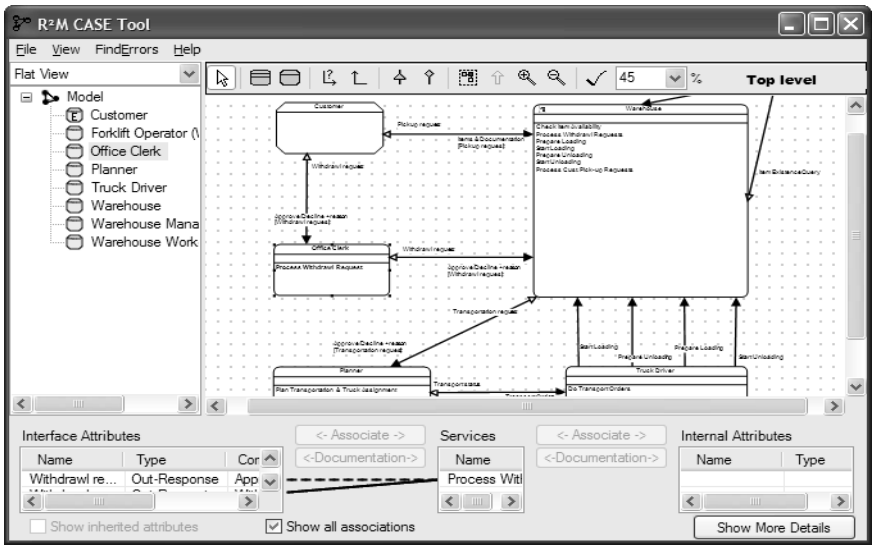


Fig. 1. The *R^2M CASE Tool* user interface

To enable construction of models of a complex environment R^2M supports a well-formalized method of and rules for decomposition. The rules assure that models at any level will be ontologically and syntactically consistent with higher and lower level models of the same domain.

As an example of decomposition using R^{2}M, Figure 2 shows part of a domain model within the *Modeling Canvas*. The view shown is the *top level* model – i.e. the highest level of abstraction. At this level in the example, both the "Customer" and the "Office Clerk" roles communicate with the "Warehouse" role.

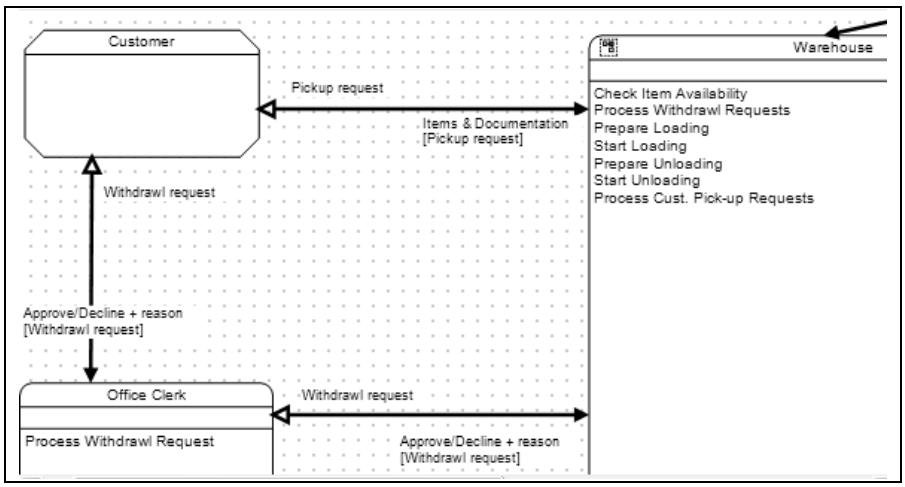


Fig. 2. Sample model – top level

Figure 3 shows part of the decomposition model of the "Warehouse" role of the same domain. This view shows roles and communications that are internal to the "Warehouse" (i.e. the "Warehouse Manager" and "Warehouse Worker" roles, and the communications between them) as well as the communications between these and the roles that – at the *top level* – appear to communicate with the "Warehouse" role. As can be seen in this example, the "Office Clerk" communicates with "Warehouse Manager" and the "Customer" communicates with the "Warehouse Worker". R^{2}M supports decomposition to any level and ensures consistency between the levels.

Figure 4 presents an example of the *Semantic Errors* pane (lower part of the figure). This pane lists all errors present in the model and can be hidden or visible as required. When visible, items on the *Modeling Canvas* relating to the listed errors are highlighted (in red). Clicking on an error brings the item in error into view on the *Modeling Canvas*. As the model is corrected, the *Semantic Errors* pane is automatically updated to reflect the current error state of the model.

We have experimented with the R^{2}M method and tool both in teaching situations and in practical (and realistic size) cases. The results have shown that the use of the method led to consistency of models across modelers. Furthermore, semantic errors identified by the tool were often an indication for the analysts to seek additional information about the modeled domain, thus leading to more complete and accurate models.

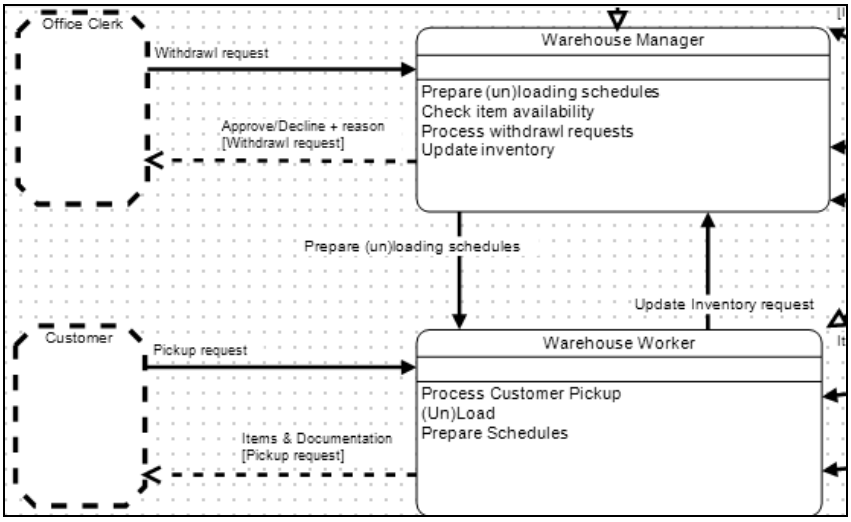


Fig. 3. Sample model – decomposition of the "Warehouse" role

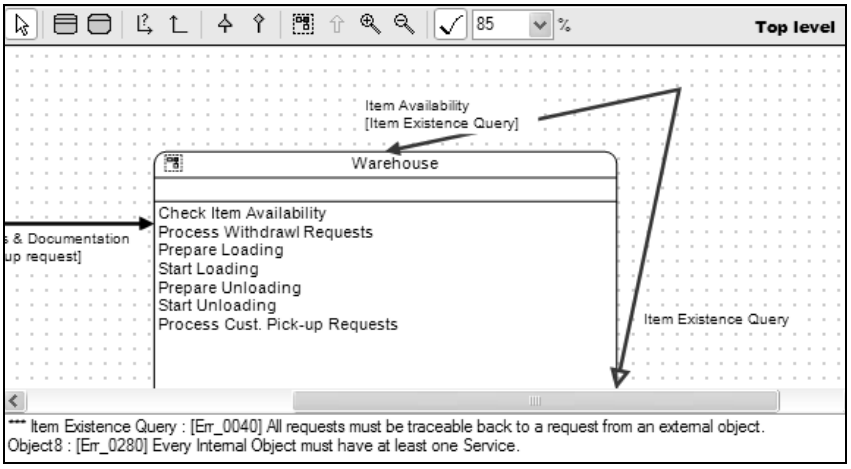


Fig. 4. Semantic checking

References

1. Object Management Group, OMG Unified Modeling Language Specification, Version 2.1.2, http://www.omg.org/spec/UML/2.1.2/
2. Wand, Y., Woo, C.: Object-Oriented Analysis - Is It Really that Simple? In: *Proceedings of the Third Workshop on Information Technologies and Systems* (WITS'93), Orlando, Florida (1993), pp. 186--195.
3. Wand, Y., Woo, C., Hui, S.: Developing Business Models to Support Information System Evolution. In: *Proceedings of the Ninth Workshop on Information Technologies and Systems* (WITS'99), Charlotte, North Carolina (1999), pp. 137—142.

Towards a Taxonomy of Process Flexibility

M.H. Schonenberg, R.S. Mans, N.C. Russell, N.A. Mulyar
and W.M.P. van der Aalst

Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{m.h.schonenberg,r.s.mans,n.c.russell,nmulyar,w.m.p.v.d.aalst}@tue.nl

Abstract. Effective business processes must be able to accommodate changes in the environment in which they operate. Many approaches have been proposed in literature and some of these approaches have been implemented in flexible workflow management systems. However, a comprehensive classification of the various approaches has been missing. In this paper, we take a first step towards a taxonomy of process flexibility by distinguishing four types of process flexibility based on an extensive literature study. An evaluation shows that each of the types can be found in contemporary offerings.
Keywords: taxonomy, flexibility, design, change, deviation, underspecification

1 Introduction

The need for process flexibility has long been recognised [8, 15] in the workflow and process technology communities as a critical quality of effective business processes to adapt to changing business circumstances, e.g., new business strategies. The notion of flexibility is often viewed in terms of the ability of an organisation's processes and supporting technologies to adapt to these changes [7, 16]. Others [12] consider flexibility from the opposite perspective, i.e., they focus on the part of the process which remain unchanged, rather than focusing on which parts have to be changed. Indeed, a process can only be considered to be flexible if it is possible to change it without needing to replace it completely [13].

There have been a series of proposals for classifying flexibility [2, 6, 8, 13], both in terms of the factors which motivate it and the ways in which it can be achieved within business processes. The individual flexibility types discussed in this paper are informed by a multitude of research initiatives in the workflow and BPM fields. Nevertheless, a comprehensive overview of distinct approaches has been missing.

Based on an extensive literature study [9], in Section 2 we identify four distinct types of process flexibility that improve the ability of business processes to respond to changes in their operating environment without necessitating a complete redesign of the underlying process specification. Then in Section 3, we explore the support of the flexibility types and show that each of them exists in contemporary offerings. Finally, the conclusion and future work are presented in Section 4.

2 Flexibility types

First, we present four distinct types of process flexibility and describe how each of the flexibility types operates [9].

Flexibility by design: for handling anticipated changes in the operating environment, where supporting strategies can be defined at design-time.
Flexibility by deviation: for handling occasional unforeseen behaviour, where differences with the expected behaviour are minimal.
Flexibility by underspecification: for handling anticipated changes in the operating environment, where strategies cannot be defined at design-time, because the final strategy is not known in advance or is not generally applicable.
Flexibility by change: either for handling occasional unforeseen behaviour, where differences require process adaptations, or for handling permanent unforeseen behaviour.

Figure 1 illustrates the distinction between the flexibility types in isolation, in terms of the time at which the specific flexibility options need to be configured (1) at design-time, as part of the process definition, or (2) at run-time via facilities in the process execution environment. It also shows the relative completeness of the process definition for each flexibility type at run-time.

Flexibility by underspecification works on the basis of an incomplete process definition. Combined with late binding only, it just offers design-time configuration options, i.e., only the fragments that have been defined during design-time can be selected at run-time. However, when combined with late modelling, also run-time configuration options are offered by providing means to define and select fragments at run-time. In the spectrum of options, *flexibility by design* distinguishes itself by being the flexibility type that is only configurable at design-time, i.e., at design-time the set of possible execution paths is fixed. Hence increasing flexibility corresponds to making this set bigger. Both *flexibility by deviation* and *change* work with complete process definitions. For both types, the configuration options are only available at run-time. Although very similar, only flexibility by change affects the process definition both at instance and type level, whereas flexibility by deviation does not affect the process definition at all, i.e., the model and the reality no longer match after deviation.

3 Evaluation of Contemporary Offerings

To validate the flexibility types, we investigate the degree of support of them in different Process Aware Information Systems (PAISs) [9]. We consider ADEPT1 [14], YAWL[1] (version 8.2b) [1, 4, 5], FLOWer (version 3.0) [3] and DECLARE (version 1.0) [10,11]. The selection of these PAISs has been based on the criterion of supporting process flexibility, which excludes classical workflow systems and

[1] The evaluation of YAWL includes the so-called Worklet Service [4, 5].

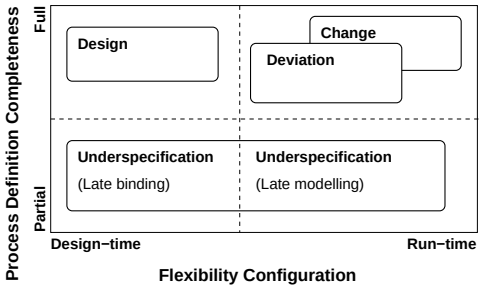


Fig. 1. Flexibility type spectrum

Flexibility by	ADEPT1	YAWL	FLOWer	DECLARE
design	+	+	+	+
deviation	−	−	+	+
underspecification	−	+	−	−
change	+	−	−	+

Fig. 2. Evaluation results

most commercial systems. Moreover, the selected systems cover distinct areas of the PAIS technology spectrum, such as adaptive workflow (ADEPT1), case handling (FLOWer) and declarative workflow (DECLARE).

The evaluation results are summarized in Figure 2, which shows whether the PAIS supports (+) or does not support (–) the respective flexibility type. A detailed evaluation can be found in [9]. None of the evaluated systems provides the full range of flexibility alternatives. Flexibility by design is (to some degree) supported by all offerings. YAWL excels in flexibility by underspecification, ADEPT1 in flexibility by change, FLOWer in flexibility by deviation and DECLARE excels in two areas, namely deviation and change.

4 Conclusion

In this paper we have identified four distinct types of process flexibility. These types are based on a wide variety of proposals for process flexibility found in literature and in practice. The distinction is based on the flexibility configuration (i.e., design-time vs. run-time), the completeness of the process definition and whether it be done without changing the process definition, or not. Our evaluation shows that each of the types exists in contemporary offerings. Interestingly, none of these offerings supports all flexibility types. As future work, we plan to formalise the identified flexibility types in a taxonomy of process flexibility.

References

1. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
2. W.M.P. van der Aalst and S. Jablonski. Dealing with Workflow Change: Identification of Issues and Solutions. *International Journal of Computer Systems, Science, and Engineering*, 15(5):267–276, 2000.
3. W.M.P. van der Aalst, M. Weske, and D. Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering*, 53(2):129–162, 2005.

4. M. Adams, A.H.M. ter Hofstede, W.M.P. van der Aalst, and D. Edmond. Dynamic, Extensible and Context-Aware Exception Handling for Workflows. In F. Curbera, F. Leymann, and M. Weske, editors, *Proceedings of the OTM Conference on Cooperative information Systems (CoopIS 2007)*, volume 4803 of *Lecture Notes in Computer Science*, pages 95–112. Springer-Verlag, Berlin, 2007.
5. M. Adams, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows. In R. Meersman and Z. Tari et al., editors, *Proceeding of the OTM Conference on Cooperative Information Systems (CoopIS 2006)*, volume 4275 of *Lecture Notes in Computer Science*, pages 291–308. Springer-Verlag, Berlin, 2006.
6. S. Carlsen, J. Krogstie, A. Sølvberg, and O.I. Lindland. Evaluating Flexible Workflow Systems. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences (HICSS-30)*, Maui, Hawaii, 1997. IEEE Computer Society Press.
7. F. Daoudi and S. Nurcan. A Benchmarking Framework for Methods to Design Flexible Business Processes. *Software Process Improvement and Practice*, 12:51–63, 2007.
8. P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke. A Comprehensive Approach to Flexibility in Workflow Management Systems. In G. Georgakopoulos, W. Prinz, and A.L. Wolf, editors, *Work Activities Coordination and Collaboration (WACC'99)*, pages 79–88, San Francisco, February 1999. ACM press.
9. M.H. Schonenberg, R.S. Mans, N.C. Russell, N.A. Mulyar, and W.M.P. van der Aalst. Towards a Taxonomy of Process Flexibility (Extended Version). BPM Center Report BPM-07-11, BPMcenter.org, 2007.
10. M Pesic and W.M.P. van der Aalst. A Declarative Approach for Flexible Business Processes Management. In *Business Process Management Workshops*, pages 169–180, 2006.
11. M. Pesic, M. H. Schonenberg, N. Sidorova, and W.M.P. van der Aalst. Constraint-Based Workflow Models: Change Made Easy. In F. Curbera, F. Leymann, and M. Weske, editors, *Proceedings of the OTM Conference on Cooperative information Systems (CoopIS 2007)*, volume 4803 of *Lecture Notes in Computer Science*, pages 77–94. Springer-Verlag, Berlin, 2007.
12. G. Regev, I. Bider, and A. Wegmann. Defining Business Process Flexibility with the Help of Invariants. *Software Process Improvement and Practice*, 12:65–79, 2007.
13. G. Regev, P. Soffer, and R. Schmidt. Taxonomy of Flexibility in Business Processes. In *Proceedings of the 7th Workshop on Business Process Modelling, Development and Support(BPMDS'06)*, 2006. `http://lamswww.epfl.ch/conference/bpmds06/taxbpflex`.
14. M. Reichert, S. Rinderle, and P. Dadam. Adept workflow management system. In W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, editors, *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, volume 2678 of *Lecture Notes in Computer Science*, pages 370–379. Springer, 2003.
15. H.A. Reijers. Workflow Flexibility: The Forlorn Promise. In *15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), 26-28 June 2006, Manchester, United Kingdom*, pages 271–272. IEEE Computer Society, 2006.
16. R.A. Snowdon, B.C. Warboys, R.M. Greenwood, C.P. Holland, P.J. Kawalek, and D.R. Shaw. On the Architecture and Form of Flexible Process Support. *Software Process Improvement and Practice*, 12:21–34, 2007.

A Work Allocation Language with Soft Constraints

Christian Stefansen[1] Sriram Rajamani[1] Parameswaran Seshan[2]

[1] Microsoft Research Lab India, Bangalore, India
{cstef@diku.dk|sriram@microsoft.com}
[2] SETLabs, Infosys Technologies Limited, Bangalore, India
parameswaran_seshan@infosys.com

Abstract. While today's workflow languages have sophisticated constructs for specifying flow of control and data, facilities for associating tasks in a workflow with humans are largely missing. This paper presents SoftAlloc, a workflow allocation language with *soft constraints*, and explains the requirements that lead to its design—in particular what soft constraints are and how they enable workflows to capture best practices and organizational goals without rendering the workflows too strict. SoftAlloc is parameterized over external data access functions and takes a user-database query sublanguage as a parameter, thus allowing SoftAlloc to be used with virtually any process language (such as *WS-BPEL* or the π-calculus based language *SMAWL* [1]).

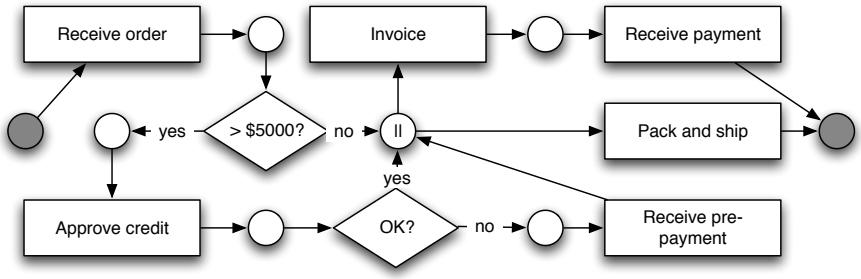
1 Introduction

Computer-orchestrated business processes are increasingly playing a direct role in how companies organize work. Computer-orchestrated processes now commonly involve both human resources and computer resources. Tasks can often be handled by many different resources. Therefore the process orchestration engine must decide in negotiation with the human resources who of the eligible resources ultimately carries out the task. Assigning a task to a resource is referred to as *allocation*. Allocating tasks to humans is inherently more complex than allocating to computer resources: in addition to having multiple, changing attributes that decide what they can, may or should do, humans have personal preferences and may choose to override the allocation rules at runtime, e.g. because they possess domain knowledge not captured in the system or because they make conscious, reflected violations to speed up processing in cases where where the process description focuses too narrowly on perfect compliance. For a treatise of these issues and *soft constraints* see Stefansen and Borch [2].

Consider the example workflow given in Fig. 1. In this workflow we might wish to say that the task *Pack and ship* should be carried out by someone with the role *Packing*, i.e. someone who is assigned to the packing unit. We can imagine specifying this by attaching a rule to the task saying:

```
role = "Packing"
```

Fig. 1 Order process (Petri net-style notation; || is a shorthand for parallel split)



Similarly, we may wish to assign the task *Receive order* to a resource with the role *Sales* who resides in the same locale as what is registered on the order. Additionally, we accept anyone with the role "Global sales". We then write:

```
   role = "Sales" and location = process.order.location
or role = "Global sales"
```

Notice that we compose small building blocks of rules into larger rules. Now imagine that we want the steps *Invoice* and *Receive payment* to be carried out by the same person to retain familiarity. We would then attach a rule saying `role = "Finance"` to the task *Invoice* and then add

```
   role = "Finance"
and user = #whoDid ("Invoice", process.id)
```

as a rule to *Receive payment*.

But something is awry here. While we would certainly *prefer* the task *Receive payment* to be done by the same person who did the invoicing, this is only a *preference*—certainly not a strict rule that should be allowed to stand in the way of timely workflow completion if the designated person happens to be busy or temporarily absent. We have just committed one of the most common mistakes in workflow specification: we have promoted a *soft goal* to a strict rule and thereby created an inflexible system!

Alternatively, we might have removed the rule and only have said `role = "Finance"`, but that would have left out useful intentional information about our best practice. So just specifying fewer constraints is not attractive either.

This example illustrates that allocation constraints can represent a wide spectrum of specifications: some rules are best kept strict (e.g. *Expense approval* must be done by a *Manager*) while other rules are simply guidelines (e.g. *Replenish printer cartridges* should be allocated on a rotation basis (*round robin*)). The latter allocation strategy represents an organizational *soft goal*, which might have been "rotate tedious tasks between qualified workers to achieve a sense of fairness and variation and keep workers happy". This is undeniably a laudable goal, but if the company is experiencing peak load, this goal must temporarily

yield to more mission-critical business goals (e.g. response time *vis-à-vis* our customers). Therefore, it cannot be written as a hard constraint, but leaving it out entirely renders the system unable to suggest the preferred person.

Going back to our example what we probably mean could be written as

```
       role = "Finance"
prefer [10] user = #whoDid("Invoice", process.id)
```

which states that we require a finance person to handle the activity under all circumstances, but we prefer the person who did the invoicing in that process. The number 10 represents a number of *points* to indicate how strong a preference this is. This becomes more interesting, when more preferences are in play. Consider the following rule for allocating the *Credit approval* step in the workflow:

```
       role = "Manager" or role = "Finance"
prefer [10] role = "Manager"
       [-#queueItems(user)]
```

The rule states that either *Manager* or *Finance* should handle the *Credit approval* task. A manager is preferred, but the number of items in the manager's queue is deducted from the preference level; i.e. someone with a short queue is preferred. Indeed, if all managers have more than 10 items in their queues, someone from *Finance* will be preferred in the interest of time. This shows how soft constraints in conjunction with hard constraints can be used to express soft goals and performance heuristics. Other typical soft constraints that we can then model are *round robin*, *prefer least loaded resource*, and *prefer shortest queue*.

Conceptually we can think of soft constraints being attached not just to a tasks, but also to a scope, a set of activities, a process or a resource itself—or they can be inserted on several levels to compose generic rules and policies with more specific ones. Soft constraints can also be used to express overall policies in the workflow engine. In this way the allocation rule language is really a hierarchical *scheduler programming language*.

2 Evaluation/Experience

The prototype has been tested and preliminarily evaluated with Infosys' PEAS platform and the language is slated for inclusion in the PEAS platform. Several prototype workflows have been tested with the allocation language, including a CRM (Customer Relations Management) workflow, a sales process, a approval/review process, and a bank transaction process.

Discussions have been held with domain experts from various industries, such as transportation, finance, insurance, call centers, etc. [3] The language overlaps with the delegating responsibilities of a call center floor manager. It would be a valuable tool to support floor managers in their decisions; if not render them entirely superfluous just yet. The insurance industry (in particular, claims processing) in many cases already has tailored systems, but those typically have a subset of the functionality here, and our language will become increasingly

relevant as companies change to SOA. The transportation industry has different demands, in particular relating to scheduling with time and location, which is future work to add to SoftAlloc. The financial sector focuses mostly on transactions so the language is sufficient, if not slightly more than what is needed.

A patterns-based evaluation in the style of Russell *et al.* [4] was considered, but found unsuitable. It is essential to note that only the patterns that specify who is *ultimately* allowed to perform a task are included; our language did not set out to specify runtime negotiation protocols, which we consider an orthogonal concern and have handled more elegantly elsewhere in the architecture. While the language does indeed cover all the patterns intended, a patterns-based analysis inadequately captures the expressive power of our language. E.g. soft constraints represent an idea that could easily be expanded to comprise an entire suite of patterns in its own right, but the patterns-based analysis did not anticipate soft constraints and therefore does not cover them. Similarly, the patterns work does not mention if patterns can be combined and if so with what constraints.

3 Conclusion and Future Work

The language SoftAlloc can express all patterns for which it was designed and all examples that were deemed necessary. The use of soft constraints has proven extremely beneficial, and as intended the language integrates with any system we have seen. A GUI for the language is being developed in the production setting where the language is to be used. Based on the discussion of resource patterns, it would be interesting to construct a collection of soft constraint patterns.

Some benefits are yet to be reaped: the language is a small non-recursive DSL and this means that allocation rules are not only expressions that can be evaluated, but also pieces of data that can be used to perform static analysis. Performance simulation can be used to identify bottlenecks, estimate capacity requirements, and suggest what resources to add. This is an important improvement over previous systems, where the lack of integration made performance analysis a non-routine job requiring specialized skills.

References

1. Stefansen, C.: SMAWL: A SMAll Workflow Language based on CCS. Technical Report TR-06-05, Harvard University, Div. of Eng. and App. Sci. (March 2005) `http://stefansen.dk/papers/ccs-petrinet.pdf`.
2. Stefansen, C., Borch, S.E.: Using soft constraints to guide users in flexible business process management systems (BPMS). International Journal of Business Process Integration and Management **3**(1) (2008)
3. Stefansen, C., Rajamani, S., Seshan, P.: A work allocation language with soft constraints. Technical report, Microsoft Research India (2007) `http://stefansen.dk/papers/SoftConstraintAllocation.pdf`.
4. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow resource patterns. Technical report, Eindhoven University of Technology (2005)

A Tool for Process Merging in Business-Driven Development

Jochen M. Küster[1], Christian Gerth[1,2], Alexander Förster[2], and Gregor Engels[2]

[1] IBM Zurich Research Laboratory, Säumerstr. 4
8803 Rüschlikon, Switzerland {jku,cge}@zurich.ibm.com
[2] Department of Computer Science, University of Paderborn, Germany
{gerth,alfo,engels}@upb.de

Abstract. Business-driven development favors the construction of process models at different abstraction levels and by different people. As a consequence, there is a demand for consolidating different versions of process models by merging them. In this paper, we study a basic scenario, derive requirements and present a prototype for detecting and resolving changes between process models.

1 Introduction

The field of business process modeling has a long standing tradition. Recently, new requirements and opportunities have been identified which allow the tighter coupling of business process models to its underlying IT implementation: In Business-Driven Development (BDD) [1], business process models are iteratively refined, from high-level business process models into models that can be directly implemented. As a consequence, business process models are a key artifact in BDD and advanced techniques for consolidating different versions of a process model are needed.

In general, such techniques for consolidating and merging process models have to provide means for identifying differences between versions of process models and resolving these by merging parts of process models. Specific techniques for process merging heavily depend on the underlying modeling environment. Existing work on process change management has focused mainly on the question of dynamic process changes where changes are made on already running processes [2, 3]. Solutions include techniques for migrating process instances to a new process schema and for identifying those cases where this is not possible. In these approaches, process changes are usually captured in a change log which is maintained by the process-aware information system [4]. Recent work by Weber et al. [5] introduces the concept of compound change operations (change patterns) for process models and compares existing workflow tools with regards to their support for process change management.

In contrast to existing work, our approach addresses the situation where no change log describing process model changes exists. This is a common situation in process modeling tools such as the IBM WebSphere Business Modeler [6] and also occurs in scenarios where process models are exchanged across tool boundaries.

In this paper, we first discuss a basic scenario for process merging and then derive important requirements for a solution. We then present key concepts of a prototype for process merging realized as a plug-in for IBM WebSphere Business Modeler.

2 Requirements for Process Merging and Tool Overview

Within business-driven development, process models are the central modeling artifacts. In this context, business process models are manipulated in a team environment and multiple versions of a shared process model need to be consolidated at some point in time. A basic scenario is obtained when a process model V_1 is copied and then changed into a process model V_2, possibly by another person. After completion, only some of the changes shall be applied to the original model V_1 to create a consolidated process model. Figure 1 shows an example process model V_1 that has been changed into a process model V_2.

Both models describe the handling of a claim request by an insurance company. V_1 starts with an *InitialNode* followed by the *actions "Check Claim"* and *"Record Claim"*. Then, in the *Decision*, it is decided whether the claim is covered by the insurance contract or not. In the case of a positive result the claim is settled. In the other case the claim is rejected and closed, represented by the *actions "Reject Claim"* and *"Close Claim"*.

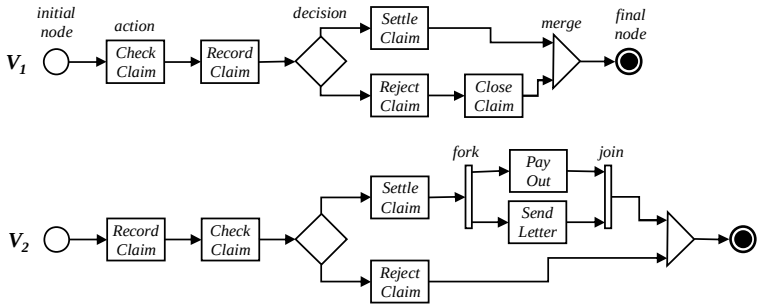


Fig. 1. Versions V_1 and V_2 of a business process model

Although process models V_1 and V_2 are similar at the first sight, there are some differences between the versions. The following differences can be detected:

- The positions of the *actions "Record Claim"* and *"Check Claim"* are changed.
- *Action "Close Claim"* does not exist in V_2.
- A new parallel structure (*Fork* and *Join*) is inserted in V_2 together with two *actions "Pay Out"* and *"Send Letter"*.

Process merging typically depends on the modeling language as well as on constraints of the modeling environment. In our case, the modeling language is given by the WebSphere Business Modeler which provides a language similar to UML 2.0 Activity Diagrams [7]. In our modeling environment, no syntax-directed editing of process models is performed and, as a consequence, also no change log is available. As such, in contrast to databases and existing approaches in process-aware information systems, there is no information about the performed changes on a process model. In the following, we describe the key requirements that a solution to process merging should fulfill:

- The solution must provide a technique to re-construct one possible change log which represents the transformation steps for transforming one process model into the other process model.
- The user should have the opportunity to select only some of the changes and apply it to the original model in order to obtain a new third model which can be considered as the merged process model.
- When applying changes, the user should not be restricted by prescribing a certain order whenever possible.
- Dependencies between change operations should be made explicit and taken into account when applying the changes. For example, when inserting a *Fork*, the corresponding *Join* should also be inserted in order to obtain a correct process model.
- The solution should provide user-friendly resolution of changes in the way that it reconnects inserted elements whenever possible and offers a possibility to perform related changes together at one time.

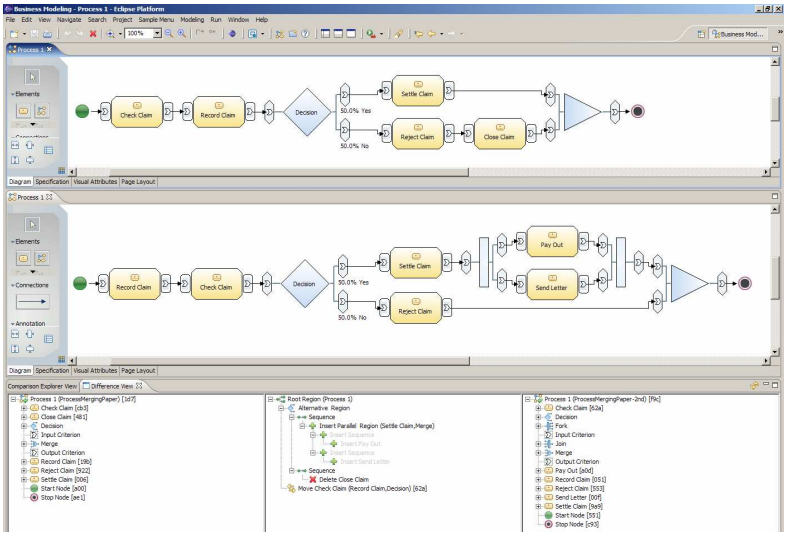


Fig. 2. Business Process Merging Prototype in the IBM WebSphere Business Modeler

Motivated by the requirements, we have developed an approach [8] for process merging which is divided into three steps. In the first step, we detect differences between the two process models using correspondences between model elements and the technique of Single-Entry-Single-Exit fragments (SESE fragments) [9]. In the second step, each detected difference is visualized according to the structure of the process models that is affected by it. The third step is then to resolve differences between the process models in an iterative way, based on the modeler's preferences. Here, for each difference, a resolution transformation is generated which resolves the difference between the two models and (if necessary) automatically reconnects the control flow.

The prototype has been implemented as an extension to the IBM WebSphere Business Modeler (see Fig. 2). It currently supports the following functionality [10]: copying

of business process models, initial creation and update of correspondences, decomposition of process models into SESE fragments and detection of differences between two versions of a process model. In addition, the prototype provides several views that allow to visualize and resolve differences as well as to manipulate correspondences.

Fig. 2 shows versions V_1 and V_2 of the business process model introduced earlier in this paper. The lower third of Fig. 2 illustrates the Difference View, which is divided into three columns. The left and right hand columns show versions V_1 and V_2 of the process model in a tree view, which abstracts from control flow details of the process and focuses only on model elements of the process. The middle column of the difference view displays the differences between the two versions, which are arranged according to the structure of the process models and visualizes dependencies between differences. Using this view, the differences can be iteratively resolved with our prototype.

3 Conclusion

User-friendly process merging is a key technique for practical business-driven development. In this paper, we have first studied a basic scenario of process merging in BDD and established key requirements. We have presented our prototype, which visualizes differences between versions of process models and enables the resolution of differences, by applying change operations in an iterative way that automatically reconnect the control flow. Future work will include the elaboration of our approach for merging process models in a distributed environment. In those scenarios, the concept of conflict becomes important because one resolution can turn the other resolution non-applicable.

References

1. Mitra, T.: Business-driven development. IBM developerWorks article, http://www.ibm.com/developerworks/webservices/library/ws-bdd, IBM (2005)
2. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data Knowl. Eng. **24**(3) (1998) 211–238
3. Rinderle, S., Reichert, M., Dadam, P.: Disjoint and Overlapping Process Changes: Challenges, Solutions, Applications. In Meersman, R., Tari, Z., eds.: CoopIS'04. Volume 3290 of LNCS., Springer (2004) 101–120
4. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems. Wiley (2005)
5. Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. In Krogstie, J., Opdahl, A.L., Sindre, G., eds.: CAiSE'07. Volume 4495 of LNCS., Springer (2007) 574–588
6. : IBM WebSphere Business Modeler. http://www.ibm.com/software/integration/wbimodeler/
7. Object Management Group (OMG): UML 2.0 Superstructure Final Adopted Specification. OMG document pts/03-08-02. (August 2003)
8. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Process Merging in Business-Driven Development. IBM Research Report RZ 3703, IBM Zurich Research Laboratory (2008)
9. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In: ICSOC 2007. Volume 4749 of LNCS., Springer (2007) 43–55
10. Gerth, C.: Business Process Merging - An Approach based on Single-Entry-Single-Exit Regions. Diplomarbeit, Universität Paderborn (October 2007)

Towards Situational Business Process Meta-Modelling

Oumaima Saidani[1], Selmin Nurcan [1,2]

[1] Université Paris 1 - Panthéon - Sorbonne Centre de Recherche en Informatique
90, rue de Tolbiac 75013 Paris, France,
[2] IAE de Paris Sorbonne Graduate Business School
Université Paris 1 - Panthéon - Sorbonne 21, rue Broca 75005 Paris France
{Oumaima.Saidani, Selmin.Nurcan}@univ-paris1.fr

Abstract. Business Process (BP) meta-models allow partial views of the processes. There may be adequate for some processes but not others. Situational engineering has proved its effectiveness in many engineering domains such as software and information system development. Reasoning on a situational approach for BP meta-modelling is a challenging research issue which can contribute to increase flexibility of meta-models and their adaptability to different organisation settings.

Keywords: Business Process meta-modelling, Flexibility, Adaptability.

1 Introduction

Current researches on business process (BP) modelling stress the importance of the flexibility and the adaptability of BP [2], [5] [7]. Reasoning on variability in modelling artifacts can meet the flexibility and context-awareness requirements by offering alternative solutions depending on the context and on the point-of-views of the decision-makers. A BP model is often formalized, at the type level, using a meta-model which captures the concepts supported by this model. We promote the idea that a single BP meta-model is still insufficient. A promising idea is to propose an approach for adapting and configuring existing meta-models according the organisation settings and users' objectives, rather than to advice for a single model which can be too complex for some requirements and simple for others. Accordingly, we focus on the flexibility at the type (meta-model) level of the BP which corresponds to the level 2 of the OMG four-level-architecture for the processes [1].

BPs are of various kinds and are defined in different levels of abstraction using various artifacts depending on the organisation settings and the purpose of the modelling. For instance, in mechanistic or production organisations, they are often prescribed in a detailed level since they shall be executed. On contrary, in adhocracies organisation, more freedom can be left to business actors for choosing how to perform the underlying business objectives. Therefore, the meta-models can be different and capture only some aspects of processes, however, sometimes their interrelationships could or should be taken into consideration and their complimentarily needs to be expressed. That is, in some situations, activity-oriented and product-oriented ones may need to be matched in order to determine which activity influences on which product and on which moment of the process. Also, strategy-oriented process meta-models require to be made operational using activity-oriented meta-models [3]. As well, [8] combines intention-oriented and state-based process modelling. Therefore, mechanisms for adapting existing models to specific requirements need to be developed. Our aim in this paper is to propose such mechanisms. Our motivation behind this proposal is that: (i) a BP meta-model which is designed for a specific organisation setting is not necessarily adequate for others; (ii) since several meta-models have proved their effectiveness in many business areas, it does not seem required to create new models.

In the information systems development (ISD) community, method engineering (ME) has been introduced as a response to the need for methods adapted to specific ISD project situations, and to the failure of the methods known as "universal" [9]. One area of ME is Situational Method Engineering (SME). SME is based on four principles: meta-modelling, flexibility, reuse and modularity [10]. We can highlight that the ISD requirements on flexibility and adaptability that are behind the ME emergence in the ISD field were similar to those currently observed in the BPM field, we thus base our reasoning on SME mechanisms. The paper is structured as follows. Section 2 introduces an overview of the proposed approach with illustrative examples. Section 3 concludes the paper.

2 Overview of the proposed approach with examples

Building the adequate meta-model can be done following several manners, for instance, by assembling relevant concepts, by constructing a core meta-model and enhancing it with required concepts, etc. With analogy to the *method* in the ISD field, we introduce the concept of *business method* which consists of a set of reusable components that we identify as *BP meta-model chunks*. In the remainder of the paper, we simply denote them by *BPM-chunk*. BPM-chunks are independent and stored in a chunk repository. They can be reused in order to build new meta-models or to enhance existing ones. The can be simple (e.g. a concept) or compound (e.g. a set of concepts, properties and relationships between them). In the reminder, we introduce some examples of BPM-chunks that constitute a partial vision of the repository. We underline the use of some operators for managing them. We are inspired from operators defined in [4]. Fig. 1 shows an example of meta-model (M_0) which can be

extended, according to the situation, by independent chunks (C_0, C_1, C_2) resulting on the meta-models shown in Fig. 1 (right).

PM0 and PM1. PM_0 (Fig. 1 (left)) keeps a minimal set of features. It may be suitable for some organisation settings, e.g. stable organisations with minor changes and few operations. Otherwise, defining operations in a finer granularity, and in frequently changing organisations, may involve a cumbersome work. In such situation, PM_0 can be extended with C_0 (Fig. 2) in order to construct PM_1. C_0 serves, in PM_1, as a link between roles and operations, BPs are relied to functions rather than operations. PM_1 is discussed in detail in [5]. Extending PM_0 requires updating the relationships *can-hold* and *comprises* and defining a new one: *satisfies*. Let *CONCEPTS* the set of concepts of the chunk repository. A relationship can associate many concepts. Formula (1) represents the mapping of a relationship *r* onto a set of concepts. Let *create-relationships*, *update-relationships* and *delete-relationships* three operators allowing respectively creating, updating and deleting relationships between entities. These operators can be applied so that the relationship *can-hold* between the entities *Role* and *Operation* -in PM_0- is removed, and the same is created in PM_1. As well the relationship *Comprises* between the entities *Business-Process* and *Operation*, in PM_0, are removed and those between *Business-Process* and *Function* are created in PM_1. Finally, the relationship *Satisfies* between *Operational-Goa*l and *Operation* is created.

$$relationship-concept(r:RS) \rightarrow 2^{CONCEPTS},\ relationship-concept(r_i) \subseteq CONCEPTS \quad (1)$$

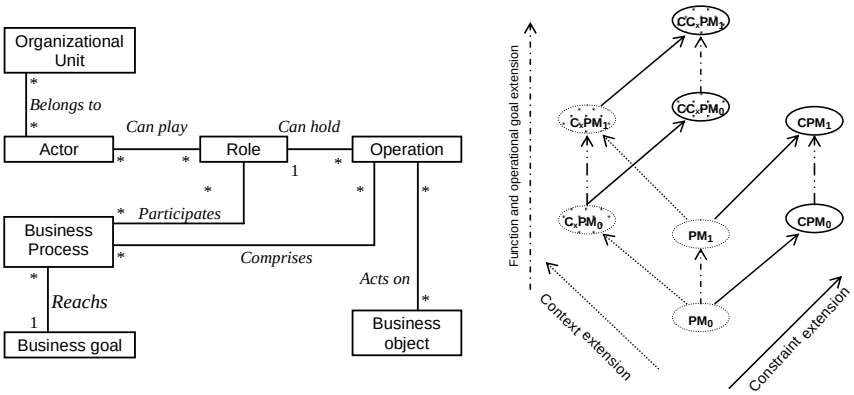


Fig.1 The Meta-model of PM_0 (left) and a set of BP meta-models and their relationships (right)

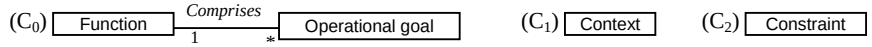


Fig.2 Examples of BPM-chunks representing respectively C_0, C_1 and C_2.

C_xPM. C_xPM_0 and C_xPM_1 extend PM_0 and PM_1 with chunk C_1 (*Context*) (Fig. 2). C_1 can be added to an existing meta-model for capturing context knowledge which can impact the assignments relationships of a process model (e.g. the ability of actors for playing roles according to a given context [6]. C_xPM_0 is constructed by extending

PM_0 with C_1. The integration of PM_0 and C_1 requires the use of the operator *update-relationships* so that the relationships *can-play*, *can-hold*, *implies* and *comprises* defined in PM_0 are related also to C_1. The same logic can be applied for C_xPM_1.

CPM. In some situations, organisation policies need to be enforced impacting assignments decisions, for instance, separation of duties (see [5] for more details about constraints). Building CPM_0 (resp. CPM_1) requires extending PM_0 (resp. PM_1) with C_2 (Fig. 2). This practice needs using the operator *update-relationships* so that the constrained binary relationships assignment (e.g., *can-play*) in PM_0 (resp. in PM_1) are related to C_2 in CPM_0 (resp. CPM_1). The same reasoning can be applied for C_xPM_1.

4 Conclusion and Future Work

This paper provides a start points for the definition of a methodology allowing the design of adaptive and flexible BP meta-models according to the situation at hand. We have introduced the concepts of BPM-chunk and business method as well as example of chunks and meta-models in order to illustrate our proposal. We promote the fact that the final business process model has to be created from the set of proposed chunks in order to suit to a particular situation. This approach aims to make easier the definition of flexible and customised meta-models. Dealing with situation-awareness raises many issues which need further research such as: the context influencing the selection of adequate chunks and the adaptation process.

References

1. BPMI.org, OMG: Business Process Modelling Notation Specification. Final Adopted Specification. Object Management Group (2006)
2. Nurcan, S.: A survey on the flexibility requirements related to business processes and modelling artifacts. Proc. HICSS'08, Big Island, Hawaii, USA, 7-10 January (2008)
3. Nurcan, S., Etien, A., Kaabi, R., Zoukar, I. and Rolland, C.: A strategy driven Business Process Modelling Approach. Business Process Management Journal, Vol. 11, N° 6 (2005)
4. Ralyté, J.: Ingéniérie des méthodes à base de composants. PhD thesis, University of Paris 1-Sorbonne (2001)
5. Saidani, O. and Nurcan, S.: A Role-Based Approach for Modelling Flexible Business Processes. In proceedings of BPMDS'06 (2006)
6. Saidani, O., Nurcan, N.: Towards Context Aware Business Process Modelling, BPMDS'07, Held in conjunction with CAiSE'07, Trondheim, Norway (2007)
7. Snowdon, R. A., Warboys, R. M. Greenwood, C. P. Holland, P. J. Kawalek, D. R.: On the architecture and form of flexible process support. SPIP Journal, Vol. 12, N° 1 (2007)
8. Soffer, P., Rolland, R.: Combining Intention-Oriented and State-Based Process Modelling. ER (2005) p. 47-62
9. Rolland, C.: L'ingénierie des méthodes : une visite guidée, e-TI - la revue électronique des technologies d'information, Premier Numéro, 25 octobre 2005 (2005)
10. Rolland, C.: Method Engineering: Trends and Challenges. In IFIP International Federation for Information Processing, Vol. 244, Situational Method Engineering: Fundamentals and Experiences, eds, Ralyté, J., Brinkemper, S., Henderson-Sellers, B., (Boston Springer) (2007)

Improving Data Independence, Efficiency and Functional Flexibility of Integration Platforms

Matthias Böhm[1], Jürgen Bittner[2], Dirk Habich[3], Wolfgang Lehner[3], and Uwe Wloka[1]

[1] Dresden University of Applied Sciences, Database Group
mboehm@informatik.htw-dresden.de
wloka@informatik.htw-dresden.de
[2] SQL Gesellschaft für Datenverarbeitung mbH Dresden
juergen.bittner@sql-gmbh.de
[3] Dresden University of Technology, Database Technology Group
dirk.habich@inf.tu-dresden.de
wolfgang.lehner@inf.tu-dresden.de

Abstract. The concept of Enterprise Application Integration (EAI) is widely used for integrating heterogeneous applications and systems via message-based communication. Typically, EAI servers provide a huge set of specific inbound and outbound adapters used for interacting with the external systems and for converting proprietary message formats. However, the main problems in currently available products are the monolithic design of these adapters and performance deficits caused by the need for data independence. First, we classify and discuss these open problems. Second, we introduce our model-driven DIEFOS (data independence, efficiency and functional flexibility using feature-oriented software engineering) approach and show how the feature-based generation of dynamic adapters can improve data independence, efficiency and functional flexibility. Finally, we analyze open research challenges we see in this context.

Keywords: Enterprise Integration Platform, Application Integration, Adapter Architecture, Dynamic Adapters, DIEFOS Approach

1 Introduction

The trend towards heterogeneous environments comes with an increase in importance of Enterprise Application Integration (EAI). Such an integration platform consists of a set of inbound adapters, a core message broker and a set of outbound adapters. The large number of supported external system types results in the need for data independence (independent-system-type data representations for internal processing) and, simultaneously, for efficient integration task processing (minimum overhead for data independence). These requirements—but particularly the first one—typically result in very generic inbound and outbound adapter architectures. There, the architecture of such adapters is quite monolithic, which results in low functional flexibility of such software components. This means that for each external system type, a single adapter is needed, though specific functional modules could be reused. An example for this is a TCP connection handler which sends the specific messages to the physical target systems—it might be reused by several adapters like HL7 and B2MML adapters.

In order to solve this problem of monolithic adapters (which affects the functionality as well as the performance), we describe the problem characteristics in Section 2 from a pragmatic perspective, influenced by the commercial enterprise integration platform TransConnect®. Further, we propose our DIEFOS approach and explain its core phases in Section 3. In general, one of the main questions in this context is whether or not model-driven approaches can be applied in the field of application integration. Finally, in Section 4, we conclude our paper and highlight open research challenges we see.

Although there is a lot of related work concerning MDA techniques [1] and MDA tools (e.g., AndoMDA, MOFLON [2] and Fujuba), only a very low support for model-driven development can be recognized in application integration platforms (e.g., SQL GmbH TransConnect, SAP XI, BEA Integration, MS Biztalk and IBM Message Broker). In this context, the so-called RADES approach [3] tries to give an abstract view on EAI solutions using technology-independent and multi-vendor-capable model-driven engineering methodologies. Unfortunately, this approach does not focus the problems considered here (data independence, efficiency and functional flexibility). Further, also approaches for automatic generation of Web service adapters [4–6] and BPEL adapters [7]. These techniques are too specific to the integration technology used. In addition, the semi-automated generation of adapters for legacy applications is addressed in [8]. However, such a semi-automated approach is not suitable. The dynamic adapter generation approach [9] addresses the dynamic adding of new data sources and their invocation rather than the functional flexibility of adapter generation.

2 Problem Description

Here, we introduce a generalized EAI server architecture and describe the addressed problems. As illustrated in Figure 1, an EAI server consists of typical components. There is a set of `Inbound Adapters`, which listen passively to incoming messages and convert these into internal representations. Further, the internal messages are processed by the runtime environment. This environment uses a set of `Outbound Adapters` to actively interact with external systems. According to the layers of transformations [10], the adapters realize the layers *transport* and *data representation*. The main problem is the monolithic adapter architecture with very generic message interfaces, which cause the use of uniform message representations (e.g., XML messages). This also causes the problem of *P1: Poor Performance*. Further problems include *P2: Functional Restrictions* (chosen technology), *P3: Development Effort* (redundant functionality) and *P4: Data Independence* (dependencies between adapter interactions). To overcome these problems, message representations (alternative representations, schemas) as well as adapter architectures (generic adapters, adapter generation) have to be reconsidered. We follow an adapter generation approach that allows different alternative message representations.

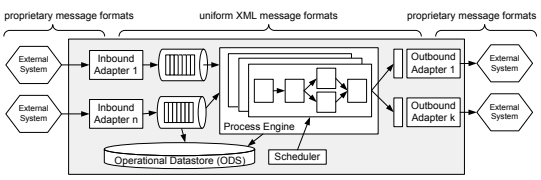


Fig. 1. Generalized EAI Server Architecture

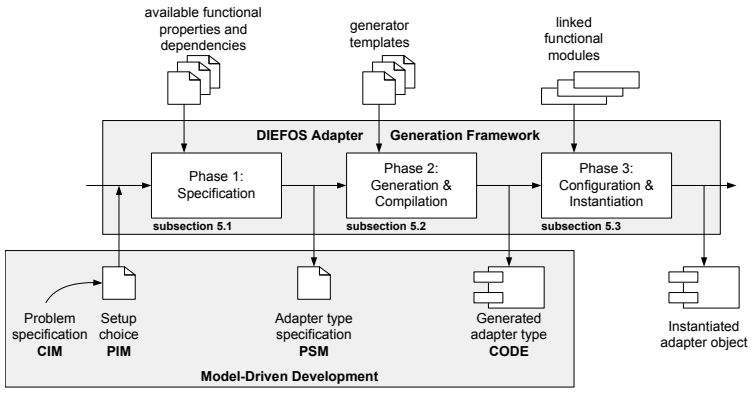


Fig. 2. DIEFOS Generator Framework - Macro-Architecture

3 DIEFOS Approach

The DIEFOS approach (**D**ata **I**ndependence, **E**fficiency and functional flexibility using a **F**eature-**O**riented **S**oftware-development) solves the problems described in Section 2. Basically, this framework—whose macro-architecture is illustrated in Figure 2—comprises the three phases *1: Specification*, *2: Generation & Compilation* and *3: Configuration & Instantiation*.

First, an informal problem specification (CIM) is provided. It is manually transformed into a setup choice (the applicable alternatives are given by feature diagrams similar to Figure 3), which represents the platform-independent model (PIM). This choice—in conjunction with available functional properties and dependencies—is used in order to create the formal adapter type specification (PSM) using an XML model representation. Second, within the generation step, a java class (CODE) is generated from the adapter type specification input, using specific code templates. Finally, this class is compiled and loaded into the JVM. Third, the created instance of the generated adapter as well as the linked functional modules have to be configured. We use an approach where specific functionality can be reused in function modules almost without any overhead. So, during runtime, these hard-coded modules are used as a library.

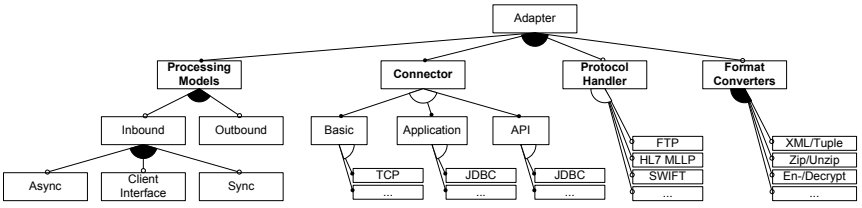


Fig. 3. Adapter Type Specification Feature Diagram

4 Summary and Open Challenges

The overall motivation for this work was the existence of the four pragmatical problems: (1) poor performance, (2) functional restrictions, (3) development effort and (4) need for data independence. The goal was to realize an adapter architecture which ensures data independence with minimal overhead concerning the processing efficiency. Further, the functional flexibility should also be maximized while minimizing the development effort at the same time using model-driven development.

In order to solve the given problems, we first observed the adapter problem characteristic of real-world integration platforms. Second, we proposed the DIEFOS approach, which overcomes the given problems using a model-driven generation approach. This allows for dynamic composition of adapters and comprises the three phases: *1: Specification*, *2: Generation & Compilation* and *3: Configuration & Instantiation*. The goal is to generate adapter types in a feature-oriented manner. The dynamic combinations of format converters, protocol handlers and physical connectors make it possible to ensure the data independence, functional flexibility and even the efficiency can be ensured. However, there are open problems and challenges. Those include but are not limited to: (1) a conceptual adapter specification model, (2) the debugging and testing of generated dynamic adapters, (3) the use of a configuration history for consistent recovery processing, (4) the self-configuration for the generation of adapter specifications based on workflow descriptions and (5) the separation of data and meta data for functional correctness and avoidance of runtime errors. Due to the practical relevance, we want to invite interested research groups and industry vendors to participate in the discussion on this approach and open challenges.

References

1. Kleppe, A., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley (2003)
2. Amelunxen, C., Königs, A., Rötschke, T., Schürr, A.: Moflon: A standard-compliant metamodeling framework with graph transformations. In Rensink, A., Warmer, J., eds.: Model Driven Architecture - Foundations and Applications. (2006)
3. Dorda, C., Heinkel, U., Mitschang, B.: Improving application integration with model-driven engineering. In: ICITM. (2007)
4. Benatallah, B., Casati, F., Grigori, D., Nezhad, H.R.M., Toumani, F.: Developing adapters for web services integration. In: CAiSE. (2005) 415–429
5. Lee, K., Kim, J., Lee, W., Chong, K.: A tool to generate an adapter for the integration of web services interface. In: CBSE. (2006) 328–335
6. van den Heuvel, W.J., Weigand, H., Hiel, M.: Configurable adapters: the substrate of self-adaptive web services. In: ICEC. (2007) 127–134
7. Brogi, A., Popescu, R.: Automated generation of bpel adapters. In: ICSOC. (2006) 27–39
8. Pieczykolan, J., Kryza, B., Kitowski, J.: Semi-automatic creation of adapters for legacy application migration to integration platform using knowledge. In: International Conference on Computational Science (4). (2006) 252–259
9. Gong, P., Gorton, I., Feng, D.D.: Dynamic adapter generation for data integration middleware. In: SEM. (2005) 9–16
10. Hohpe, G., Woolf, B.: Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions. Addison-Wesley (2004)

Validation of UML Conceptual Schemas with Operations*

Anna Queralt and Ernest Teniente

Universitat Politècnica de Catalunya
{aqueralt, teniente}@lsi.upc.edu

Abstract. The purpose of validating a conceptual schema is to check whether it specifies what the designer intended. Our approach to validation consists in translating the schema into logic in such a way that any reasoning method can be used to perform the validation tests defined by the designer. An important contribution of this work is that it takes into account the operations defined in the schema.

1. Introduction

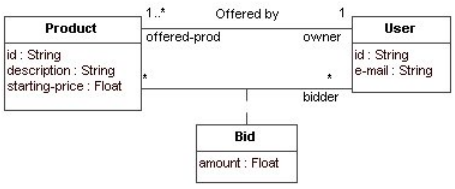
In software quality assurance, the purpose of the validation process is to answer to the question *Am I building the right system?*. In the context of conceptual modeling, validation can be used to assure the quality of a conceptual schema instead of a piece of code. To this end, it is desirable to provide the designer with some assistance, so that he can check whether the conceptual schema properly specifies what he intended.

A *conceptual schema* consists of a *structural schema*, which defines the relevant static aspects of the domain, and a *behavioral schema*, which defines the only changes that can be performed on the information. It includes a set of *system operations,* which view the system as a black box and are not assigned to classes [4].

Fig. 1 shows the structural schema of an on-line auction site that we will use as an example. The system stores information about users, and each user is the owner of a set of products. Users bid for products by specifying the amount they offer. Additionally, this structural schema includes some textual integrity constraints that must be satisfied.

A test that the designer can perform to validate the schema is to check whether it accepts at least one instance satisfying all the constraints. For example, the following instantiation of the schema: *"Mick is a user who owns a book, and bids 200$ for a bicycle, owned by Angie, who had set a starting price of 180$"* satisfies all the graphical and textual constraints. However, the fact that the structural part of a schema is satisfiable does not necessarily imply that the whole conceptual schema also is. That is, when we take into account that the only changes admitted are those specified by the operations, it may happen that the properties fulfilled by the structural schema alone are no longer satisfied. For instance, if the schema does not contain any operation that successfully populates the class *User*¸ it will not be possible to populate any other class (instances of product will neither exist, since each *Product* needs an *owner* and, in turn, bids need products and users).

* This work has been partly supported by the Ministerio de Ciencia y Tecnología under projects TIN2005-06053 and TIN2005-05406



Integrity constraints:

- Users and Products are identified by their id
- The amount of a bid must be greater than the starting price of the product
- The owner of a product cannot bid for it

Fig. 1. The structural schema of an on-line auction site

This means that our conceptual schema must include, for example, an operation *registerUser* that a designer could define by means of the following operation contract:

```
Op:   registerUser(id:String, e-mail:String)
Pre:
Post: User.allInstances()->exists(u | u.oclIsNew() and
      u.id=id and u.e-mail=e-mail)
```

We assume a strict interpretation of the contracts [7], which prevents the application of an operation if a constraint is violated by the state satisfying the postcondition.

In this work we propose an approach to validate a UML conceptual schema, with its constraints and operations specified in OCL[1]. To do this, we translate the schema into a set of logic formulas. The result of this translation ensures that the only changes allowed are those specified in the behavioral schema, and can be validated using any reasoning method or tool that is capable of dealing with negation of derived predicates.

2. Translation of a Conceptual Schema into Logic

When considering the behavioral schema in the validation, it must be taken into account that the population of classes and associations at a certain time *t* is just the result of all the operations that have been executed before *t.* For instance, *Angie* may only be an instance of *User* at a time *t* if the operation *registerUser* has created it at some time before *t* and no other operation has removed it between its creation and *t.*

For this reason, it must be guaranteed that the population of classes and associations at a certain time depends on the operations executed up to that moment. To do this, we propose that operations are the basic predicates of our logic formalization. Classes and associations will be represented by means of derived predicates, and their derivation rules will ensure that their instances are precisely given by the operations executed.

Then, an instance of a predicate *p* representing a class or association exists at time *t* if it has been added by an operation at some time *t2* before *t*, and has not been deleted by any operation between *t2* and *t.* Formally, the general derivation rule is:

$$p([P,],P_1,...,P_n,T) \leftarrow addP([P,]P_1,...,P_n,T2) \wedge \neg deletedP(P_i,...P_j,T2,T) \wedge T2 \leq T$$

$$deletedP(P_i,...,P_j,T1,T2) \leftarrow delP(P_i,..,P_j,T) \wedge T > T1 \wedge T \leq T2$$

where *P* is the OID (object identifier), which is included if *p* is a class. $P_i,...,P_j$ are the terms of *p* that suffice to identify an instance of *p.* In particular, if *p* is a class or association class, $P=P_i=P_j$. Predicates *addP* and *delP* are also derived predicates that hold if some operation has created or deleted an instance of *p* at time *T*, respectively.

[1] The subset of OCL considered consists of all the OCL operations that result in a boolean value, including *select* and *size*, which can also be handled by our method.

Let $op\text{-}addP_i$ be an operation, with parameters $Par_1,...,Par_n$ and precondition pre_i such that its postcondition specifies the creation of an instance of a derived predicate *p.* For each such operation we define the following rule:

$addP([P,]Par_i,...,Par_kT) \leftarrow op\text{-}addP_i([P,]Par_1,...,Par_m,T) \wedge pre_i(T_{pre}) \wedge T_{pre}=T\text{-}1$

where $Par_i,..,Par_k$ are those parameters of the operation that indicate the information required by the predicate *p,* and *T* is the time in which the operation occurs. The literal $pre_i(T_{pre})$ is the translation of the precondition of the operation [6].

Similarly, for each operation $op\text{-}delP_i(Par_1,...,Par_n,T)$ with precondition pre_i that deletes an instance of *p* we define the derivation rule:

$delP(Par_i,...Par_j,T) \leftarrow op\text{-}delP_i(Par_1,...,Par_n,T) \wedge pre_i(T_{pre}) \wedge T_{pre}=T\text{-}1$

where $Par_i,...,Par_j$ are those parameters that identify the instance to be deleted.

For instance, the class *User* of our example will be represented by:

$user(U,Id,Email,T) \leftarrow addUser(U,Id,Email,T2)$

$addUser(U,Id,Email,T) \leftarrow registerUser(U,Id,Email,T)$

where *U* corresponds to the unique OID. In turn, *addUser* is a derived predicate whose definition depends on the operations of the behavioral schema that create instances of *User*. In particular, it will hold if the operation *registerUser* has been executed.

Since our schema does not include any operation to remove users, the derived predicate *deletedUser* must not be defined in this case.

Additionally, a set of constraints must be added to the translation to ensure the correct occurrence of the operations. In particular, since two operations cannot occur at the same time, for each operation *O* with parameters $p_1,...,p_n$ we define the following constraint for each parameter p_i: $\leftarrow o(P_1,...,P_n,T) \wedge o(P_1,...,P_n,T) \wedge P_i1 <> P_i2$.

And for each pair *O, O2* of operations: $\leftarrow o(P_1,...,P_n,T) \wedge o2(Q_1,...,Q_m,T)$.

Moreover, all constraints of the UML structural schema are also translated into formulas in denial form according to [6], but now they are defined in terms of derived predicates instead of basic ones.

3. Our Approach to Validation

Our approach to validation is aimed at providing the designer with the ability to define his own tests to see how the schema behaves in a particular situation, and then compare the results obtained with the ones expected according to the requirements. This will be done taking into account both the structural and the behavioral parts of the schema.

Our method consists in reducing the problem to checking the satisfiability of a derived predicate. In this way, a derived predicate that formalizes the desired test is defined. With this input, together with the translated schema itself, any satisfiability checking method that is able to deal with derived predicates can be used to validate the schema. For instance, an interesting question could be "Can all the classes of the schema be populated?". The following derived predicate formalizes this test:

$populated \leftarrow user(U,Uid,Em,T) \wedge product(P,Pid,Price,Own,T) \wedge bid(B,Pr,Us,Amt,T)$

It can easily be seen that the schema of our example does not satisfy this property, since its behavioral schema does not allow creating an instance of *User* owning at least one *Product*, as required by the cardinality constraint in *Offered by.* This means that the conceptual schema is not correct and the designer must solve this situation either by

making the operation *registerUser* responsible of creating instances of the association *Offered by*, or by changing the cardinality constraint from 1..* to *.

By studying the results of the tests, and with his knowledge about the requirements, the designer will be able to decide if the schema is correct, and modify it if necessary.

4. Related Work

We briefly summarize the work related to the validation of conceptual schemas with a behavioral part. One of the first methods to do this belongs to the area of deductive databases [2], and proposes a framework to validate a schema using planning methods.

In the context of UML, there is an approach that combines two methods: UML-B [8] to translate a UML schema into B, and ProB [5], to validate it. However, UML-B only accepts a subset of the UML, and does not admit OCL. Moreover, ProB requires that the possible values of types are enumerated, which does not guarantee completeness.

The rest of existing UML/OCL approaches that somehow consider the behavioral part may report as valid a state satisfying all the constraints but that is impossible to construct using the operations defined in the schema [1, 3].

5. Conclusions

We have proposed a new approach to validate a complete UML conceptual schema, with its textual constraints and operations expressed in OCL. Our approach helps the designer to check that the schema defined correctly specifies the requirements.

This is achieved by translating the conceptual schema, including its behavioral part, into a logic representation such that any satisfiability checking method able to deal with derived predicates can be used to validate the schema.

References

1. Brucker, A. D., Wolff, B.: The HOL-OCL Book. Swiss Federal Institute of Technology (ETH), 525 (2006)
2. Costal, D., Teniente, E., Urpí, T., Farré, C.: Handling Conceptual Model Validation by Planning. CAiSE'96 LNCS 1080 (1996) 255-271
3. Gogolla, M., Bohling, J., Richters, M.: Validating UML and OCL Models in USE by Automatic Snapshot Generation. Software and System Modeling 4(4) (2005) 386-398
4. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd edn. Prentice Hall PTR (2004)
5. Leuschel, M., Butler, M.: ProB: An Automated Analysis Toolset for the B Method. Software Tools for Technology Transfer DOI: s10009-007-0063-9 (2008)
6. Queralt, A., Teniente, E.: Reasoning on UML Class Diagrams with OCL Constraints. In: Conceptual Modeling - ER 2006. LNCS 4215 (2006) 497-512
7. Queralt, A., Teniente, E.: Specifying the Semantics of Operation Contracts in Conceptual Modeling. Journal on Data Semantics JoDS VII (2006) 33-56
8. Snook, C., Butler, M.: UML-B: Formal Modeling and Design Aided by UML ACM Trans. on Soft. Engineering and Methodology 15(1) (2006) 92-122

Towards a Framework for B2B Integration Readiness Assessment and Guided Support of the SMEs

Spiros Mouzakitis[1], Fenareti Lampathaki[1], Dimitris Askounis[1]

[1] National Technical University of Athens, Athens, Greece
{smouzakitis, flamp, askous},@epu.ntua.gr

Abstract. In today's world with companies operating in a global business environment. Most enterprises, and especially the SMEs, lack the necessary business culture, technical and non-technical infrastructure and economic flexibility in order to efficiently adjust to the environment of a B2B integration framework. This paper proposes an Enterprise Integration Assessment Framework (EIAF) and its support software system that aims to aid enterprises in adopting a multienterprise (B2B) integration approach by evaluating its situational status and by estimating the expected integration impact based on the evaluation results.

Keywords: B2B integration, readiness assessment, interoperability, impact assessment, performance indicators.

1 Introduction

In today's competitive business environment, companies are seeking ways to perform transactions efficiently and effectively. The Internet has created a flexible platform for the buying and selling of products and services. As businesses recognize the need for employing efficient methods for the vertical exchange of goods and services, they are considering the adoption of functional business-to-business (B2B) applications and technologies that allow transactions in "real time." [2]. The purpose of B2B integration is to improve profitability through establishing relationships with other organizations that will allow supply-chain planning, collaboration, product pricing, logistics and distribution management, and procurement efficiencies [1].

Modern B2B technologies, have solved major technical issues of traditional EDI but due to a vast number of non-technical adoption barriers, the efforts for business-to-business integration are still enormous [10]. Although there are some approaches and guidelines available that address the adoption phase, most Enterprises, especially the SMEs, struggle to overwhelming the existing hurdles due to the following key-barriers.To solve the current issues, we present a comprehensive framework that measures the readiness of an enterprise to adopt a multienterprise (B2B) integration approach and, based on the findings, provides thereafter guided support to the SMEs with a view to overcoming the related barriers.

2 Methodology

2.1 Framework Overview

The Enterprise Integration Assessment Framework (EIAF) presents a research framework and a web-based platform with an aim to aid the Enterprises, and especially the SMEs transition to a B2B integration environment. The EIAF will provide insight for the B2B integration adoption phase by:

- Recognizing and classifying common integration patterns and styles
- Identifying key technical and non-technical factors that affect the transition
- Presenting a comprehensive methodology for the assessment of an Enterprise's readiness to integrate with other Enterprises
- Identifying aspects that affect the integration impact
- Developing a "knowledge framework" which can support the enterprises in their brainstorming for B2B integration

2.2 Framework's Design Scheme

2.2.1 Modelling an abstract B2B integration framework

The first step in the conception of the EIAF framework is the abstraction of existing and upcoming B2B integration architectures and solutions. The abstraction process involves the study and analysis of the most important both dominant and promising integration technologies, solutions and standards. This analysis will lead to a categorization and classification of the involved patterns based on both technical and non-technical aspects of a B2B integration solution. Then, a generic model will be produced for each possible abstraction of two or more categories.

The output of this procedure will be a set of generic enterprise application integration models covering a broad range of integration styles and technologies.

2.2.2 Specification of the Assessment Indicators

In order to specify practical and appropriate evaluation indicators measurable objectives must be first identified clearly. Based on the prototype B2B integration framework model, a number of goals can be realized through discourse and negotiation with representative enterprises, such as seamless data exchange in automated transactions between suppliers and partners that is characterized by maintainability, trust and confidentiality, strong security, low implementation/integration cost/effort, low operational cost/effort, value-added functionality, high quality of service aspects, such as speed and availability.For each section and integration level the appropriate indicators are conceived on the basis of

that they are related to one or more of the defined goals. Since hundred of combinations can occur, the number of indicators must be set to the minimum possible by eliminating indicators that coincide with each other or indicators that are expected to offer the lowest visibility of the objectives.

2.2.3 Design and development of the evaluation methodology

EIAF's evaluation methodology will be based on one thorough questionnaire that will contain all the questions depending on the business sector of the enterprise. In both cases, questions represent one or more assessment indicators. A basic question for example could be "What is the number of (major) internal IT control breaches during this year?" Since most questions refer to intangible assets, a normalized performance scale must be defined, and all answers (belonging to different measures - percentage, numeric, pre-defined choices) must be transformed to values in the common normalized scale. In order to achieve this, for each question different quality points Li should be defined that have corresponding points to the normalized performance scale. The performance indicators FID are conflated with given weights wi in a similar way in order to produce a performance value for each Indicator Category. Then intermediate tables are used that contain the indicator value thresholds with the corresponding performance score descriptions (Excellent, Good, Fair, Poor, and Very Poor) for translating these values into graspable scores.

2.2.5 Performance Impact estimation design

Enterprise integration has been found to lead to improved enterprise performance [7][8][9]. In the context of EIAF's assessment framework performance impact means that a B2B integration solution when used in the enterprise and interdependencies environment will improve some unit level performance measure. EIAF's will provide a rough impact estimation that depends closely to a vast number of input parameters provided by the stakeholders in combination with the B2B integration readiness results. These parameters are organized to macroeconomics,legal and statutory framework,pricing,integration effort,the exchange's technology vendor relationships,the exchange's partnerships and members.

2.2.6. Design and development of the support software system

In the context of the EIAF project, a modern technological platform is developed to support the application of the EIAF methodology in a cost-effective and easy manner. This platform is an intelligent web based system which will evaluate the situational status of a member Enterprise. More specifically , it will provide the level of readiness to adopt a B2B integration solution, detailed analysis of the evaluation results and an n depth examination of the weak points that diminish the worth of the B2B integration

3 Conclusions and Future Work

The primary contribution of this paper was to propose a research methodology that evaluates the readiness of an enterprise to adopt a B2B integration solution. Future work includes collecting the complete set of the assessment indicators, adjusting the evaluation method and proving the framework's merits by collecting data and performing statistical analysis to validate each of the proposed methodologies. Work is going forward on using the research framework to understand SMEs B2B integration in the Greece. Additional findings and results are expected during the EIAF system's pilot operation that will be circulated through further dissemination activities.

Acknowledgments. The authors of this paper gratefully acknowledge the financial support from Greece General Secretariat of Research and Technology, Ministry of Development, under the program "Business models and technologies for automated business to business transactions".

References

1. Ranganathan, C. (2003) Evaluating the Options for Business-To-Business Exchanges, Information Systems Management, 20, 3, 22-29.
2. Donald L. Amoroso, Sandy Vannoy, Translating the Adoption of B2B e-Business into Measurable Value for Organizations, Proceedings of the 39th Hawaii International Conference on System Sciences – 2006
3. Ranganathan, C., Thompson, S.H. Teo, Dhaliwal, Jasbir S., Ang, James S.K., and Hyde Micki. (2001) Facilitators and Inhibitors for Deploying Business-To-Business E-Commerce Applications: A Multi-Method Cross-Cultural Study, Proc
4. Spiros Mouzakitis, Fenareti Lampathaki, Christoph Schroth, Ulrich Scheper, Till Janner, Towards a common repository for governmental data: A modeling framework and real world application, in Enterprise Interoperability II: New Challenges and Approaches (Springer), Proceedings of the 3rd International Conference Interoperability for Enterprise Software and Applications I-ESA 2007, Funchal (Madeira Island) Portugal, March 2007
5. Ronald E. Giachetti, et al, Research Framework for Operationalizing Measures of Enterprise Integration, IFIP International Federation for Information Processing 2005
6. Giachetti, R. (2004). "Enterprise Integration: An information integration perspective." International Journal of Production Research 42(6): 1147-1166.
7. Armistead, C. and J. Mapes (1993). "The impact of supply chain integration on operating performance." Logistics Information Management 6(4): 9-15.
8. Frohlich, M. T. and R. Westbrook (2001). "Arcs of integration: an international study of supply chain strategies." Journal of Operations Management 19: 185- 200.
9. Brunnermeier, S. B. and S. A. Martin (2003). "Interoperability costs in the US automotive supply chain." Supply Chain Management: An International Journal 7(2): 71-82.
10. George Gionis, Spyros Mouzakitis, Till Janner, Christoph Schroth, Sotirios Koussouris and Dimitris Askounis, Implementing Next Generation e-Business Platforms for Small and Medium Enterprises, PCI2007 Conference, Patras, 2007 +
11. Ruh, W. A., F. X. Maginnis, et al. (2000). Enterprise Application Integration: A Wiley Tech Brief. New York, NY, John Wiley & Sons Inc.