13th IEEE INTERNATIONAL

Requirements Engineering CONFERENCE

DOCTORAL CONSORTIUM
August 30th, 2005

La Sorbonne, FRANCE

August 29th - September 2nd, 2005

Sponsors

IEEE Computer Society Université Paris 1

IEEE COMPUTER SOCIETY

IEEE

– PANTHÉON - SORBONNE –
UNIVERSITÉ PARIS 1

r

Proceedings of the Doctoral Consortium at the 13th IEEE International Requirements Engineering Conference

August 30, 2005
Paris, France

Edited by
Nancy A. Day
University of Waterloo

University of Waterloo
School of Computer Science
Technical Report CS-2005-23

Preface

The Doctoral Consortium of the IEEE International Conference on Requirements Engineering (RE) 2005 is intended to bring together PhD students working on foundations, techniques, tools and applications of requirements engineering, and give them the opportunity to present and discuss their research in a constructive and international atmosphere. The goals of the Consortium are:

- to provide a setting for mutual feedback on participants' current research and guidance on future research directions,
- to develop a supportive community of scholars and a spirit of collaborative research,
- to provide an opportunity for student participants to interact with established researchers and others in the requirements engineering community.

The Doctoral Consortium Committee selected ten submissions for this year's event from the eighteen submitted proposals. The presentations cover a range of requirements engineering topics. One accepted student was unable to attend.

I thank all of the students who submitted proposals, the members of the Doctoral Consortium Committee who thoroughly reviewed and discussed all submissions, and the local arrangements organizers and all others who helped to organize this Doctoral Consortium.

Waterloo, July 2005

Nancy Day
RE'05 Doctoral Consortium Chair

RE'05 Doctoral Consortium Committee

Daniel Berry, University of Waterloo, Canada

Betty H.C. Cheng, Michigan State University, USA

Nancy Day, University of Waterloo, Canada (Chair)

Orlena Gotel, Pace University, USA

Sol Greenspan, National Science Foundation, USA

Table of Contents

Development of a Requirements Engineering Method For Pervasive Services

L. Kolos-Mazuryk
CTIT, Twente University,
P.O. Box 217, 7500 AE Enschede, The Netherlands
l.kolos@ewi.utwente.nl

Abstract

We are at the edge of a paradigm shift, which will move the centre of information and communication control away from the providers to the individual end-users. A world is envisioned in which people have access to a plethora of application services, at any time and at any place. Current requirements engineering methods are not suitable for these services; therefore new approaches have to be developed. In this article we describe our ongoing work on such an approach. We present a conceptual framework of the domain and approach to the solution. Further we discuss method used for solution design and validation of obtained requirements engineering approach.

1. Introduction

A pervasive computing environment is a system that is pervasively and unobtrusively embedded in the environment, completely connected, intuitive, effortlessly portable, and constantly available [1]. The goal of my PhD project is to design a requirements engineering method for services to be provided by a pervasive computing environment. This is part of a larger project – A-MUSE, the goal of which is to design a complete method for the development and provisioning of services to be provided by a pervasive computing environment.

2. Conceptual Framework

Conceptually, finding a solution of a certain problem consists of a number of steps. Those are: problem analysis, solution design, solution validation, solution implementation and implementation evaluation. Within my Ph.D. project only the first three steps will be performed. They are discussed in sections 3, 4 and 5, respectively.

2.1 Pervasive Computing

Within the A-MUSE project, pervasive computing services are considered to have the characteristics of context-awareness, personalization, mobility and pro-activeness.

Context-aware computing is the ability of the software system to be determined by the circumstances in which it finds itself. Therefore the environment has an exceedingly powerful impact on a particular application either because the latter needs to adapt in response to changing conditions, or because it relies on resources availability of which is a subject of constant change [2].

Personalization is the ability of a service to be shaped or reshaped so as to better meet the individual needs or wants of a user [3]. In a taxonomy and psychology study of personalisation [4], the motivation for personalisation is divided into two major categories: personalisation to facilitate work and personalisation to accommodate social requirements. Services that are personalised today are rarely of category 1, but rather of category 2 [5].

Mobility is the quality of moving freely [6]. In the context of service provision mobility has two aspects: mobility of service providers and mobility of service consumers.

Proactiveness means that services do not simply act in response to their environment, but they do exhibit opportunistic, goal-directed behavior and take the initiative when appropriate [7].

Further, on the service level, a distinction is made between complex infrastructural or third-party services, such as typically offered by traditional providers, and personalized services that are composed from available services through the definition of relatively simple context conditions and service flows. The A-MUSE project explicitly aims at also supporting the latter, enabling service design and provisioning by developers, in a similar way as they are currently able to design web pages.

2.2 Requirements Engineering Research

As stated above, our work focuses on problem analysis, solution design and solution validation. Further sections describe these in more details.

The method we are developing should satisfy a number of criteria. As a part of the research these criteria will be operationalised in order to enable adequate assessment of the proposed method. First of all, our RE method should be oriented towards service development. Second, it should support relevant service properties – context-awareness, proactiveness, mobility, personalization. Third, the method should be usable by service providers. Fourth, the method should include a requirements validation part. This implies that the method should have a sufficient degree of formality. Note that we have to make a trade-off between level of formality and usability (easy-to-learn, and easy-to-use). Usually formality increases the complexity, and therefore more effort is required in order to learn and to use the method. Most of these criteria need to be operationalised in order to be adequately assessed.

3. Current RE Methods

During the initial phase of the project, a study of existing requirements reengineering methods and techniques was performed. The focus was mainly on goal-oriented [9] requirements engineering (GORE) [9, 10, 11] because this abstracts from a system's task, processes, and location, and, therefore, allows to reason about the requirements even if we do not know about the processes and tasks. In this respect we think it is more useful to our purpose than task-oriented methods, which are more bound to the current way users perform their tasks [20]. A number of goal-oriented methods were studied. KAOS [12] aims supporting the whole process of requirements engineering – starting from the definition of high-level goals which have to be achieved up to the requirements, objects and operations which are assigned to the various agents in the system. I*[13] is a framework for representing and organizing of knowledge and for supporting reasoning in the early stage requirements engineering. GBRAM [14] focuses on the initial identification and abstraction of goals from all available sources of information, regardless of the scope of the knowledge base.

All too frequently, traditional market research methods fall short in providing both the detail and the broad context. Effective market research requires more than a list of discussion areas or a broad satisfaction surveys [15]. Given the complexities in delivering services, an in-depth understanding of users' needs, expectations, and experiences is needed to identify opportunities to create and capture value, facilitate users in their daily life [16].

Quality Function Deployment (QFD) is the method used in industrial design field [17]. The strong point of the QFD is precise prioritization of the customers' interests, and quantitative analysis of the current and future product features.

Besides of the reach variety of existing approaches to the elicitation of user requirements, none of them has a complete set of techniques suitable for the development of pervasive services. According to the comparison we made in our study, every discussed method has significant disadvantages and inconveniences when applied to the service development. Due to space limitations, we will not present all of these disadvantages and inconveniences here. For example, GBRAM is a goal-centric method, where goal hierarchy is the main focus, and the rest of the concepts, such as stakeholders and agents, are auxiliary. These make RE for service-oriented systems with GBRAM rather a complicated task.

4. Solution Proposal

4.1 Approach

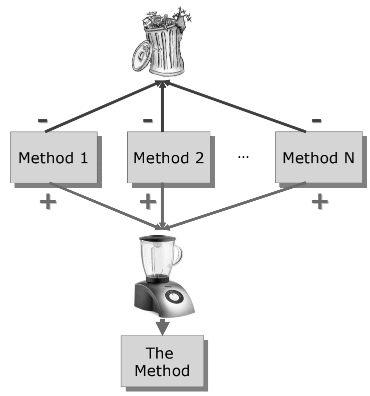

Figure 1 Solution definition

In principle, we would like to base our method on existing approaches. In order to figure out how it would be possible to combine a number of existing methods, and what are the advantages of those with respect to the services development we conducted a number of studies. In the first place, we analyzed and compared several goal-oriented methods with respect to service development. Secondly, we have performed illustrative cases with all of the studied methods in order to prove or disapprove our theoretical observations, and to gain more insight into the methods. These studies showed that existing methods do have a number of deficiencies with respect to the service development, and, generally, they do not cover all the relevant aspects of the domain.

Therefore, we would like to "blend" the useful elements of existing approaches, while getting rid of the elements not useful for pervasive service RE. This is not a simple addition of methods, since they may base on different assumptions, operate with different objects. So, the features we would like to use in our approach have to be brought to the common ground. There is a high probability that new steps and approaches have to be designed. Figure 1 shows the general scheme of the approach we've undertaken. We study a number of methods (such as GBRAM, i-star, KAOS) both theoretically and practically. As a result of these studies we obtain an indication to advantages and disadvantages of the methods for the service development. Advantageous constituents of every method serve as the building block of the new approach. Negative sides are noted, and we avoid those in the new approach. Following section describes the approach to the practical studies in more details.

4.2 Solution Design Method

Throughout the project a number of case studies will be performed. In these case studies we will elicit requirements for a number of systems that will provide mobile, context-aware, proactive and personalized services. Two big case studies are made in the healthcare domain.

There are too many variables influencing the usability of the requirements elicitation techniques. Therefore, the most trustworthy results can be obtained by practical validation of the mentioned techniques. So, we decide to take an action research approach [19] in our case studies: apply the techniques in a real project, reflect, improve techniques, apply improved techniques etc.

4.3 Case Studies

The approach to the case studies is as follows. We take a number of existing or invent new requirements elicitation techniques, which we think might be useful for the services and apply them in a certain project. The end result of these is a set of requirements. After the set of requirements for a system was built, we evaluate several things. First of all, the requirements themselves – how do they fit the initial needs of the stakeholders. Secondly, we look at the way chosen techniques were applied in the project – what did work well, what was completely not useful, what needed improvements. Based on such reflection we improve our instruments – requirements elicitation techniques, so we get an improved set of techniques, which is actually applied in the next case study. After a number of iterations, no more substantial improvements of requirements elicitation techniques would be needed. At that point we can claim that we have a requirement engineering methods suitable for service development. Validation of this claim is discussed in section 5.

4.4 The proposed RE method

The starting point of our requirements engineering method is following. We observe various stakeholders of the systems in development, and try to extract their needs with respect to the system from various sources, such as interviews, diaries, user testing, workshops, etc [20].

Understanding the user's environment is done through a number of semi-structured interviews. Information obtained from the interviews allows us to understand the daily routine of the users, and get insights into their goals that can be reached with help of the system. The next step is to build more detailed and complete model of the environment. Based on the information we gained, diaries are composed. They give us a possibility to capture all the aspects of the daily routine, which were overlooked during the interviews. Thus, based on the results of analysis, we refine our requirements model.

Besides, we are thinking of the systems that are completely new and not yet available on the market. So, we are combining diaries with a game, in which every participant is asked to draw a picture of how he/she imagines the system without any limitations at all. This allows us to see the so-called mental model of the system from participants. In our opinion, such a model can play an important role on the early stages of development. It helps to align ideas of developers with those of participants in a playful and informal way.

Another important type of activity with respect to the idea generation is a workshop. The goal of the workshop is facilitation of concept creation for future services. Workshop allows to see all the aspects not only with respect to the usage of technology, but also to obtain information about users' motivation regarding utilization of future services. All the participants are divided into groups and brainstorming is performed in each group. Afterwards every group has to select two best ideas they've generated. The groups come together and discuss these ideas. Then all the participants receive fake money, and they can spend it on everything they'll like, including the new services they proposed. This will circle out the more realistic ideas, since people tend to be more critical when financial issues are in play. These ideas further are transferred into goals, which the user aims to reach with the help of the system being developed.

5. Solution Validation

For our proposed RE method, it is necessary to validate two things. First, how adequate are the results achieved by applying our method. Second, how usable is the method for service developers. The first is done by validating user requirements obtained by our method in case studies, described below, and must be done with the end-users of services in those cases (doctors, patients, etc.). The second should, in fact, assess developers experience with the application of proposed method.

In order to validate the adequacy of obtained requirements, a workshop takes place after a concept is completely formed. All the stakeholders are invited to participate in the workshop, and a concept (or a prototype) is presented. The goal of the workshop is to minimize the contradiction among the needs and wishes of the stakeholders. As result, we obtain valuable feedback about the concept itself and we get an idea what tradeoffs between the contradictory requirements have to be considered.

To assess developers' experience in order to validate the usability of the method, ideally joint service development projects with the industrial partners should be done. We do not expect this to be possible in the Ph.D. project, and therefore a number of student projects will be carried out instead. Students will work on the same case study, but using different sets of techniques, which, in fact, form the RE method.

6. Current State of Affairs

Currently, the project is in the initial phase. A number of requirements engineering methods were studied, and the need for the new method, which focuses on the development of context-aware personalized mobile proactive service, was confirmed by the results of the study. An approach to the development of the method was proposed, and the initial set of techniques was selected.

Acknowledgements

This work is part of the Freeband A-MUSE project. Freeband (http://www.freeband.nl) is sponsored by the Dutch government under contract BSIK 03025.

References

[1] Russell, D. M., Weiser, M. “The future of integrated design of ubiquitous computing in combined real and virtual worlds”, In proceedings of CHI-98, Los Angeles, May, 1998, pp 275-276
[2] G.-C. Roman, C. Julien, J. Payton, "A Formal Treatment of Context-Awareness", in proceedings of FASE'2004, pp. 12 -36
[3] J.O’Looney, "The Future of Public-Sector Internet Service: Pt. 1, A national survey reveals the state of government participation on the web", September 2001.
[4] J. Blom, "Why do we personalise?", 4th Human Centred Technology Postgraduate Workshop, October 2000
[5] I. Jørstad, D. V. Thanh, S. Dustdar. "Personalisation of Future Mobile Services", in proceedings of the ICIN'2004, Bordeaux, France, Oct 18-21, 2004
[6] Wordreference.com dictionary. http://www.wordreference.com/definition/mobility
[7] M. J. Wooldridge, N. R. Jennings, ``Intelligent Agents: Theory and Practice", The Knowledge Engineering Review 10 (2). 1995, pp. 115-152.
[8] P.A.T. van Eck, H.M. Blanken, R.J. Wieringa. "Project GRAAL: Towards Operational Architecture Alignment", International Journal of Cooperative Information Systems, volume 13, number 3, September 2004, pp. 235-255.
[9] Z. Zlatev, P. van Eck, R. Wieringa, J. Gordijn, "Goal-Oriented RE for e-services", Proceedings of the International Workshop on Service-oriented Requirements Engineering (RE’04), Kyoto, Sept. 2004
[10] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice ", in proceedings of RE’04, 12th IEEE Joint International Requirements Engineering Conference, Kyoto, Sept. 2004, 4-8
[11] A. van Lamsweerde, “Goal-Oriented Requirements Engineering: A Guided Tour”, in proc. RE’01: 5th Intl. Symp. Req. Eng., Aug. 2001, pp. 249-263
[12] A. van Lamsweerde, "Building Formal Requirements Models for Reliable Software", Invited Paper for 6th International Conference on Reliable Software Technologies, Ada-Europe 2001, Leuven, May 14-18, 2001
[13] Eric S. K. Yu. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", in proc. RE'97, Annapolis, 1997, pp.226-235
[14] A. I. Antón, "Goal Identification and Refinement in the Specification of Software-Based Information Systems", Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, June 1997.
[15] Consumer Involvement in Relationship Marketing of Services Varki and Wong Journal of Service Research.2003, 6, pp. 83-91
[16] M. Steen, "Exploring “human centered” approaches to market research and concept development", proc. of COST269 conference, Helsinki, Finland, 2003
[17]N. Roozenburg and J. Eekels, Product design fundamentals and methods, John Wiley, June 1995
[18] D. Bolchini, J. Mylopoulos, "From Task-Oriented to Goal-Oriented Web Requirements Analysis" in proc. WISE'03. Roma, Italy. December 2003
[19] D. Avison, F. Lau, M. Myers, P. A. Nielsen, "Action Research". Communications of the ACM, January 1999/ Vol. 42 No. I, pp. 94-97
[20] Mayhew D. J. The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. Morgan Kaufmann, 1999.

Towards Semi-automation in Requirements Elicitation: mapping natural language and object-oriented concepts

K. Li, R.G.Dewar, R.J.Pooley
Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
{kl21, rick, rjp}@macs.hw.ac.uk

1. Introduction

Due to the lack of domain-specific knowledge of requirements engineers, Requirements Elicitation (RE) often involves domain experts (or customers and end-users). The difficulty of automation in supporting RE arises from this human-centered approach. Although a wide range of methods offer a better understanding and description (e.g. use case, scenarios, apprenticing [1], mind mapping [2]) to support communication, they are manual and cannot readily be executed by a computer. In our view, such communication is a complex interactive process. Engineers and domain experts do not just simply ask and answer questions. Instead they also need to respond to the answer, as which might have an influence on what to say next. Some commercial CASE tools claim to support user interaction, but generally they offer no more than an interface for data input.

The domain-specific data obtained during an interview needs to be further analyzed in terms of completeness and preciseness. Hence, requirements engineers need to meet domain experts for clarification. The number of iterations depends often depends on the experience of the parties involved.

Using appropriate modeling techniques, information is obtained and analyzed to discover the requirements; these are decomposed into system building blocks to model the static and dynamic aspects of the developed system.

The past research in object-oriented modelling has investigated heuristic-based solutions for mapping NL elements onto OO concepts. From a review of research on automatic requirements analysis, such mappings executed by NLP systems (e.g. [3][4][5][6]) only provide an incomplete coverage of the necessary notations.

For static modelling of systems, some have claimed that nouns and noun clauses can be used to derive classes and some others that Natural Language (NL) structures can be used to derive relationships between these classes [7]. However, as Halliday states "Noun names a class of things that could be concrete objects or persons, as well as abstracts, processes, relations, and states, attributes" [8]. This diverse nature of nouns makes identification of a class and its attributes extremely difficult. It is also apparent that most mappings between NL and OO concepts are based on *open-class* word; i.e. "is a type of" signifies inheritance, "belong to" and "are part of" denote aggregation [9]. From our investigations, those NL structures using *open-class* words (i.e. type, belong, part) remain very infrequent in the text. Furthermore dynamic modeling (i.e. modeling of behavior) appears even more difficult to automate.

From our investigation of past efforts in this area, we assume the premises that there is no unique one-to-one relationship between a NL element and an OO concept. For instance, "belong-to" and "are part of" both denote aggregation. In attempting to provide computer-assisted requirements elicitation, we see there is a demand for supporting requirements collection, interpretation, analysis and stakeholders interaction that would help tease out a more comprehensive system model.

2. The Proposed Approach

NL structure patterns are defined for candidate class identification. Unlike those structures mentioned early in the paper, which often are open-class words based (e.g. noun, verb etc.), we suggest that close-class words (e.g. prepositions) and Part Of Speech (POS) are perhaps a better choice in terms of reducing the alternatives in interpretation as well as improving the coverage. Our investigations investigation in NL structures show that there are some relationships between NL structures in terms of class identification. For instance, the SubjectNoun-Verb-ObjectNoun pattern could indicate association between two candidate classes (two nouns) via a method. Noun-of-Noun pattern could indicate an attribute and a class. Considering the requirements description, "*when customers order shoes, they need to specify the size of shoes. ... Following a successful payment, the ordered shoes will be sent to the postal address of the customer*". Applying the above NL patterns on the underlined words, two candidate classes "customer" and "*shoes*" can be identified. "*customer*" has a attribute "*address*" and "*shoes*" has a attribute "*size*" and a method "*order*", as well as a association relationship "*order*" between them. In summary, our class identification approach considers the whole text rather then a single sentence.

However, NL structured patterns only can provide general criteria for deriving classes and relationships. For instance, a NounSubject-verb-NounObject pattern generally identifies two classes (the subject and object) and one method (verb) of the object class. An exception to this is "*students provide their name for registration*", where the object "*name*" is an attribute of the class "*students*". To address this issue, each NL pattern is associated with number of questions to obtain further data for the missing parts of a pattern, as well as to identify the correct OO concept from alternatives. Questions are also used to determine the validity of identified classes. Candidate class validation criteria are (i) a valid candidate class must have attributes; (ii) a valid candidate class must have methods; (iii) a valid candidate class must have a relationship with another class.

In terms of implementation of the proposed approach, Fig 1 represents how the initial text-format domain-specific data can be processed towards class identification. The initial input is domain documents that are processed by the NLP component employing word tagging techniques. Each word of the input file is assigned a POS tag (e.g. Noun, Adverb, etc.) and documented in an XML format. Then, the XML file is parsed to extract text that matches the pre-defined NL structure patterns. The structured data that completely matches a pattern is stored and interpreted as OO concepts and represented in the XML Metadata Interchange (XMI) [10] format. For those parts of the text that are not thoroughly matched with any pattern, the questions corresponding to the closest matching structure pattern are used to obtain further information from users to clarify incomplete or ambiguous data.

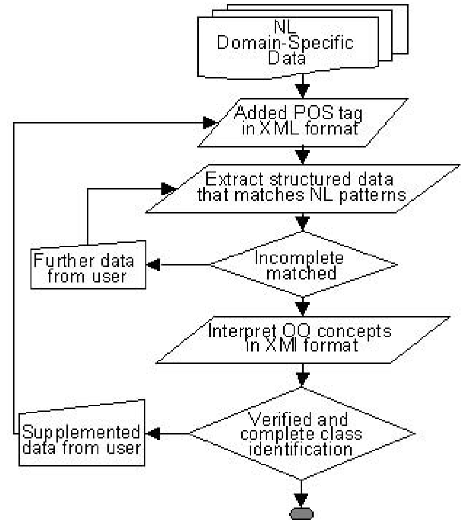

Fig 1. A Scenario of Proposed Approach

The significance of this approach is the participation of stakeholders (domain expert, customer, etc.) in the class identification process. This additional and essential data can be supplied at the earliest possible time. The use of XML makes pattern searching straightforward and XMI file construction simple. The final output of XMI offers a convenient start in object-oriented analysis/design, since most UML CASE tools can accept input class models in details of this standard format; and potentially forward engineer these models to preliminary code.

3. Planned Evaluation

Our on-going open-source development project will be used to run the proposed NL structured patterns. (The details of this system are out of the scope of this paper). The evaluation will consider the following points.

1. Correctness of class identification
2. Understandability of the pattern-associated-questions
3. Overall Efficiency

In terms of evaluating 1, case studies will compare and contrast the results generated manually and by the related work in this field. Both technical and non-technical users will be involved to use the future system in order to evaluate 2. The measure of number of questions asked and the number of iterations needed to reach a useful solution will be collected in experimental studies in order to evaluate 3.

4. Expected Contributions

We anticipate contributing NL patterns and associated questions, as well as mappings between NL and OO concepts. Furthermore, our work will facilitate requirements interpretation, domain modelling and requirements specification. In addition, we expect to assist communication among diverse stakeholders and provide a continuity mechanism so that requirements can be traced further into the development lifecycle through our use of XMI.

5. Conclusion and future work

In this paper, we discussed issues in assisting the RE activity using a computer system and provided a brief review of related work. The proposed approach, how it will be evaluated and the potential contribution are also introduced.

In terms of the future of this research, we now have a pressing need to complete the ongoing system, as well as develop a comprehensive set of NL patterns and associated questions that can allow us to construct a richer OO model from NL descriptions.

Beyond that, we see inadequacies in provision for dynamic system views. Although it is beyond of the scope of this PhD work, in the long term, we are also keen to encourage further integration of this technology to eliciting dynamic requirements.

6. References

[1] Beyer H.R., Holtzblatt K. (1995) Apprenticing With the Customer. Communications of the ACM. Vol. 38(5). pp. 45-52.

[2] Buzan T., Buzan B. (1995) *The Mind Map Book*, London: BBC Books.

[3] L. Mich, NL-OOPs: “From Natural Language to Object Oriented Using the Natural Language Processing System LOLITA.”, Natural Language Engineering, 2(2), 1996, pp. 161-187.

[4] Delisle S. Barker K. Biskri I. "Object-Oriented Analysis: Getting Help from Robust Computational Linguistic Tools." The Fourth International Conference on Applications of Natural Language to Information Systems (OCG Schriftenreihe 129), Klagenfurt, Austria, 1999, pp 167-171.

[5] H.M. Harmain and R. Gaizauskas, “CM-Builder: An Automated NL-based CASE Tool”, In Proceedings of the 15th IEEE International Conference on Automated Software Engineering (ASE'2000), 2000, pp. 45-53.

[6] J. Börstler, “User-Centered Requirements Engineering in RECORD - An Overview”, the Nordic Workshop on Programming Environment Research, Proceedings NWPER'96, Aalborg, Denmark, May 1996, pp. 149-156.

[7] R.J. Abbott, “Program Design by Informal English Descriptions”, Communications of the ACM, Nov. 26(11), 1983, pp. 882-894.

[8] M.A.K. Halliday and edited by Jonathan Webster, *On Language and Linguistics*, Continuum, New York, 2003

[9] N. Juristo and A.M. Moreno, “How to Use Linguistic Instruments for Object-Oriented Analysis”, IEEE Software, May/June 2000, pp. 80-89

[10] XMI, http://www.omg.org/technology/documents/formal/xmi.htm [last accessed 28/4/5.]

Requirements Analysis Framework for Alignment of IT with Competitive Strategy of Business Organizations

Steven J. Bleistein

Computer Science and Engineering,
University of New South Wales /
National ICT Australia
steven.bleistein@nicta.com.au

Abstract

To help ensure IT-business strategy alignment, I propose an integration of requirements engineering and business strategy analysis techniques. An integrated framework of problem diagrams and goal modeling is developed. Strategic objectives are modeled using goals with reference to business model context. VMOST analysis is used to deconstruct business strategy, and then a goal model is constructed using a strategy modeling framework proposed by the Business Rules Group. Validation will first be via application to an exemplar developed from the literature, followed by a real-world, industry case study. The expected contributions of this research are described in the final section.

1. Introduction

Strategic alignment of IT exists when a business organization's goals and activities are in harmony with the information systems that support them [1]. CIOs have consistently considered alignment of IT with business strategy a top priority [2], and such alignment has been shown to lead to superior business performance [3]. Hence, any requirements for a business organization's IT should be in alignment with its business strategy. It is thus important that requirements analysis capture both an organization's strategic business objectives and the activities and processes by which those objectives are to be achieved.

Oliver defines business strategy as "the understanding of an industry structure and dynamics, determining the organization's relative position in that industry and taking action either to change the industry's structure or the organization's position to improve organizational results" [4]. Business strategy includes both the rationale for and the means by which a business organization competes with industry rivals [5]. Strategy is often expressed in written documents, such as business plans, or can be discussed in interviews with executive stakeholders.

However, business strategy usually and unfortunately falls outside the scope of current requirements engineering approaches. Some requirements analysis approaches highlight organizational aspects of requirements such as dependency relationships among actors in a system [6] or value analysis [7]. Others focus on requirements elicitation without considering explicit and coherent statements of business strategy [8]. While these approaches address important aspects of organizational IT, they do little to help requirements engineers validate system requirements against the more abstract high-level requirements that represent the business strategy the eventual IT system is intended to support.

The objective of my research is thus to propose a requirements analysis framework for business-IT strategic alignment. The objective of my research has two parts: that such a requirements framework is (1) possible; and (2) can enable verification and validation of a business organization's IT system requirements in terms of alignment with and support for business strategy in a manner that satisfies executive stakeholders.

2. Proposed Solution

My research proposes a framework that integrates Jackson problem diagrams [9] with requirements engineering goal modeling [10]. I use a problem diagram framework to represent business strategy in a requirements engineering context. A goal model integrated into a problem diagram framework is used to represent the business objectives of strategy as well as the requirements of the system, thus enabling validation of system requirements against business strategy. The goal model and the business model context are decomposed systematically in parallel down to system-level requirements and context, as illustrated in Figure 1.

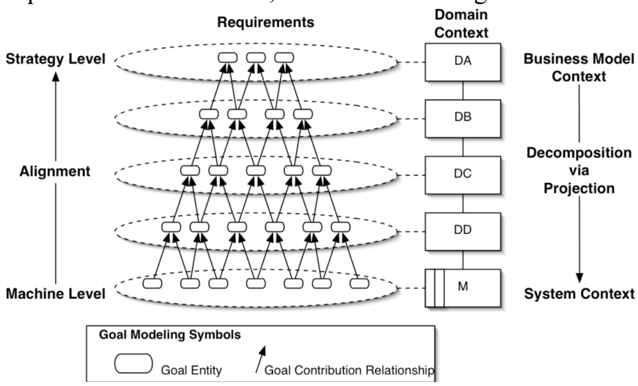

Figure 1. Integrated Goal Model – Problem Diagrams

The goal model within a problem diagram framework facilitates use of tools and techniques for business strategy modeling and analysis. These include *VMOST analysis* [11], a technique for deconstructing business strategy into core components, and the Business Rules Group's *Standard Model for Business Rules Motivation (BRG-Model)* [12], which provides rules for relating the components of business strategy to each other when constructing a model of business strategy.

A goal model of business strategy is developed first using VMOST analysis to deconstruct business strategy from abstract business plans and executive stakeholder interviews. VMOST stands for *vision, mission, objective, strategy,* and *tactic.* Based on the resulting VMOST analysis, a goal model, using a requirements engineering goal modeling notation such as *i** [6], is then constructed according to BRG-Model rules. This process is described in Figure 2.

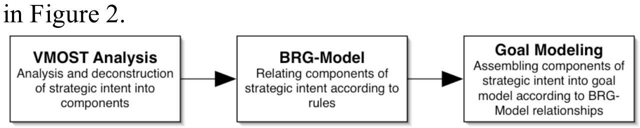

Figure 2. Process for Goal Modeling Strategic Intent

Through this process of analyzing strategy using VMOST analysis, assembling the component parts of strategy according to the BRG-Model rules, and constructing a goal model, it is possible to develop a model of the objectives of business strategy. Once integrated into the progression of problem diagram framework, it is possible to understand those objectives in their business model context at appropriate levels of abstraction. Decomposition of the business context down to system context in parallel with goal refinement down to system requirements enables complete top-down modeling from business strategy to system requirements, illustrated in Figure 1. The contribution links in the goal model enable bottom-up validation of system requirements against business strategy, or strategic alignment

3. Validation Method

I propose validation in two steps: first via an exemplar developed from the literature, followed by a real-world, industry case study.

It is common practice in software engineering research to apply a new technique to a software engineering exemplar for initial validation [13]. In doing so, it is important for the exemplar to be suited to the problem being addressed, or to demonstrating the capabilities specific to the new technique being assessed [13].

While software engineering research literature is replete with exemplars, such as the elevator, the library, the ATM, etc., the exemplars most often cited describe only small, self-contained problems. I require a well-documented exemplar of organizational IT that encompasses business strategy for my research objective; i.e., demonstrating a capability of verifying and validating requirements in terms of alignment with business strategy.

Having found no such exemplar in the software/requirements engineering literature, I chose to develop one on my own. The exemplar is based on research on Seven-Eleven Japan's IT appearing in management and information systems research literature [14-19] and business press [20]. This exemplar has enabled initial validation of my research. I have used the Seven-Eleven Japan exemplar in a number of successive iterations, each time learning from the experience and improving the approach I propose in my research. Some of these experiences appear in [21-26].

The next step in validating my research will be in real-world application as part of a collaboration project with an industry partner. I intend to publish the results of this project after its completion as a case study. At the time of this writing, I am still in discussions with several organizations.

The validation approach I have chosen, application to an exemplar I developed from the literature followed by an industry experience case study, has left me pondering a number of questions on which I wish to seek the advice and viewpoints of others in the requirements community.

Software engineering researchers use existing exemplars frequently. However, is it a valid research approach to *create* an exemplar to promote one's own research, even if that exemplar is based on real cases suited to the technique being proposed? What possible threats to validity are there in developing one's own exemplar? How can one mitigate threats to validity? Do the benefits of developing an appropriate exemplar outweigh risk of threats to validity? My view is that it is important to develop new exemplars appropriate to the research problem when none exists, but threats to validity are a concern for me.

Information systems research often uses an action research methodology in working with a collaboration partner to validate a proposed approach [27]. However, I have rarely seen reference (much less adherence) to action research methodology in the software engineering and requirements engineering research literature even though industry studies are common. Dan Berry has talked about the "Just do it!" approach to validation of requirements engineering research. What methodology is appropriate when trying out a new technique with an industry partner? Is action research methodology simply too heavy for the engineering disciplines? Are lighter-weight methodologies, such as the industry cases and experience studies that appear more frequently in the requirements research literature, more appropriate and valid? My personal preference would be for a lighter-weight approach rather than action research methodology, although some information systems researchers might argue against this.

Good collaboration partners are often very difficult to find, particularly for research problems on which the well-being or indeed the survival of a business may depend. This is particularly true in the case of engineering of IT systems used for strategic purposes. Students often need a backup plan for completing their dissertations. For a PhD thesis that addresses such problems, would a series of case studies demonstrated via exemplar taken from research literature serve as adequate validation? Should the relevance or importance of the research question justify some leeway in the rigor of the means of validation for practical reasons? Is it better to pursue lower-impact research questions that can be validated more rigorously rather than the harder, high-impact, industry problems whose validation is more difficult? What is preferable for a PhD dissertation? My personal view is that it is better to address the more difficult, industry-relevant, high-impact problems, even if the validation suffers somewhat from some lack of rigor. I find this preferable to research questions of less relevance whose primary attractiveness is the possibility of rigorous validation. While this is an ongoing debate in the information systems community [28], I have yet to hear such a debate in the software engineering community.

4. Contribution

My research makes a number of contributions to the field of requirements engineering:

1) **The integrated framework of a progression of problem diagrams with goal modeling as a means for explicit traceability and validation of requirements at multiple levels of abstraction.** This is a possible solution to what Al Davis calls the "what versus how dilemma" [29]. This dilemma refers to how different stakeholders of a system view requirements at different levels. Accomplishing the abstract, strategic business objectives for business manager stakeholders are requirements of the system just as much as the software requirement specification passed to the engineer who builds it. The framework proposed in my research integrates all these requirements in a single model at appropriate levels of context. Understanding where to place requirements in terms of level of abstraction, and how to validate them, is a significant requirements engineering problem that my research addresses.

2) **A requirements engineering approach that encompasses explicit analysis of business strategy, and an integration of business strategy analysis techniques within a requirements analysis framework.** While some requirements research has drawn on the domains of marketing and economics, such as *e^3-value* [7], mine is the first requirements engineering research to draw on competitive business strategy [21-26], and integrate recognized strategy analysis techniques [26].

3) **A means of validating system requirements against business strategy via explicit and traceable links.** This contribution is likely the result of the other contributions above, but nonetheless important to mention. In requirements engineering research, there is no discussion of IT alignment with competitive business strategy at all, despite the well-documented and evidenced importance of strategic alignment. This contribution represents the ultimate end goal of my research.

4) **The case of Seven-Eleven Japan as a requirements engineering exemplar.** The case of Seven-Eleven Japan is now available for other researchers and educators to use. Al Davis has expressed interest in using it as a teaching case, and the Seven-Eleven Japan exemplar is currently in use in graduate-level software engineering curriculum at the University of New South Wales in Sydney, Australia.

References

[1] J. D. McKeen and H. Smith, *Making IT happen: critical issues in IT management*. Chichester; Hoboken, NJ: Wiley, 2003.

[2] R. T. Watson, G. G. Kelly, R. D. Galliers, and J. Brancheau, "Key Issues in Information Systems Management: an International Perspective," *Journal of Management Information Systems*, vol. 13, pp. 91-115, 1997.

[3] A.-M. Croteau, "Organizational and technological infrastructures alignment," presented at 34th Annual Hawaii International Conference on System Sciences, 2001.

[4] R. W. Oliver, "What is Strategy, Anyway?" *Journal of Business Strategy*, pp. 7-10, 2001.

[5] M. Porter, "What is Strategy?" *Harvard Business Review*, vol. 74, pp. 61-78, 1996.

[6] E. Yu, "Modeling organizations for information systems requirements engineering," presented at IEEE International Symposium on Requirements Engineering, 1993.

[7] J. Gordijn and J. Akkermans, "Value-based requirements engineering: exploring innovative ecommerce ideas," *Requirements Engineering Journal*, vol. 8, pp. 114-135, 2003.

[8] C. Rolland and N. Prakash, "From conceptual modelling to requirements engineering," *Annals of Software Engineering*, vol. 10, pp. 151-176, 2000.

[9] M. Jackson, *Problem Frames: Analyzing and Structuring Software Development Problem*, 1st ed: Addison-Wesley Publishing Company, 2001.

[10] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," presented at 5th IEEE International Symposium on Requirements Engineering, Toronto, 2001.

[11] R. Sondhi, *Total Strategy*: Airworthy Publications International Ltd., 1999.

[12] A. B. Kolber, C. Estep, D. Hay, D. Struck, G. Lam, J. Healy, J. Hall, J. A. Zachman, K. Healy, M. Eulenberg, N. Fishman, R. Ross, T. Moriarty, and W. Selkow, "Organizing Business Plans: The Standard Model for Business Rule Motivation," The Business Rule Group November 15 2000.

[13] M. S. Feather, S. Fickas, A. Finkelstein, and A. van Lamsweerde, "Requirements and specification exemplars," *Automated Software Engineering*, vol. 4, pp. 419-438, 1997.

[14] M. Bensaou, "Seven-Eleven Japan: Managing a Networked Organization," INSEAD Euro-Asia Centre, Case Study 1997.

[15] S. Whang, C. Koshijima, H. Saito, T. Ueda, and S. V. Horne, "Seven Eleven Japan (GS18)," Stanford University Graduate School of Business 1997.

[16] W. V. Rapp, "Retailing: Ito-Yokado Seven-Eleven Japan," in *Information technology strategies: how leading firms use IT to gain an advantage*. New York: Oxford University Press, 2002, pp. 163-186.

[17] P. Weill and M. Vitale, *Place to Space: Moving to eBusiness Models*. Boston: Harvard Business School Publishing Corporation, 2001.

[18] R. Kunitomo, "Seven-Eleven is Revolutionising Grocery Distribution in Japan," *Long Range Planning*, vol. 30, pp. 887-89, 1997.

[19] N. Makino and T. Suzuki, "Convenience Stores and the Information Revolution," *Japan Echo*, vol. 44, pp. 44-49, 1997.

[20] The Economist Newspaper Limited, "Over the Counter e-Commerce; E-Strategy Brief: Seven-Eleven; Japan's Seven-Eleven," *Economist (US)*, vol. 359, pp. 77, 2001.

[21] S. Bleistein, A. Aurum, K. Cox, and P. Ray, "Linking Requirements Goal Modeling Techniques to Strategic e-Business Patterns and Best Practice," presented at Australian Workshop on Requirements Engineering (AWRE'03), Sydney, Australia, 2003.

[22] S. Bleistein, A. Aurum, K. Cox, and P. Ray, "Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach," *Journal of Research and Practice in Information Technology*, vol. 36, pp. 259-276, 2004.

[23] S. Bleistein, K. Cox, and J. Verner, "Problem Frames Approach for e-Business Systems," presented at 1st International Workshop on Advances and Applications of Problem Frames (IWAAPF) at the International Conference on Software Engineering (ICSE'04), Edinburgh, 2004.

[24] S. Bleistein, K. Cox, and J. Verner, "Modeling Business Strategy in e-Business Systems Requirements Engineering," presented at Fifth International Workshop on Conceptual Modeling Approaches for E-Business (eCOMO 2004), Shanghai, China, 2004.

[25] S. Bleistein, K. Cox, and J. Verner, "Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagrams with Goal Modeling and BPM," presented at 11th Asia Pacific Software Engineering Conference, Busan, Korea, 2004.

[26] S. Bleistein, K. Cox, and J. Verner, "Strategic Alignment in Requirements Analysis for Organizational IT: an Integrated Approach," presented at 20th ACM Symposium on Applied Computing (SAC'05), Santa Fe, NM, USA, 2005.

[27] R. Baskerville and A. T. Wood-Harper, "Diversity in information systems action research methods," *European Journal of Information Systems*, vol. 7, pp. 90-107, 1998.

[28] A. Lee, J. Liebenau, and J. DeGross, "Rigor and relevance in MIS research - Introduction (Reprinted from Information Systems and Qualitative Research, 1997)," *Mis Quarterly*, vol. 23, pp. 1-2, 1999.

[29] A. M. Davis, *Software requirements: analysis and specification*. Englewood Cliffs, N.J.: Prentice Hall, 1990.

Methodological Support for Engineering Strategic Requirements for Commercial Products using Goals

Samuel Fricker

ABB Switzerland Ltd. Corporate Research
samuel.fricker@ch.abb.com

University of Zurich, Dept. of Informatics
fricker@ifi.unizh.ch

Abstract

For commercial products, requirements engineering starts by assessing how the supplying company can compete successfully in the future market place. The resulting strategic business goals are then used to guide the development of product usage requirements, which finally are refined into functional specifications.

In the requirements engineering literature, goals have received ample consideration. Not so, however, the contextualization of goals for linking product use with product strategy. This paper proposes how goal contextualization shall be investigated to better support commercial product development.

1. Introduction

The success of product development companies depends on their abilities to identify the needs of customers and to quickly create differentiating products that meet these needs at low cost [16]. To achieve these goals, a company needs to understand its situation in the markets targeted by current and future products with its capabilities for efficiently creating new products and product features. Hence, at this early stage of requirements engineering, a new product is largely considered as a means to achieve the company's goals in the economic system the product is part of.

Today, goals are utilized to express early-phase requirements. To identify the objectives that a system under consideration should achieve, the identification, analysis, and documentation of goals is of primary importance: goals provide the rationale for requirements [17].

1.1. Goals in Requirements Engineering

Two frameworks have become dominant for goal documentation and analysis: a formal and a qualitative [17]. The formal, KAOS [6], structures goals using AND/OR goal refinement and satisfaction graph structures. The qualitative, the NFR framework [4], uses weaker positive and negative contribution links to relate goals that can only be satisficed.

Goals have been catalogued and their structure formalized. Taxonomies have been defined for operationalizing goals as actions [6], for eliciting goals from system objectives stated in natural language [3], and according to system states they generate [15]. The structure of goals has been formalized using linguistic methods [11].

Typical uses of goals have been surveyed [17, 19]. They include the acquisition and analysis of requirements. Goals are elicited by asking means-ends questions. High-level goals are decomposed until the detailed goals can be realized by individual system components. Such a goal is then called requirement. Requirements are achieved or maintained by the system components using a set of actions on objects [6].

Goals relate requirements to business problems. The i* framework uses agents with goals, beliefs, abilities, commitments, and dependencies to understand why actors engage in business processes [18]. Lightswitch models communities with goals, beliefs, and dependencies to analyze the regulation mechanisms between depending enterprises [12]. Requirements for business information systems are then derived.

Further uses of goals include dealing with conflicts [4, 6], authoring scenarios [9, 13, 14], and driving software design [4].

1.2. Limitations of Current Goal Approaches

To effectively support early-phase requirements engineering for commercial products, two research aspects are missing: the support of innovation and early product definition processes and, a prerequisite, the integration of goal models with system models.

To understand commercial product definition, it is useful to look at success criteria and development processes. The most important criteria that influence product requirements include [5]: the definition of a differentiated product that delivers unique benefits and superior value to the customer, the selection of large global markets with large growth potential and low competitive intensity, leverage of the company's core competencies, existence of adequate resources in the company, including time, money and people, and short time to market with quality of execution.

A development process may be conceptualized as follows [16]. Research and marketing are continuously engaged in an innovation process, whose results are made accessible to the company. With iterations typically lasting ½-1 year, product management selects, scopes, and analyzes ideas to define the features of future products, which finally are implemented, tested, and validated by product development.

The early phases of requirements engineering for commercial products may thus be understood as collecting and structuring ideas that result from innovation, and selecting and analyzing such ideas for the definition of product features for implementation. Such focus of research in goal-oriented requirements engineering could not be identified.

Closest are approaches that use goals in business process reengineering [12, 18] and approaches to select commercial off-the-shelf software [1]. Still, criteria relevant for commercial product success and product development are not considered, in addition to the application domain and user requirements. Consequently, companies struggle to tie their business strategies into technical product requirements, the transition always being made in an ad-hoc fashion.

A prerequisite for the successful use of goals for early-phase requirements engineering are effective methods for working with goal models. If goals are to be used for translating between the economic system and the application domain of a product, the relationships between goal models and system models need to be well understood. Today, goal models are essentially considered in isolation, however. For example, in KAOS only low-level operational goals are connected to models of agents, actions, and entities. Higher-level goals and goal refinements are not connected to system design.

Still, goals in current goal-based methods have attributes referencing system elements. The structure of these elements is of little interest, though. In addition, hierarchic structuring to describe non-trivial systems is not foreseen. Consequently, there is no comprehensive understanding of how to systematically intertwine reasoning on goals with system design.

The effect of goal isolation can be seen during goal elicitation, analysis, and validation. The abstract nature of goals makes it difficult to formulate correct and useful goals [14, 17]. Analyzing goals in isolation from a description of the system to be built is difficult and involves much guess work [2, 14]. As a consequence, goal models suffer from hidden correctness, completeness, ambiguity, and consistency problems.

The correctness of a goal model is tested with complementary concepts intuitive for stakeholders: typically with scenarios. Scenarios, however, are just one systems view – a description of functionality and collaboration. Structural views, which are captured by languages like UML [10] and Adora [8], are playing an important role in the description of a system too, but are neglected to serve here as a check for correctness.

A specification is said to be complete if it is so with respect to a set of goals [17], simply assuming the completeness of that set of goals. Also the domain-independent patterns for ensuring completeness [7] leave holes open: none could be identified to exhaustively enumerating OR-decompositions, all milestones in milestone-driven refinement, and all cases in decomposition by cases. The completeness criterion needs thus be extended to completeness with respect to a description of the system concerned [20].

It is well known that isolated terms are ambiguous. Their ambiguity may be reduced by providing a definition or by describing their use, i.e. by showing their relationships to other terms. In goal models, goal topics appear merely as attributes. By visualizing the structure of the topics, the ambiguity of goal models would be significantly reduced for the model users.

While domain-independent goal refinement patterns help eliminating formal-logic consistency problems, no effort has been made to reduce conflicting use of goal topics. The introduction of complementary views like system structure, behavior, and collaboration, may highlight formerly unrevealed consistency concerns.

The combination of goals with scenarios [9, 13, 14] and of goals with business process reengineering concepts [12, 18] are good examples of how goals may be contextualized to ease their engineering. By that it has been shown that goals are related to system design. There is, however, still a lack of systematic integration of goals with models of system structure and behavior. Chung et al confirm this weakness, but indicate that this integration may well be possible [4].

2. Thesis Objectives

Improvements are needed in goal-oriented requirements engineering for commercial products. To translate from economic models to models of product use, goals should be tightly linked to their system contexts. If goals, systems, goal refinements, and design decisions can be represented in the same model at multiple levels of abstraction, the interaction between the concepts can be used to improve a range of qualities of early product requirements.

Two kinds of systems shall be considered and integrated with goals, illustrated in Figure 1: the system of product use and the economic system. The system of product use is typically called application domain and is about the evolving structured spaces in which a product is to be installed, used, maintained, and decommissioned. The economic system is about the evolving structured spaces that reflect the situation of products, companies and markets, including their needs, capabilities, and plans to fulfill these needs.

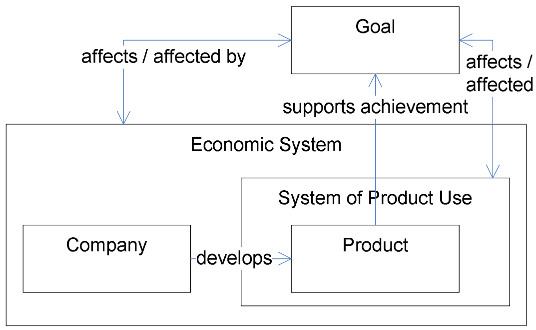

Figure 1: Goals affect the evolution of the system of product use as well as the economic system which the former system and the supplying company are part of.

Goals represent objectives to be achieved by a collaboration of systems, subsystems, and system elements, which comprise the change or maintenance of the status of such entities or properties thereof. It shall be possible to reason on goal and system models by considering the mutual influence between system design and goal refinement.

Multiple parties benefit from this research. Analysts will understand how goals are used to improve early-phase requirements engineering of commercial products. Language designers will understand how to extend languages to support goals. Tool suppliers will understand how to extend tools to support goal analysis and documentation.

2.1. An Example

For illustrating our objectives, a possible intertwining of the development of a KAOS goal model with an Adora system model is shown here. The views are complementary: goals set out design criteria and the system structure guides goal refinement.

Means to satisfy a goal "transportation request satisfied" include trains, cars, airplanes, elevators, or cranes. The solution space is greatly reduced by defining a building as the goal context. Further, the building may be defined to contain a set of elevators and passengers so that milestone-driven goal refinement may be applied to identify subgoals. The resulting models are shown in Figure 2.

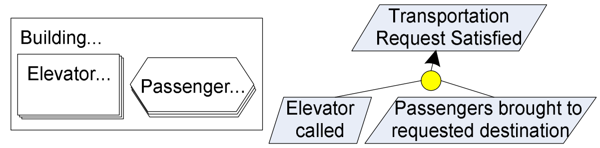

Figure 2: Intertwined Adora [8] and KAOS [6] models.

Defining how an elevator may be called requires a design change. Shafts appear that contain one elevator for passenger transportation and button-based interfaces for elevator calls. With this design we can apply milestone-driven refinement of "Elevator called", which leads to the result shown in Figure 3.

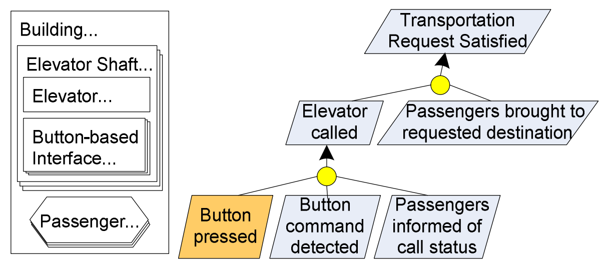

Figure 3: Further elaborated models.

Goal models and system models may be usefully combined. Goals are tightly connected to design decisions, not only at the requirements level, but also at highest goal abstraction levels. Patterns may exist to connect goals and goal refinements to system design.

Goal models may be used to structure a range of design alternatives: different goal refinements lead to different designs. Tracking and evaluating designs helps to systematically explore solution spaces.

Not shown are the visual integration of goal and system models, which is not trivial. Also excluded are economic considerations of an elevator product.

2.2. Research Problem and Constraints

A theory integrating goals with product use and product strategy shall be conceived. Methodological building blocks shall be derived for engineering goals for commercial products linking economic and usage contexts. These method fragments shall allow using goals for structuring, analyzing, and selecting commercial product ideas for implementation.

Following questions are brought up. How are goals conceptualized for commercial product strategy? How are economic systems related to systems of product use? What common concepts and principles relate reasoning on goals to system design? How can these concepts serve to building a modeling ontology? What methods are needed for commercial early-phase requirements engineering? Why are they more useful than today's approaches? And what quality aspects of requirement specifications can be improved?

To ensure applicability of the research results to industrial practice, close collaboration has been started with ABB, a global leader in power technologies and automation with many thousand products of varying complexity. To be useful to extend a range of system modeling languages, the well integrated hierarchical system modeling language Adora [8] and the widely accepted UML [10] are considered for extension.

3. Research Methodology and Validation

The research will go through the following stages. Literature of the concerned areas is reviewed, including goal-oriented requirements engineering, systems engineering, product life-cycle management, market analysis, and business strategy. From the acquired understanding, a theory of goal contextualization will be derived. To narrow the search for a fitting theory, interviews with senior practitioners and examples of product plans and requirements specifications are used. The theory is validated with an industrial case study.

In a second stage, additions to selected system modeling languages and method fragments to support early product development are defined and validated with another industrial case study. Further, experiments are made to compare goal contextualization with today's approaches.

4. Contributions and Discussion

Contributions are of theoretical, methodological, and applied nature: a validated theory explaining the relationship between reasoning on goals and system design, derived method fragments supporting early phases of commercial product engineering, case studies and experiments showing how the integrated representation of goals and systems improves a range of qualities of early product requirements, and a demonstration how system modeling languages may be extended to provide adequate tool support.

The proposed research targets important weaknesses in goal-orientation: the capability of goals for translating business considerations to a product specification, and the support of industrial innovation and product management processes.

The research is building on existing knowledge: goal modeling, system modeling, and commercial product strategy. Novel is the integration of these three subjects into one theory for goal elicitation, analysis, and validation for commercial products.

The research needs to be close to the phenomena that are considered. Access is thus required to experiences in formulating product plans, developing requirements for commercial products, and designing modeling languages. This is achieved by collaborating with ABB and with Adora language designers.

5. References

[1] C. Alves and A. Finkelstein, "Challenges in COTS-Decision Making: A Goal-Driven Requirements Engineering Perspective," *Workshop on Software Eng. Decision Support*, Ischia, Italy, 2002.

[2] A. I. Antón, "Goal-Based Requirements Analysis," *2nd Intl. Conference on Requirements Engineering (ICRE'96)*, 1996.

[3] A. I. Antón and C. Potts, "The Use of Goals to Surface Requirements for Evolving Systems," *20th Intl. Conference on Software Engineering (ICSE'98)*, 1998.

[4] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.

[5] R. G. Cooper, *Winning at New Products*, 3rd ed, Perseus Publishing, 2001.

[6] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-Directed Requirements Acquisition," *Science of Computer Programming*, vol. 20, pp. 3-50, 1996.

[7] R. Darimont and A. van Lamsweerde, "Formal Refinement Patterns for Goal-Driven Requirements Elaboration," *4th ACM Symposium on the Foundations of Software Engineering (FSE4)*, San Francisco, USA, 1996.

[8] M. Glinz, S. Berner, and S. Joos, "Object-Oriented Modeling with ADORA," *Information Systems*, vol. 27, pp. 425-444, 2002.

[9] P. Haumer, K. Pohl, and K. Weidenhaupt, "Requirements Elicitation and Validation with Real World Scenes," *IEEE Trans. on Software Engineering*, vol. 24, pp. 1036-1054, 1998.

[10] OMG, "Unified Modeling Language Specification Version 1.5," Object Management Group, 2003.

[11] N. Prat, "Goal Formalisation and Classification for Requirements Engineering," *3rd Intl. Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'97)*, Barcelona, Spain, 1997.

[12] G. Regev and A. Wegmann, "Defining Early IT System Requirements with Regulation Principles: The Lightswitch Approach," *12th IEEE Intl. Requirements Engineering Conference (RE'04)*, Kyoto, Japan, 2004.

[13] C. Rolland and C. Ben Achour, "Guiding the Construction of Textual Use Case Specifications," *Data and Knowledge Engineering Journal*, vol. 25, pp. 125-160, 1998.

[14] C. Rolland, C. Souveyet, and C. Ben Achour, "Guiding Goal Modeling Using Scenarios," *IEEE Transactions on Software Engineering*, vol. 24, pp. 1055-1071, 1998.

[15] A. Sutcliffe and N. Maiden, "Bridging the Requirements Gap: Policies, Goals and Domains," *7th Intl. Workshop on Software Specification and Design*, Redondo Beach, California, 1993.

[16] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*, 3rd ed, McGraw-Hill, 2003.

[17] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," *5th IEEE Intl. Symposium on Requirements Engineering (RE'01)*, Toronto, Canada, 2001.

[18] E. Yu, "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering," *3rd Intl. Symp. on Requirements Engineering (RE'97)*, Annapolis, Maryland, USA, 1997.

[19] E. Yu and J. Mylopoulos, "Why Goal-Oriented Requirements Engineering," *4th Intl. Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'98)*, Pisa, Italy, 1998.

[20] K. Yue, "What does it Mean to Say a Specification is Complete?" *4th International Workshop on Software Specification and Design*, Monterey, California, USA, 1987.

Managing Security IT Risk: a Goal-Based Requirements Engineering Approach

Nicolas Mayer
Public Research Centre Henri Tudor
29, av. J. F.Kennedy, L-1855 Luxembourg, Kirchberg
Nicolas.Mayer@tudor.lu

Abstract

Security is currently a major concern of Information Systems (IS) and it is generally recommended to take care of security at the early stage of IS development. Requirements engineering process seems to be a good step to handle security. In the IT security engineering domain, risk management is one of the most efficient tool, because it permits to compare security needs and costs of security measures. We propose to match some requirements engineering approaches with risk assessment approaches, to deal with IT security of an IS. The aim of this work is to provide some tools and methods to support handling of security during the first stages of a system development. A modeling framework is a cornerstone of such an approach.

1. Introduction

As well as they increase their importance in the business domain, Information Systems (IS) need currently more and more security, due to the number of attacks. Today, security is no more a desirable quality of IT systems, but a required compliance to international regulations. A number of technical answers are available in response to IT security issues. Each of these technical answers has its own level of protection and, also, its own cost. Therefore, one of the challenges is to determine the most suitable compromise between the level of security achieved and its associated cost, to obtain the best ROI (Return On Investment). This compromise should be based on the correct evaluation of the IT risk, which is usually defined by a threat and a vulnerability, with their associated potentiality, and its impact on the business assets of the organization. So it is necessary to adapt the security measures, depending on the risk and its associated components.

The analysis of risks in terms of the links existing between the business assets of an organization and the technical aspects associated with its IS, seems to be best suited for the application of a Requirements Engineering (RE) approach.

2. Problem statement

One of the main key to a good alignment between business domain and security of IT structures, is to keep the focus on the assets of the business. *Assets* are anything that has economic value for the organisation and that is central in the realization of its business objectives. Figure 1 shows different kinds of business assets in the financial domain. For example, information business assets are customers name, address and phone number. The process of account management is a core activity of a bank. Business assets are also knowledges like the ability of doing relevant economic analysis. Otherwise, we are calling IT assets those IT processes and resources of the IS and its environment, linked to the business assets. They are often considered as the "mirror" of the business assets, because many business goals are achieved with assistance of the IS. For example, IT assets are the banking application, managing customer accounts, and the customers data, stored in a database and on a server. People encoding data are also considered as IT assets, because they are part of the IS environment and essential in a good account management. IT assets are therefore the IS components (or its environmental ones) needed to be secured, in order to ensure the achievement of the business objectives.

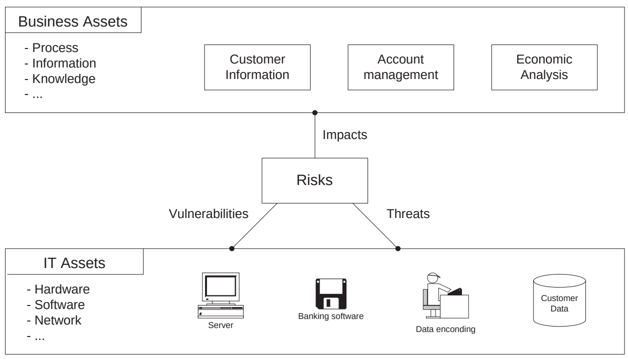

Figure 1. Risk management concepts

Assets need to be secured, because they are exposed to risks. Note that our work only focuses on risks targeting the IS, other risks like financial risks (investment) or organisational ones (hiring of a CEO) are out of the scope. Risk is most often defined by three components :

```
Risk = Threat * Vulnerability * Impact
```

In other words, risk is characterized by the opportunity of exploiting one or multiple vulnerabilities, from one or many entities, by a threatening element using an attack method, causing an impact on business assets. Figure 1 shows the links between risk components and assets. Vulnerability is a characteristic of the IT system and threat targets the IS, but the impact is reflected on the business of the organization.

A lot of work has already been done in the context of risk management and particularly risk analysis, which is the activity of analysing threat, vulnerability and impact on each component of the system. We can cite some methods based on risk analysis:

- OCTAVE [9], from the USA, developed by the Carnegie Mellon University
- MEHARI [8] from CLUSIF[1] and EBIOS [7] from DCSSI[2], two french methods
- CRAMM [11], developed in the UK

Some risk management methods are most focused on security requirements and control selection, for a standard level of protection :

- BS7799-1:1999 Information Security Management - Part 1: Code of Practice for Information Security [3] ; a british standard, also declined in the ISO 17799 norm
- IT Baseline Protection Manual [10] from BSI in Germany, even specifying security control implementation

These methods are applied in a bottom-up manner, used once the architectural design has been defined. This allows only an "a posteriori" approach of IT security, resulting in a gap between security requirements and business security needs. Our view is that an "a priori" approach of security engineering, based on risk management, could improve IT security.

3. Proposed theory

As exposed in Section 2, a lot of work has already been done in the risk management domain, particularly with industrial methods and norms. But there is a mismatch between security methods and IS system development. Our aim is to handle security in the first steps of IS design, during the RE stage.

Risk management methods are considered as semi-formal and are often a good process for a risk assessment. But the product of these methods is informal, most often in natural language. There is so a gap in automation, evolution, monitoring or traceability of risk management. The aim of the research is then to provide a layer of formalisation in the products of risk management.

3.1 Security engineering approach

The proposed approach links first business assets with security engineering. RE is the fitting domain for linking business assets, driven by business goals, with the security engineering domain (Figure 2). On the other side, architectural engineering is the domain linking IT assets, included in the IS architecture, with security engineering. The objective of security engineering is, as already explained, to mitigate risks by providing security requirements.

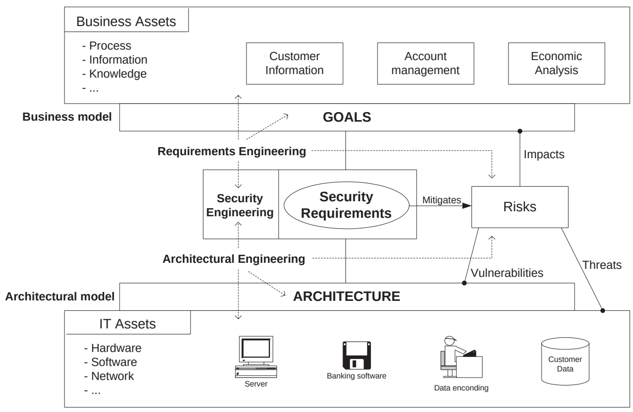

Figure 2. RE and Architectural Engineering in the IS design process

The tools used for reasoning about requirements and architectural engineering is respectively architectural and business modeling. Models provide the basis for formalisation, documentation and evolution. Our approach will be however more focused in the RE domain represented in Figure 2, i.e. doing the link between business assets overseen by business goals and security engineering used for mitigating risks. In the next section will be presented some RE approach considering security, most of them improved by modeling.

3.2 Related Work

The RE community has started to be aware of the problem of security in the last years and a lot of security RE

[1] Club de la Sécurité des Systèmes d'Information Français
[2] Direction Centrale de la Sécurité des Systèmes d'Information

approaches have been developed.

- Extensions of UML, especially of the Use Cases models, were proposed to model security aspects, such as Misuse Cases [12] and Abuse Cases [13]. CORAS UML profiles [5] are also considering security risks aspects.
- Problem Frames decomposition of Jackson deals with security, with the Abuse Frames proposal [14].
- The i* framework [17] was developed for the modeling and analysis of organizational environments and their IS. Some security applications of the framework were studied [16] and an extension for handling risks issues was developed [1].
- The KAOS approach [6] has specialized the goal analysis technique to critical system engineering (e.g. safety critical systems), which is adapted for securing critical business assets. Another goal-oriented modeling framework is the NFR framework [15], handling security as a class of non-functional requirements.

The preceding methods and techniques will be investigated and enhanced according to the research objectives. Risk management methods presented before are also naturally a source of interest for the research work.

3.3 Expected outcome

Despite many approaches were developed in the domains of risk management and security RE, few approaches integrate the two aspects. The PhD work will first pursue the objective of identifying the IT security and risk management concepts needed to enhance the RE process. It will be then necessary to identify the RE methodologies best suited to integrate the previously identified concepts, for handling IT security as early as possible.

Once all of this preliminary work is done, it is necessary to develop models and methodologies to deal with security and risk management in the early steps of an IS development. A modeling framework seems to be best suited to formalize and exploit these elements. As explained before, more than a support for analyze and reasoning, it permits some improvements like (semi-)formalisation or traceability. But it seems unnecessary to completely redesign a new modelling framework, but improving an existing one with risk management concepts should be more relevant. The focus will be more on assets identification and business modeling (i.e. the RE side of Figure 2), but the link with architectural modeling is necessary to complete the process of IS design. A study of existing security standards and references (ISO 15408, ISO 17799, NIST and CERT documents…) to extract technical and organisational requirements can also improve the method, by providing security measures to mitigate the risks, as the outcome of the security engineering process.

The development of a prototype supporting the overall approach is finally considered. Automation and deliverables produced by the process are thus a main part of the expectations and the prototype should be fulfilling them. A case study is then necessary to experience and validate the work.

The improving of such a method, apart from managing security during the first steps of software engineering, is the claim of risk management constituted by the models. Moreover, as already mentioned during the introduction, models can help system designers and managers to improve the ROI of their IS security. Despite the study will not provide some quantitative method for calculating ROI, having a clear view of assets and safeguards linked to them helps practitioners to deal with security costs.

4. Progress

This project research deals with two major scientific domains: RE and risk management. The first step was to do a state of the art of these domains. We studied the most used risk management methods (security experts estimate there is more than 200 risk management methods, so an overall study is inconceivable). The RE domain, being very large and varied, we tried to focus only on our main interests. We investigated mainly the goal-oriented, security driven or business modeling RE approaches. This work of bibliography is still in progress.

As exposed in Section 3.2, some contributions were very close to our approach, but the bibliography denotes that no current work is able to tackle every part of our problem. We are currently interested by collecting some worthwhile RE approaches and try to merge them with risk management methods.

5 Acknowledgments

Thanks to my advisors Eric Dubois and André Rifaut for their guidance in this project. The work is partially supported by the Research National Fund of Luxembourg. Part of the research is performed within the context of the LIASIT (Luxembourg International Advanced Studies in Information Technologies) Institute.

References

[1] P. Gaunard, E. Dubois: *Using Requirements Engineering Techniques for Bridging the Gap Between Risk Analysis and Security Policies*, 18th IFIP International Information Security Conference, Athens, Greece, May 2003.

[2] J. D. Moffett, B. A. Nuseibeih: *A Framework for Security Requirements Engineering*, Department of Computer Science, YCS368. University of York, UK, 2003.

[3] *BS7799-1:1999 Information Security Management - Part 1: Code of Practice for Information Security*, British Standards Institution, London, 1999.

[4] L. Chung: *Dealing with Security Requirements During the Development of Information System*, 5th International Conference on Advanced Information Systems Engineering, CAiSE'93, Paris, France, June 1993.

[5] R. Fredriksen, M. Kristiansen, B. A. Gran, K. Stølen, T. A. Opperud, T. Dimitrakos: *The CORAS framework for a model-based risk management process*, Proceedings of the 21st International Conference on Computer Safety, Reliability and Security (Safecomp 2002), LNCS 2434, pp. 94-105, Springer, 2002.

[6] A. Dardenne, A. van Lamsweerde, S. Fickas: *Goal-Directed Requirements Acquisition*, Science of Computer Programming Vol. 20, North Holland, pp. 3-50, 1993.

[7] *Expression des Besoins et Identification des Objectifs de Sécurité (EBIOS)*, Direction Centrale de la Sécurité des Systèmes d'Information (France), February 2004. http://www.ssi.gouv.fr/

[8] *Méthode Harmonisée d'Analyse de Risques (MEHARI)*, CLUSIF, Version 3, Octobre 2004. http://www.clusif.asso.fr/

[9] *Octave Critical Threat, Asset and Vulnerability Evaluation (OCTAVE)*, Carnegie Mellon - Software Engineering Institute, June 1999. http://www.cert.org/octave/

[10] *IT Baseline Protection Manual*, BSI - Germany, October 2003. http://www.bsi.bund.de/english/gshb/

[11] *CRAMM*, CCTA[3] Risk Analysis and Management Method. http://www.cramm.com/

[12] I. Alexander: *Misuse Cases Help to Elicit Non- Functional Requirements*, Position paper for Policy Workshop 1999, Bristol, U.K., November 1999.

[13] J. McDermott, C. Fox: *Using Abuse Case Models for Security Requirements Analysis*, 15th Annual Computer Security Applications Conference, Phoenix, Arizona, December 1999.

[14] L. Lin, B. Nuseibeh, D. Ince, M. Jackson: *Using Abuse Frames to Bound the Scope of Security Problems*, RE'04, Kyoto, Japan, 2004.

[15] L. Chung, B.A. Nixon, E.Yu, J. Mylopoulos: *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, Boston, 2000.

[16] L. Liu, E. Yu, J. Mylopoulos: *Analyzing Security Requirements As Relationships among Strategic Actors*, 2nd Symposium on Requirements Engineering for Information Security (SREIS), Raleigh, North Carolina, 2002.

[17] E. Yu: *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*, Proceedings of the IEEE Int. Symp. Requirements Engineering, Annapolis, Maryland, pp. 226-235, January 1997.

[3] Central Computer and Telecommunications Agency

Task-Driven Tools for Requirements Engineering

Pedro Campos

Department of Mathematics and Engineering,
University of Madeira, Portugal

E-mail: pcampos@uma.pt

Abstract

This research aims at designing and evaluating a new generation of usable and multimodal Requirements and Analysis Tools, capable of promoting artifact co-evolution in a useful manner, enabling cooperation and communication of multiple stakeholders over a common semantic model. The main goal is to leverage the elicitation of functional and non-functional requirements by using multimodal interaction techniques, and driving software development using a conceptual architecture easily extracted from user task flows.

1. Problem Statement

Requirements management and elicitation is widely recognized to be one of the major problems in modern software development. This stage of development involves multiple stakeholders, usually with different backgrounds, and is currently faced with the advent of multi-platform development [10]. In this context, new tools are required to enable cooperation and communication of multiple stakeholders over a common semantic model that is capable of driving modern software development. Lack of user involvement has traditionally been "the number one reason for project failure" and requirements engineering tools seem to have "the most significant impact on a project's success" [13].

The available Requirements Engineering (RE) tools are currently limited to modeling and management tools [14] that are hard to use and only of interest to disciplined engineers, leaving all other stakeholders (executives, marketers, clients and end-users) aside. The lack of adequate and usable tools has also been blocking technology transfer from academia to industry [14, 15, 6].

Co-authoring and co-evolution of requirements models are not adequately dealt with by current tools. Applying User-centered design to the design of new tools for promoting co-authoring and co-evolutionary development of requirements over a common semantic model could bring many benefits, such as increased stakeholder involvement and information sharing, increased traceability and usable ways to negotiate requirements as well as prioritize development tasks.

2. Prior Research

In the quest for more usable and useful requirements tools, architectures for intelligent support have been proposed [14]. These are believed to help tool developers build scalable, integrated and expanded tools. However, little attention is given to the usability aspects that need to be addressed in order to promote stakeholder involvement.

The ART-SCENE Scenario Presenter is a web-based tool aimed at providing support for scenario-based requirements engineering [12]. ARENA [7] provides negotiation methods to foster mutually satisfactory agreements between stakeholders and was based on the WinWin negotiation model [1], one of the most well-known attempts to support distributed requirements negotiation.

Recognizing the potential of mobile tools, Seyff et al. [11] proposed a tool using PDA's to foster the capture of requirements with an easier participation from stakeholders.

These examples demonstrate that there is a clear trend towards building a new generation of requirements tools, fully coupled with the whole software engineering process and fully supporting the workstyles [4] of all stakeholders in usable and useful environments.

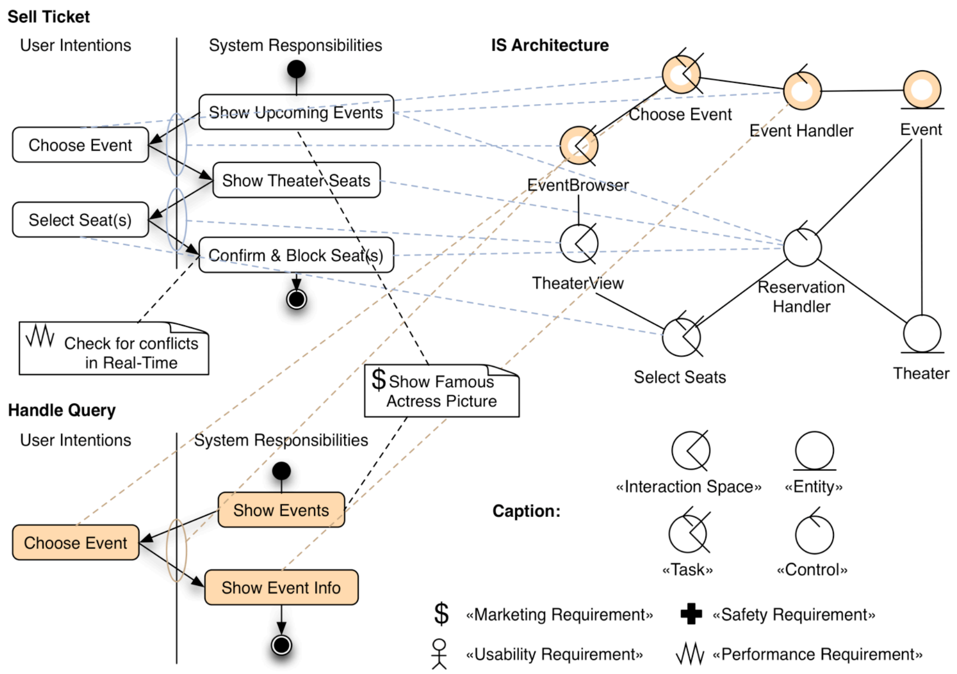

Figure 1: Tracing use cases and task activities to the conceptual architecture of a System.

3. Proposed Approach

3.1 Hypothesis and Goals

We hypothesize that (a) the structure of a system's use influences the conceptual system architecture and that influence can be used for requirements negotiation and prioritizing development tasks; (b) the UML can be successfully used as a common semantic model to facilitate communication and promote artifact co-authoring by different-background stakeholders, as long as this is supported by user-centered tools; and (c) requirements elicitation can be leveraged through multimodal, user-centered collaborative environments, not just descriptive modeling tools like the current ones.

The main goals of our research are:

- Achieve a technological solution for facilitating information sharing during requirements elicitation by a background-diverse group of stakeholders;

- Trace the requirements of a system, in terms of user intentions and system responsibilities, to the conceptual architecture of that same system, and easily extract that architecture from task flows;

- Ease the process of prioritizing development tasks as well as requirements negotiation by allowing all stakeholders to view the impact of a given set of use cases in the conceptual architecture of a system.

3.2 Method

To test our hypothesis, we propose to: (i) develop, adapt and evaluate notations from different fields (software engineering, marketing, management and usability engineering) that enable multiple representations of requirements over a common semantic model; (ii) design, implement and test a traceability model between different representations that enables synchronized artifact changes from the different stakeholders; and (iii) develop and evaluate prototype tools in the sequence of CanonSketch [2, 3], that support and demonstrate automated support for the methods, techniques and models.

Our methodology is based on Design Research [8], which has as final output an *instantiation* which "operationalizes constructs, models and methods".

TaskSketch. TaskSketch[1] focuses on linking and tracing use cases to the conceptual architecture of a system. The idea is to use the Wisdom extension to the UML [9], which can be summarized as the UML class stereotypes in the lower right part of Figure 1: «Interaction Space» models the interaction between the system and human users within the user interface of that system; «Task» models the structure of the dialogue between the user and the system in terms of meaningful and complete sets of actions required to achieve a goal; «Control» encapsulates complex derivations and calculations, such as business logic, that cannot be related to specific entity classes and «Entity» models perdurable, often persistent, information.

Figure 1 shows a simple example of two UML activity diagrams representing task flows of two distinct

[1] For more info such as screenshots and videos of the tool, please refer to http://dme.uma.pt/tasksketch

use cases in an Arts Center ticket selling IS[2]: Sell Ticket (in blue) and Handle Query (in orange). Each use case is detailed using two swim lanes: User Intentions and System Responsibilities (the tool also provides a participatory view and a narrative view). For example, in the use case "Sell Ticket", it is the system's responsibility to show upcoming events so that the user chooses one of them. The system then shows the available theater seats for that event and the user selects the desired seats, which are blocked by the IS. Each crossing of the swim lane originates an interaction space (Event Browser and Theater View). Each action on the User Intentions' swim lane corresponds to a task and each action on the System Responsibilities' swim lane is associated with a control. Figure 1 describes these relations for these two distinct use cases. Using color, the developer can look at the architectural view of the system and see which classes handle which use cases. This simple support to requirements traceability can be very powerful for, e.g., prioritizing development by deciding which classes are more urgent to implement. Figure 1 also shows an example of two non-functional requirements: a marketing requirement attached to the Show Upcoming Events action ("Show Famous Actress Picture") and a performance one ("Check for Conflicts in Real-Time"). Figure 2 shows part of the tool.

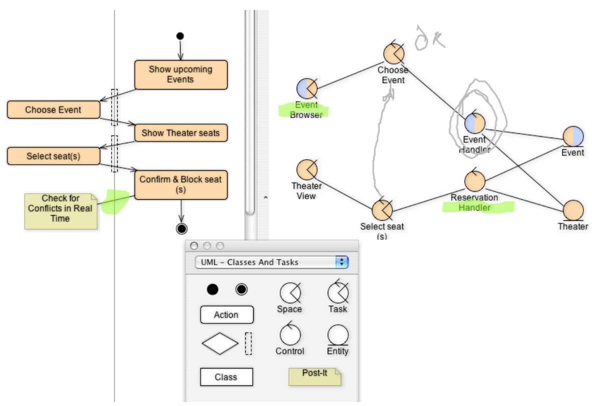

Figure 2. The TaskSketch tool.

Through easy drag-and-drop between the views, the conceptual architecture of a system can be easily extracted from the use cases' task flows.

Requirements Elicitation. We are also exploring the possibilities offered by gesture recognition, mixing formal and informal notations and collaborative development using speech recognition and a shared display.

[2] This example was provided by L. Constantine and was thoroughly used and implemented in both MSc. and BSc. HCI courses at the University of Madeira.

In this context, there is evidence [5] that real-time collaboration tools incorporating speech recognition and displaying information about a group's dynamics can positively impact the group's interaction. In some decision tasks, in particular during requirements elicitation, there is a risk that some stakeholders holding important information will not effectively share it, thus leading the team to less informed discussions.

In the "Brainstorm Environment" we propose (as part of the TaskSketch tool mentioned above), each stakeholder is associated a color and types in ideas for requirements of the system being developed. Every time someone sends a requirement to the screen, a shape color-coded by the user who sent it starts slowly falling through the center of the window.

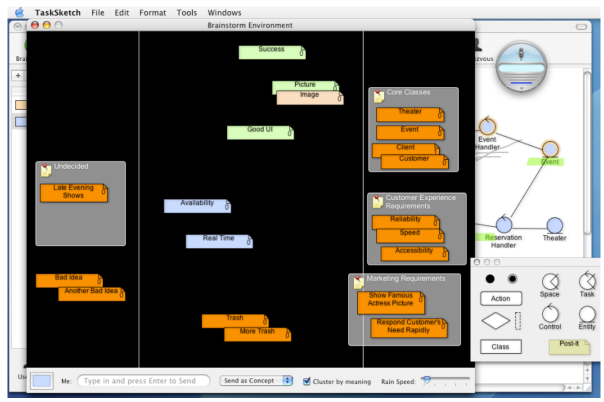

Figure 3. Collaborative Elicitation screenshot.

The content of this shared display can be manipulated by anyone, so it becomes useful to cluster concepts manually. Dragging a shape to the left or right sides of the window makes it stop falling. Concepts that remained untouched become grouped in the bottom of the window [5]. Clustering of requirements can also be made partially automatically, because this system uses a thesaurus and every time someone sends a common concept, such as "client" and "customer", the two shapes become aggregated. The speech recognition system is set to dynamically recognize any of the phrases or words in the shared display. Every time a concept is recognized, the shape shows a number, which counts the number of times that concept has been spoken during the meeting. Figure 3 shows the look of this environment.

We foresee the following usage scenario for an environment like this one: different stakeholders meet to discuss and elicit functional and non-functional requirements. This includes clients, marketers, programmers and interaction designers. Each uses a microphone and has its own color. As they suggest ideas, they watch them fall and the display becomes color-filled. In this manner, the system will attempt to increase the discussion of ideas as well as to foster collaboration between people with different backgrounds through an engaging environment. It is also expected that under-speakers will participate more and over-speakers will participate less, like [5] have shown. However, by the end of the meeting, it is also expected to achieve a better clustering and definition of concepts as well as have an idea of what requirements and concepts are more important (by looking at which words were more referred to during a given context in the meeting).

4. Current Progress and Expected Contributions

The TaskSketch tool as described here is almost completely implemented and ready for usability tests. A predecessor tool, CanonSketch [2, 3, 4] was already tested and results were promising.

Our contributions can be summarized as follows:

1. A different way to leverage requirements elicitation between different-background stakeholders, taking advantage of user-centered design techniques and informal, multimodal interaction (using speech, gestures and shared displays);
2. Innovative tools fully supporting these methods and techniques;
3. A general development framework for RE tools, based on all lesson learned during the research.

5. Evaluation

We have already conducted several usability studies during HCI courses at our University and the results seem promising. We plan to test our hypothesis through:

- Empirical validation over the usability and effectiveness of our tools and methods; this will be accomplished mainly through usability studies with different-background users; we will take measures such as user's satisfaction, error rates, expressiveness and ease of communication.
- Proofs of concept, in which we apply real-world problems (such as [10]) to our tools and methods.
- Testing and assessing the methods, techniques and tools in an industrial setting.

References

[1] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy. "Using the WinWin Spiral Model: A Case Study". *IEEE Computer*, July 1998, pp. 33-44.

[2] P. Campos and N. J. Nunes. "CanonSketch: a User-Centered Tool for Canonical Abstract Prototyping". *Proceedings of the International Workshop on Design, Specification and Verification of Interactive Systems*, Springer-Verlag, 2004.

[3] P. Campos and Nunes, N. "A UML-Based Tool for Designing User Interfaces". *UML Modeling Languages and Applications: UML 2004 Satellite Activities*, 2004, Revised Selected Papers, Springer-Verlag LNCS Vol. 3297.

[4] P. Campos and N. J. Nunes. "Galactic Dimensions: a Unifying Workstyle Model for User-Centered Design". *10th IFIP TC13 International Conference on Human-Computer Interaction*, *INTERACT 2005*, 12-16 Sep, Rome, Italy, 2005.

[5] J. M. DiMicco, A. Pandolfo, and W. Bender. "Influencing Group Participation with a Shared Display". *ACM Conference on Computer Supported Cooperative Work* (CSCW 2004). Chicago, IL.

[6] H. Kaindl, S. Brinkkemper and J. A. Bubenko Jr. "Requirements engineering and technology transfer", *Requirements Engineering*, 7(3):113-123, Sep. 2002.

[7] P. Grunbacher and P. Braunsberger. "Tool Support for Distributed Requirements Negotiation". *Cooperative Methods and Tools for Distributed Software Processes*, A. Cimitile, A. De Lucia and H. Gall, Eds., FrancoAngeli, Milano, Italy, pp. 56-66.

[8] S. March and G. Smith, G. "Design and Natural Science Research on Information Technology." *Decision Support Systems* 15 (1995): 251 - 266.

[9] N. J. Nunes and J. F. Cunha. "Wisdom: a Software Engineering Method for Small Software Development Companies". *IEEE Software*, 17(5):113-119, Sep.-Oct. 2000.

[10] L. Patrício, J. F. Cunha, R. Fisk and N. J. Nunes. "Customer Experience Requirements for Multi-platform Service Interaction: Bringing Services Marketing to the Elicitation of User Requirements". *Proceedings of the 12th IEEE International Requirements Engineering Conference*, 2004.

[11] N. Seyff, P. Grunbacher, N. Maiden, and A. Tosar. "Requirements engineering tools go mobile". In *Proceedings of 26th International Conference on Software Engineering*, ICSE 2004, pp. 713-714, 23-28 May 2004.

[12] A. G. Sutcliffe, N. A. M. Maiden, S. Minocha and D. Manuel. "Supporting Scenario-Based Requirements Engineering". *IEEE Transactions on Software Engineering*, 24(12):1072-1088, 1998.

[13] The Standish Group, *Extreme CHAOS*, Update to the CHAOS Report, http://www.standishgroup.com, 2001.

[14] Q. Zhang and Eberlein, A. "Architectural design of an intelligent requirements engineering tool", *Electrical and Computer Engineering*, IEEE CCECE 2003. Volume 2, 4-7 May 2003, pp. 1375-1378.

[15] Q. Zhang and Eberlein, A. "Deploying good practices in different requirements process models", *Proceedings of the 6th International Conference on Software Engineering and Applications*, pp. 76-80, 2002.

An Aspect-Oriented Approach to Model Requirements

Lyrene Fernandes da Silva
Computer Science Department – Catholic University of Rio de Janeiro - Brazil
{lyrene,julio}@inf.puc-rio.br

Abstract

The principles of crosscutting concern separation and composition have been used by the Aspect-Oriented Development Community in order to solve the problems of tangling and scattering. In this work we present a proposal for integrating crosscutting concerns during the requirements engineering process. This approach uses goal models and the concepts defined in aspect-oriented languages to provide separation, composition and visualization of crosscutting concerns in order to facilitate their modeling and the traceability between them.

1. Introduction

Requirements are continually changing and understanding their impact is a problem. This problem is even greater if we consider the crosscutting nature of requirements. Sometimes they influence or constrain each other, and this is known as crosscutting concerns [19].

The separation and composition of crosscutting concerns is a way of decreasing complexity and facilitating the analysis of each concern, both individually and in combination with others. These principles have been used in programming languages, by aspect-oriented languages [7]. However this level of abstraction hides many prior design decisions made without taking into account their crosscutting nature.

Research has been looking for higher abstractions related to aspects; modeling languages and methods have been proposed [4] and Requirement Approaches have mainly focused on the identification of candidate aspects [1]. In contrast to this, we are proposing a method for modeling requirements using concepts defined in aspect-oriented languages. This method involves separation, composition and visualization activities. We provide a modeling language based on goal models [13], a composition mechanism and different views of the created model. Our approach contributes to comprehension, evolution and reusability of requirement models.

The rest of this paper is organized as follows. In Section 2 we present the related work and main concepts used in our approach. In Section 3, we define a new approach to model requirements, the contributions and how we hope to validate this approach. In Section 4, we illustrate this approach with a case study. In the last Section, we present a summary and our conclusions.

2. Related work

This thesis is related to two main subjects: Requirement Modeling and Aspect-Oriented Development. We propose to use the concepts defined in aspect-oriented languages in order to reduce the difficulties related to the different characteristics of requirements and the problems in modeling and changing these requirements.

V-graph is the model used in our approach, which is a type of goal model [20]. Goal models represent the functional and non-functional requirements through decomposition trees [13]. V-graph is defined by goals, softgoals, tasks and the following decomposition relationships – contribution links (and, or, make, help, unknown, hurt, break) and correlation links (make, help, unknown, hurt, break). Each element has a Topic and a Type. The Type defines a generic functional or non-functional requirement. The Topic defines the context of that element.

V-graph was chosen because with this model we can consider requirements at three abstraction levels (softgoals, goals and tasks). This is important because in the same model we can represent reasons and operations, the context and how each element contributes to achieving the goals. Furthermore, there are important results in goal modeling concerning: how to analyze obstacles to the satisfaction of a goal [9]; how to qualitatively analyze the relationships in goal models; how to analyze variability [6]; how to analyze conflicts between goals through a propagation mechanism of labels [5]; how to identify aspects in goal models [20]; how to derive a feature model, a state model and a component model from goal models

[21]; and how to provide goal reuse [11] – this last work mentions a composition mechanism to integrate a goal model and a reusable goal model from a library.

Our method for separating and composing goal models does not change these approaches, but increases their potential. We have extended the goal models with information about how to compose them. We were influenced by aspect-oriented languages, which deal with crosscutting concerns in the implementation phase [7]. In AspectJ [8], for example, this separation is achieved by using a new element called 'aspect'. The combination is made by a component called 'weaver'. The 'weaver' processes the code, changing its elements, including the behavior or structure defined in the aspects. Similarly, we use the elements 'pointcut', 'advice' and 'intertype declaration' in order to represent how different goal models or parts of them affect each other.

It is not clear what an aspect is in the early stages of software production [19], but there are some approaches trying to provide techniques and methods for treating crosscutting concerns during the requirement process. Many of them aim to identify candidate aspects [1]. They use view points [14][15], lexical analysis [2] and catalogues of non-functional requirements [3][18]. Templates are used to describe how and where candidate aspects have an impact [12][3][15][18]. Use cases are used to represent functional requirements and the 'extend' relationship is used to represent candidate aspects [18]. Composition rules are also defined, but they are manually applied [18][2]. In Rashid's paper [15], an interesting way to automate this process using XML models is demonstrated. However, just one view is created from composition, and requirement sentences are used.

All these approaches differ from ours. First, we do not use the concept 'candidate aspect', because for us, knowing if a requirement will be an aspect in the implementation is not an issue at this point. We want to offer an easier way to model them. They may or may not be aspects in the code. The important thing is to be able to consider the scattering and tangling problems early on. Second, we want to model sets of requirements separately and offer a way to model the relationships between them. Furthermore, we want to offer different views originating from the composite model. Identifying crosscutting concerns is not our focus because they naturally appear during modeling. Crosscutting relationships are necessary, either because a requirement impacts on many points, or because it is important to keep one requirement separate from the others. Finally, we use goal models, which are an intentional view, and thus more informative representation than requirement sentences or use cases, and more user-friendly than templates.

3. Using aspects for facilitating the requirement modeling

Lemma: Using concepts of aspect-oriented languages helps to deal with the tangling and scattering problems.

Hypothesis: Considering the tangling and scattering problems early on in the process improves the manageability of the software construction process.

Some requirements scatter and tangle many others. This makes it difficult to modify the model and to perform impact analysis. In order to reduce these problems, we have defined an integration method for crosscutting concerns. The method is made up of three activities, called: separation, composition and visualization, see Figure 1.

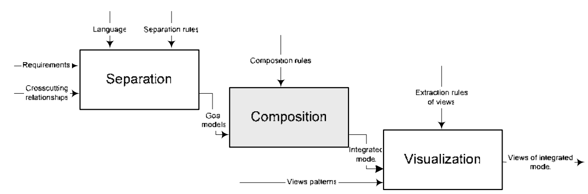

Figure 1. Integration of crosscutting concerns

The separation activity supports the requirement modeling - sets of requirements are modeled separately. In this way the complexity of modeling is reduced and the developer can consider each set of requirements more effectively. In order to model the requirements we developed a language based on the V-graph. This language is composed of a goal model specification and a crosscutting relationship specification [17]. Figure 4 shows information about the crosscutting relationship specification. We used the XML pattern to define the grammar of both models.

```
For each pointcut do {
  select advice
  for each operand do {
    if primitive = “add” then
      include advice as a sibling where operand_name = component_name
    if primitive = “include” then
      include advice as a child where operand_name = component_name
  ...
```

Figure 2. Example of composition rules

The composition activity achieves the combination of different goal models. This activity processes the crosscutting relationships creating a new goal model that contains all the original information. It uses composition rules, as shown in Figure 2. The composition activity is similar to the weaver in aspect-oriented languages. However, the weaver generates just one view of the system because computers are able to interpret (execute) complex models. In contrast to this, the visualization activity offers the developer different models or views [10]. This way, the

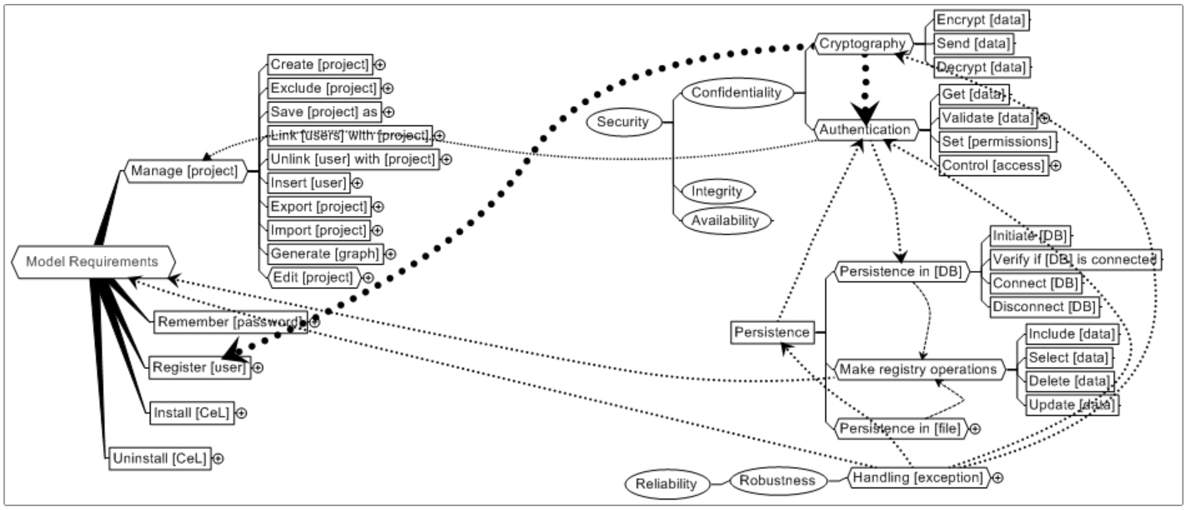

Figure 3. Separation of goal models

developer can continue elaborating the application model.

The idea is to provide requirement engineers with a way to model how the different concerns impact on each other. Therefore, while requirement engineers are modeling goal models they can concentrate on one group of requirements at a time and use crosscutting relationships to link these groups of requirements, representing the trace or impact between them. In order to be able to continue the modeling process, the engineer can obtain different views of the integrated model. This integrated model is created by an automatic composition mechanism.

3.1. Evaluation

In order to validate our approach, we are going to demonstrate our hypothesis through case studies. We will attempt to demonstrate that, through using some concepts of aspect-oriented languages for modeling requirements and providing views of the compound model, we will deal with the scattering and tangling problems during the requirement process. Therefore, we can consider, earlier on in the development process, some of the problems which may cause serious difficulties if they are only discovered during the implementation activity.

We are also implementing a set of tools to support our strategy. . Furthermore, we have modeled a set of crosscutting concerns that can be reused in different projects. Some examples are: Security, Persistence and Exception Handling. Although these examples are considered reusable, we know that each system may have a different definition for them. Therefore, our integration method helps the requirements engineer, facilitating the modeling of crosscutting concerns, the modification, and the analysis of these models.

4. Case study

This section presents an illustrative example of our approach. This example has four goal models: a goal model for an information system that helps to write scenarios and lexicon [16]; a goal model for Security; a goal model for Persistence; and a goal model for Reliability. Figure 3 shows these goal models. The ellipses are softgoals, hexagons are goals and rectangles are tasks. The pointed links are crosscutting relationships and the others are decomposition links.

Each crosscutting relationship has one or more pointcuts. Each pointcut is associated with 'advices' or 'intertype declarations'. For example, the relationship between Cryptography and Authentication (in Figure 3) has two pointcuts, called encrypt and decrypt, see Figure 4. In this example each pointcut is associated to one advice. The advices define what from Cryptography model is going to be included into Authentication model.

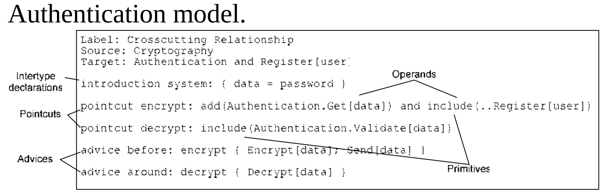

Figure 4. Crosscutting relationship

The crosscutting relationship links two elements in the same goal model and the composition mechanism processes this information creating a new goal model, see Figure 5 (one view of integrated model). In Figure 5 note the new decomposition relationships inserted into the original model. If these relationships are created manually, when changes occur, it is necessary to go through the whole model looking for where the change has had an effect. In contrast to this, in our method we can see how changes affect each part of system separately. For example, if we decide that Cryptography in the system-to-be is unnecessary, we only have to eliminate the crosscutting relationships with Cryptography.

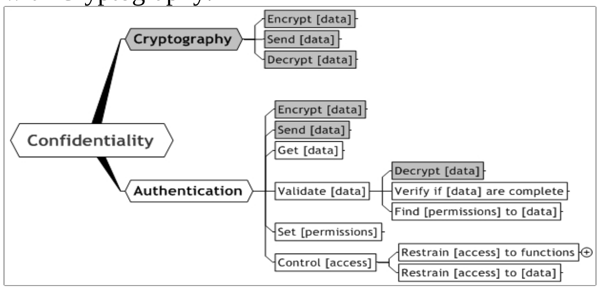

Figure 5. Composition of goal models

5. Conclusion

This thesis contributes mainly to modeling requirements, considering the tangling and scattering properties of functional and non-functional requirements. Our approach provides a new way to deal with crosscutting concerns early on in the development process. Using concepts of aspect-oriented languages, we have defined a method based on three main activities: separation, composition and visualization. “Separation” provides a language to model requirements, “composition” defines a component responsible for joining requirement models, and “visualization” makes it possible for the user to visualize different views of compound models.

Our approach improves the treatment of crosscutting concerns while defining requirements. We hope that it has a positive impact on the entire software development process. In order to implement this approach we are working on the development of a set of tools, the specification of a modeling language, the definition of composition rules, the definition of views to be extracted from goal models and on the modeling of a set of reusable crosscutting requirements.

6. References

[1] J. Bakker, B. Tekinerdogan and M. Aksit, “Characterization of early aspects approaches”, Proceedings of the Early Aspects Workshop at AOSD, 2005.

[2] E. Baniassad and S. Clarke, “Theme: An approach for aspect-oriented analysis and design”, 26th International Conference on Software Engineering (ICSE'04), Scotland, 2004, pp. 158-167.

[3] I. Brito and A. Moreira, “Integrating the NFR framework in an RE model”, Proceedings of the Early Aspects Workshop at AOSD, England, 2004.

[4] C. Chavez, “A Model-Driven approach to aspect-oriented design”, PhD Thesis, Computer Science Department, PUC-Rio, Rio de Janeiro, Brazil, 2004.

[5] P. Giorgini, J. Mylopoulos, E. Nicchiarelli and R. Sebastián, “Reasoning with goal models”, Proceedings of the 21st International Conference on Conceptual Modeling, 2002, pp. 167-181.

[6] B. Gonzáles, M. Laguna and J. Leite, “Visual variability analysis with goal models”, Proceedings of IEEE International Symposium on Requirements Engineering (RE'04), Japan, 2004, pp. 38-47.

[7] G. Kiczales et al., “Aspect-oriented programming”, Proceedings of the European Conference on Object-Oriented Programming (ECOOP’97), LNCS (1241), Springer-Verlag, Finland, 1997.

[8] G. Kiczales et al., “An overview of aspectJ”, Proceedings of the European Conference on Object-Oriented Programming (ECOOP’01), Hungary, 2001.

[9] A. Lamsweerde and E. Letier, “Handling obstacles in goal-oriented requirements engineering”, IEEE Transaction Software Engineering, 26(10):978–1005, 2000.

[10] J. Leite and P. Freeman, “Requirements Validation Through Viewpoint Resolution”, IEEE Transactions on Software Engineering: Vol. 17, N. 12, 1991, pp 1253-1269.

[11] J. Leite, Y. Yu, L. Liu, E. Yu and J. Mylopoulos, “Quality-Based Software Reuse”, Proceedings of the CAiSE 2005-LNCS 3520, 2005, pp. 535-550.

[12] A. Moreira, J. Araújo and I. Brito, “Crosscutting quality attributes for requirements engineering”, Proceeding of the 14th International Conference on Software Engineering and Knowledge Engineering, ACM Press, Italy, 2002.

[13] J. Mylopoulos, L. Chung, and B. Nixon, “Representing and using nonfunctional requirements: A process-oriented approach”, IEEE Transactions on Software Engineering, 18(6):483–497, June 1992.

[14] A. Rashid, P. Sawyer, A. Moreira and J. Araújo, “Early aspects: a model for aspect-oriented requirements engineering”, Proceedings of IEEE Joint Conference on Requirements Engineering, Germany, 2002, pp. 199-202.

[15] A. Rashid, A. Moreira and J. Araújo, “Modularization and composition of aspectual requirements”, Proceedings of the 2nd International Conference on Aspect-Oriented Software Development, ACM, 2003, pp. 11-20.

[16] L. Silva, J. Leite and K. Breitman, “Teaching Software Engineering: Report on Experiments”, Education in Computing Workshop (XII WEI), 2004 (in Portuguese).

[17] L. Silva, J. Leite, “An aspect-oriented language for requirement modeling”, Proceedings of the Requirement Engineering Workshop at CAiSE 2005, Porto-Portugal, 2005, pp. 13-25. (in Portuguese)

[18] G. Sousa, S. Soares, P. Borba and J. Castro, “Separation of crosscutting concerns from requirements to design: Adapting the use case driven approach”, Proceedings of the Early Aspects Workshop at AOSD, England, 2004.

[19] B. Tekinerdoðan, A. Moreira, J. Araújo, P. Clements, “Early aspects: aspect-oriented requirements engineering and architecture design”, Report on Early Aspects Workshop at AOSD, England, 2004.

[20] Y. Yu, J. Leite and J. Mylopoulos, “From goals to aspects: discovering aspects from requirements goal models”, Proceedings of IEEE International Symposium on Requirements Engineering (RE'04), Japan, 2004, pp. 38-47.

[21] Y. Yu, J. Mylopolous, A. Lapouchnian, S. Liaskos and J. Leite, “From stakeholder goals to high-variability software design”, Internal report, 2005.

Improving the Consistency of the Conceptual Models Through the Activity–Purpose Analysis

Davor Svetinovic
University of Waterloo, Waterloo, Canada
dsvetino@uwaterloo.ca

Abstract

This paper describes an early version of an activity–purpose analysis approach for increasing the consistency of conceptual object-oriented domain models.

1 Introduction

Starting with the first object-oriented programming (OOP) language [3], through the first object-oriented design (OOD) method [1], and the establishment of the first object-oriented analysis (OOA) method [8], object orientation (OO) has become one of the most widespread development methodologies. Despite being widely used, OO has not been extensively validated. In fact, Hatton [4], Kaindl [6], and Kramer [7] have recently indicated an urgent need for experimentation aimed at validating the effectiveness of not just OO but of all software engineering abstraction techniques and methods. This work attempts to explore and validate one particular issue — the issue of the consistency of the independently specified conceptual models.

Ever since one of the first meetings on object-oriented modeling [5, p. 74], we have known that:

> *An object-oriented model needs a formal basis that supports verifying [that] the model is complete and consistent.*

One might argue that from the requirements engineering (RE) perspective, *consistency* is one of the most important aspects of the high-quality analysis models. One particular form of *consistency* is across different groups of analysts independently analyzing the same problem domain. How consistent are the OOA models of the same problem domain produced by independent groups of analysts?

The motivation for this research project comes from our observation of students' work on the requirements analysis and specification of a system composed of (1) a small telephone exchange or a voice-over-IP telephone network and (2) the related accounts management subsystem. Production of the specification, in the form of a Software Requirements Specification (SRS) document, is the term-long project carried out in SE463/CS445, the first course of a three-course sequence of software engineering courses that spans the last year and a half of the Software Engineering undergraduate degree program at the University of Waterloo [2].

Over the last five years, we have been observing and evaluating consistency properties of the OOA models produced pursuant to these specifications. These observations motivate this thesis work.

2 Observations

Through specification reviews, interactions with students, and grading preliminary partial and the final full SRSs, we have observed many difficulties that arise throughout the specification process. The most frequently observed difficulty is that of *performing object-oriented domain analysis (OODA)*, i.e., of

1. identifying concepts of the system's domain, and
2. ascribing the system's functionality to these concepts.

The result of performing OODA is most frequently captured through the use of conceptual models.

We have observed large inconsistencies among conceptual models produced by the different groups. For the purposes of this work, we call the inability to achieve consistent conceptual models the *fundamental problem (FP)* of OODA. We have found that the FP is a result of the difficulty of performing OODA, and as such, is directly proportional to the impact of the techniques used to perform OODA.

The FP seems to occur during the analysis of all parts of the system, independent of the complexity level. In a recent case study, we have found that the FP exists *even in very small systems*. The results, background, and related work are described by Svetinovic, Berry, and Godfrey [9].

3 The Key Idea

When working with some of the groups, the author suggested that they focus on modeling processes in the domain system and in the software system before they attempt to perform conceptual decomposition. This has led to more consistent models, but with fewer concepts.

Performing full process-based decomposition of both business and software systems before attempting the conceptual decomposition of the domain system was probably too radical a change, but the core idea of attempting to derive some intermediate analysis artifacts that are more constrained than conceptual models might be worth exploring.

4 Problem Statement

The thesis work asks, "**How can we reduce the FP? That is, how can we achieve more consistent conceptual models?**"

Why is this an important question? One can argue that there are two important roles of an analysis model, and of a conceptual model in particular. The first role is to serve as a transition vehicle from the system requirements to system design artifacts. The second role is to serve as a vehicle for sharing the important domain concepts among the developers, i.e., to be used to communicate domain knowledge. Each role is tightly linked to our understanding of the domain system as captured through the gathered requirements. The consistency issue comes to play here.

In most cases, the most relevant measure of the quality of a conceptual model is the subjective evaluation of each of the team members. For such subjective evaluation, an analyst can evaluate the models with respect to only the criteria that she has used in her own decomposition; so, if the criteria vary, then the actual evaluations will vary too. As we have seen in the preliminary case study, OOA does not provide us with strict criteria to use when specifying conceptual models; thus, the quality and subjective evaluations vary widely. This study focuses on altering and specifying more precisely criteria to be used for creating conceptual models. This criteria specification will be achieved through the use of an alternative OOA approach. As such, this work will provide us with the insights into the differences observed when applying the alternative OOA approach.

Probably the most important impediment to communication of models is the difference among models created by different analysts. Sharing understanding of the system through analysis models is tightly linked to the models' use for design purposes, and inconsistent models can yield many ambiguities. As such, it is of crucial importance to have the means that will lead each analyst to as consistent model as early as possible; ideally even before they start any communication.

Why is the conceptual model so important? The typical object-oriented process advocates directly mapping the conceptual model to the class or object models. Each of these models drives, in turn, the generation of other OOA and OOD models. Such strong influence directly results in propagation of any errors and misunderstandings captured in the conceptual models to all other OOAD models. This propagation ultimately carries any negative effect all the way to the implementation.

To the best of the author's knowledge, there have been no studies conducted on the effects of the ordering of functionality and structural OOA with respect to the consistency of the conceptual models. In this thesis work, the author will perform a comparative study of the ordering effects in the two cases: (1) concepts first, functionality later and (2) functionality first, concepts later.

5 Thesis Statement

The thesis hypothesis is, "**The FP can be reduced through the activity–purpose analysis.**"

Finding ways to measure the FP reduction, benefits, drawbacks, and side-effects of modifying the traditional OOA process will be the additional contribution of this thesis work.

6 Preliminary Solution Outline

Investigating the thesis hypothesis requires an approach that allows capturing business activities and validating activity–concept matching through purpose analysis. The approach will have to allow evaluating the results taking into account the core methodological and pragmatic difference in comparing:

- the traditional approach in which one is just *observing* the concepts, and
- the modified, activity-focused, approach in which one is *looking* for a concept that is responsible for a particular activity and is then *validating* the concept through activity–purpose analysis.

The author's strong belief at this stage is that a combination of *state-based modeling* and *goal-based analysis* will be sufficient as the underlying techniques for the activity–purpose analysis. Each technique is well understood, formalized, and widely used.

Figures 1, 2, and 3 summarize this transition from the traditional OOA process to a modified, activity-focused, OOA process.

Figure 1 shows the traditional OOA approach with the relationships between the use-case model, a popular re-

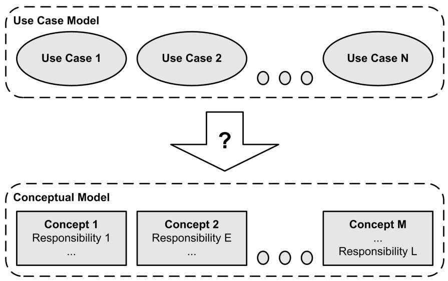

Figure 1. Traditional OOA Workflow and Model Influence

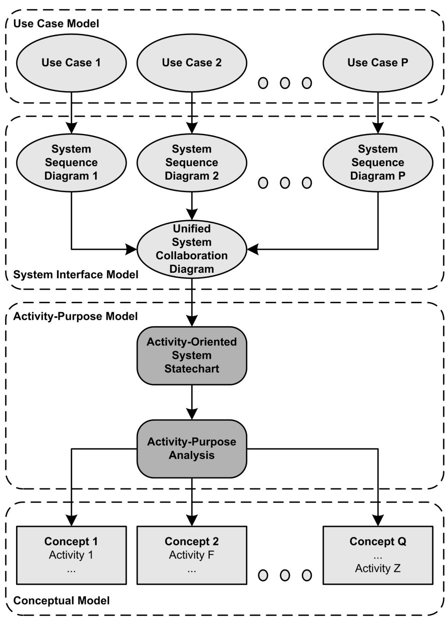

Figure 2. Modified OOA Workflow

quirements model, and the conceptual model. The traditional OOA workflow is based on discovering the concepts from the use-case descriptions, and the influence is one-directional — the use-case model influences the conceptual model. The main problem with this approach [9] is in the *criteria* used in the transition from use cases to conceptual models. The transition techniques, such as *noun extraction* and *category lists*, rely only upon our observational skills and do not provide (1) any means to validate the relevance of an extracted concept and (2) the rationale for its inclusion in the conceptual model.

In order to increase the consistency of the produced models, we have to provide analysts with means to validate the fitness of a certain concept for the inclusion in the conceptual model, as depicted in Figure 2.

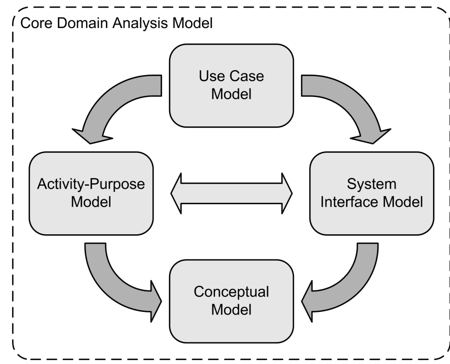

Figure 3. Modified OOA Model Influences

Figure 2 shows the modified OOA approach with two new models introduced before the conceptual model and the general workflow. The two new models are the *system interface model* and the *activity–purpose model*. The purpose of the system *interface model* is to show system's boundary and to clearly indicate all the external events captured in the system's use cases. These events will be used as the initial driving force for the transition to the *activity–purpose model* and the activity analysis.

The two main activities performed in order to build the activity–purpose model are:

- capturing and relating domain activities through the use of state-based modeling techniques and statechart notation, and
- discovering concepts responsible for performing previously discovered activities, and validating relevance of these concepts through the activity–purpose analysis.

Finally, while Figure 2 shows the theoretical workflow, Figure 3 shows the expected influences among the models. Shaded arrows indicate that we expect a strong influence among the artifacts from the different models. The arrow direction shows which model influences which other model.

As we can see in Figure 3, the *system interface model* and the *activity–purpose model* act as a buffer and a validational filter for transition from the use-case model to the conceptual model. The weaker influence between the system interface model and activity–purpose model reflects the need to iterate between the two until the system boundary and the interface are stabilized and clearly defined.

In summary, the idea of moving from an *observational* OODA to a *validational* OODA will be examined through the use of *state-based modeling* for performing early functional decomposition of the system followed by the use of modified *goal analysis* for the concept–purpose validation. The *state-based modeling* will lead to the creation of the *activity-oriented system statechart*, and the *goal analysis* will lead to the creation of *activity–purpose analysis* artifacts. The main contribution of this particular setup is that it will allow us to move from the *observational* OODA to the *validational* OODA, in which each concept is required to (1) fulfill a previously discovered activity, and (2) is validated to match its own purpose and the purpose of the particular activity for which it is responsible.

Note, this is the early outline of a possible solution and is expected to change through the detailed elaboration and the feedback obtained during the early case-study validations.

7 Research Method

The research will be carried out in six phases. Two of these phases are concerned with the theoretical foundation of activity–purpose analysis, and the other four phases are concerned with the evaluation. At each stage, we will use manual, qualitative comparison and evaluation as the assessment technique. The problem itself is not well suited for any pure quantitative and automated comparison and evaluation, which would require taking certain aspects of the system and the results out of the context. Therefore, in this study we will focus on the qualitative empirical evaluation of the results.

The informal specification phase — We will informally specify the initial approach through the refinement of the early solution described in the previous section. The main focus will be on the elaboration of the *activity–purpose model* techniques and artifacts.

The independent expert evaluation (1st evaluation phase) — The approach will be analyzed and evaluated by experts from various software engineering areas based on their experience and its applicability for the domain analysis.

The individual case-studies evaluation (2nd evaluation phase) — We will perform the analysis and evaluation of the six elevator systems using the modified approach. The results will be compared with the original specifications. The approach will be refined based on the observed results and experiences.

The formal specification phase — If necessary, we will formally specify the underlying semantics of any ambiguous component of the modified approach. For example, we might need to formally specify the semantics of the activity–purpose analysis artifacts and techniques. Also, we expect a need for a formalization of the state-based model semantics for our early requirements capture needs.

The independent expert re-evaluation (3rd evaluation phase) — Any changes to our approach that arise in the previous two phases will be re-evaluated by the same experts as in the first evaluation phase. This phase serves as an incentive for the project to stay on its track and to discourage any uncontrolled changes and side effects.

The independent group case studies (4th evaluation phase) — The approach will be independently applied and evaluated through use by different groups in the SE463/CS445 class for analysis of the medium sized system described by Svetinovic, Berry, and Godfrey [9]. The results obtained using the modified approach will be compared to the results obtained using the transitional approach in the previous terms. Again, the assessment will be performed manually and from a qualitative perspective. For example, each concept will be compared not just through its naming, but also through its semantics within the system's context and through its purpose. This evaluation cannot be automated. It must be done by a careful manual inspection of each SRS that takes into account all the artifacts from use cases to assigned responsibilities.

8 Conclusion and Contributions

From the validation of the thesis hypothesis, we will know if it is possible to achieve more consistent conceptual models by focusing explicitly on the functionality and business activities before performing the conceptual decomposition. The author expects significant increase in the consistency as compared to the traditional observational conceptual decomposition. The improvement will be evaluated through the measurement of the reduction of the FP and the analysis of the concrete benefits, drawbacks, and side-effects of applying the modified OOA approach.

References

[1] G. Booch. Object-oriented design. *Ada Lett.*, I(3):64–76, 1982.

[2] SE463/CS445 course project. `http://www.student.cs.uwaterloo.ca/~cs445/project/`; accessed January 10, 2005.

[3] O.-J. Dahl, B. Myhrhaug, and K. Nygaard. *SIMULA 67 Common Base Language*. Norwegian Computing Centre, Oslo, Norway, 1968.

[4] L. Hatton. Does OO really match the way we think? *IEEE Software*, 15(3):46–54, 1998.

[5] R. Holibaugh. Object oriented modelling. In *OOPSLA '91: Addendum to the proceedings on Object-oriented programming systems, languages, and applications*, pp. 73–78. ACM Press, 1991.

[6] H. Kaindl. Is object-oriented requirements engineering of interest? *Requirements Engineering*, 10(1):81–84, 2005.

[7] J. Kramer. Abstraction: The key to software engineering? *Keynote: JSSST Japan Society for Software Science and Technology Conference*, 2004.

[8] S. Shlaer and S. J. Mellor. *Object-oriented systems analysis: modeling the world in data*. Yourdon Press, 1988.

[9] D. Svetinovic, D. M. Berry, and M. Godfrey. Concept identification in object-oriented domain analysis: Why some students just don't get it. In *International Conference on Requirements Engineering RE'05*, 2005. To appear.

Mobile Tools for Context-Aware Requirements Elicitation and Negotiation

Norbert Seyff
Johannes Kepler University Linz
Systems Engineering & Automation
4040 Linz, Austria
ns@sea.uni-linz.ac.at

Abstract

Conventional Requirements Engineering (RE) tools that support face-to-face or distributed collaborations are typically available on desktop-based workstations. The advancing capabilities of mobile devices such as Personal Digital Assistants or Smartphones enable us to envision mobile RE tools that support context-aware requirements elicitation and negotiation. This paper reports our ongoing research in designing, implementing, and evaluating mobile RE tools. It describes the pursued research approach, presents results achieved so far, and discusses open issues.

1. Introduction and Motivation

Requirements elicitation and negotiation are collaborative processes in which different stakeholders communicate their needs, expertise, and experiences. Communication and knowledge transfer are essential for defining requirements representing the needs of success-critical stakeholders. Often, requirements are gathered in face-to-face meetings involving all success-critical stakeholders. This approach fosters collaboration and direct interaction among stakeholders. At the same time meetings have several limitations: The time constraints of all participants are often hard to reconcile, meeting time is limited and key stakeholders are unable to attend. The group size is limited by the meeting room capabilities. The costs of the equipment and the required analyst or facilitator have to be considered.

Due to the increasing distribution of development activities and the limitations of conventional face-to-face meetings new approaches are explored [1]. For example, web-based requirements engineering tools enable distributed stakeholders to communicate and negotiate their goals.

Such RE tools help to overcome some of the limitations of face-to-face meetings, but distributed collaboration also poses new problems. The absence of an analyst or facilitator might influence the process as there is only limited human guidance - the tools have to compensate this. The lack of face-to-face collaboration might affect the negotiation process; once again the tools have to cater for this limitation.

A significant limitation of both face-to-face and distributed collaboration settings is that stakeholders discuss and negotiate their expectations out of the work context of future system users. Schmidt *et al.* [2] define context as a three-dimensional space: self (physiological, cognitive), environment (physical, social), and activity (behavior, task). The context of a stakeholder therefore influences her requirements. In conventional RE meetings stakeholders typically leave their known environments to discuss their needs outside their work context. This can result in incomplete and incorrect requirements. For example, an air traffic controller who is able to manage complex emergency scenarios (activity-dimension) under great personal pressure (self-dimension) might not be able to describe these requirements for a new system correctly when he is out of this particular context [3].

Also, analysts or facilitators who are guiding requirements elicitations or negotiations often have hardly any knowledge of the stakeholders' work environment which can negatively affect the elicitation process.

In many cases only representative key stakeholders are invited to participate in a requirement elicitation session. These representative key stakeholders have to make decisions for a whole group of peers, without any possibility to discuss important issues with colleagues (e.g., only one key developer is invited to the negotiation process and has to decide for all the developers within the project).

2. Research Hypothesis and Approach

In aiming to overcome some of these problems the focus of the ongoing research is to develop and evaluate mobile RE tools supporting context-aware requirements elicitation and negotiation.

The author's hypothesis is that existing requirements engineering approaches can be complemented by using mobile context-aware tools directly in the workplace of future system users. We expect that such an approach will lead to more correct, consistent, and complete requirements.

The research is organized as follows:

- Analysis and selection of mature and proven RE approaches and tools;
- Identification of potential usage-scenarios for mobile RE tools;
- Design and development of mobile RE tools based on these approaches and usage-scenarios;
- Concept and usability evaluations of the tools;
- Improvement of the tools to address experiences and lessons learned;
- Analysis of different usage-scenarios of the tools;
- Empirical evaluation of the mobile tools.

3. Results and Status

Our first step in developing mobile RE tools was the *analysis of existing requirements engineering approaches and tools* resulting in the selection of ART-SCENE (ART-SCENE Scenario Presenter tool) and EasyWinWin (ARENA tool):

ART-SCENE (Analyzing Requirements Trade-offs: Scenario Evaluations) is a scenario-driven technique for discovering, acquiring, and analyzing requirements [4]. It automatically generates scenarios from use case specifications. A scenario includes normal course events. ART-SCENE automatically generates alternative course events ('what-if' questions) for each normal course event. The ART-SCENE environment includes several specialized tools.

The ART-SCENE Scenario Presenter is a web-based tool which provides scenario walkthrough capabilities. Walking through scenarios allows stakeholders to elicit new requirements and comments. The automatically generated 'what-if' questions stimulate the stakeholders to think about abnormal and unusual system behavior (e.g., 'What if the end of this action is delayed?'). The ART-SCENE Scenario Presenter is usually used in face-to-face workshops [4], where an analyst is guiding the stakeholders. However, the tool can support a group of distributed stakeholders, too.

EasyWinWin (EWW) is based on the WinWin negotiation model [5]. It enables a set of success-critical stakeholders to negotiate their requirements by following a step-by-step negotiation process [6]. Stakeholders collect, elaborate, prioritize, and agree on requirements.

ARENA (Anytime, Anywhere REquirements Negotiation Aids) [7] is a web-based tool adopting the EWW approach. It supports a team of key stakeholders in eliciting and negotiating their requirements. ARENA is designed to support distributed and asynchronous collaboration but it can also be used for face-to-face negotiations.

Based on our analysis we *identified potential usage-scenarios for mobile RE tools*. Our goal was to combine the strengths of the existing approaches with the frequently advertised benefits of mobile devices. For example, the features of Personal Digital Assistants (PDAs) include: anytime/anywhere computing; connectivity/ browser support; or different input modes such as audio recording, drawing, handwriting, and typing. As reported in [8] we proposed the idea of mobile analysts working directly in the work environment of future system users [9]. Another reported scenario was using mobile tools to support stakeholders in negotiating requirements anytime and anywhere.

While *designing and developing the mobile tools* we also had to overcome existing limitations of mobile devices such as screen size, performance, accumulator capacity, network bandwidth, or storage capability:

The Mobile Scenario Presenter (MSP) has been developed in cooperation with City University London. The tool (shown in Figure 1) is designed to support mobile analysts and stakeholders to acquire requirements systematically and in situ by using structured scenarios. Besides capturing requirements as plain text the Mobile Scenario Presenter allows to attach audio notes and drawings to requirements.

ARENA-M (Anytime, Anywhere REquirements Negotiation Assistant - Mobile) is based on the EasyWinWin approach and allows mobile stakeholders and facilitators to negotiate requirements directly on-site. Stakeholders can express their goals and expectations by using an electronic brainstorming tool. Electronic polling is used to prioritize the requirements in situ. ARENA-M visualizes the categorized polling results and idetifies situations where a team disagrees. Stakeholders then identify arising issues and covering options for each requirement. Finally the team agrees on the requirements. The ARENA-M tool highlights new user entries and makes them immediately available to the whole team.

Figure 1: MSP screen shots showing a scenario's normal course (left) and alternative course (right)

Our next research milestone was the *concept and usability evaluation of both tools*. Two studies were carried out to evaluate the MSP. A third study was done to evaluate ARENA-M compared to the desktop-based ARENA system. In this paper we focus on the most significant results, a description of the results and the design of the studies can be found in [10].

The first study was carried out at the ICSE 2004 conference where 7 ICSE delegates used the MSP to walk-through a pre-defined scenario. While sitting in a session the experts identified new requirements for a conference presentation support system. Using the Mobile Scenario Presenter all 7 delegates were able to walk through the scenario and to define new requirements. Some delegates had problems with navigation errors which sometimes required our assistance. However, the overall feedback was really encouraging. The MSP fulfilled the expectations and the experts managed to gather requirements in a particular context.

Motivated by the successful concept evaluation of the MSP we performed usability evaluations, this time also for ARENA-M. After a short introduction to the respective tool the evaluation participants (8 to 10 people, most of them students of the Johannes Kepler University) worked with the tools and completed a questionnaire based on ISO/IEC 9241.

We used a scale from 1 (lowest value) to 7 (highest value) to structure and to aggregate the results. Table 1 reports the aggregated results from comparing ARENA to ARENA-M.

Although the overall results were satisfying participants reported problems with the data input using the PDA's pen control. To our surprise participants mentioned no significant problems concerning the limited size of the PDA's screen (320x240 pixels). For both mobile tools, participants reported fault tolerance problems. This was partly caused by temporary problems with the Wi-Fi network which required participants to re-login. One major difference between the desktop-based tools and mobile tools is system performance: while the average rating for ARENA was 6.6, the average result for ARENA-M was just 4.4.

Besides these studies we performed another evaluation study for ARENA-M done in cooperation with Vienna University of Technology. Furthermore, the City University of London is currently analyzing the results of joint evaluation studies on the MSP.

Table 1: Usability results for ARENA (-M)

ISO/IEC 9241 dimensions	ARENA	ARENA-M
Task adequacy	5.8	5.1
Self-descriptiveness	3.9	3.7
Controllability	5.4	5.3
Learnability	5.7	5.4
Conformity to user expectations	5.7	4.7

4. Open Issues

Based on the evaluation results and lessons learned we are currently *improving and enhancing the MSP and ARENA-M*. One major goal is to provide alternative input techniques to simplify data input. For instance, wizards for adding requirements - which provide word completion features - will help to overcome the reported input problems. Another focus is on more sophisticated help functions such as tool tips or tutorials.

Besides evolving the tools we are currently *elaborating optimal conditions for using the mobile tools*. We distinguish between out-of-context and context-aware settings: For example, mobile RE tools as well as distributed tools can be used to involve absent stakeholders in ongoing face-to-face meetings. Using mobile tools provides high flexibility as they can be used anytime and anywhere.

Although these scenarios make it easier for stakeholders to participate in face-to-face or distributed RE processes, using the MSP context-aware is certainly the most promising option. For instance, the MSP can be used by the analyst to directly undertake scenario walkthroughs in the work context of future system users.

Using the MSP analysts have the possibility to observe current system behavior and stakeholders interactions at the same time. The MSP's what-if capabilities enable the analyst to ask questions about abnormal and unusual behaviour in situ, thus leading to more complete and correct requirements. The tool can be also used by stakeholders to elicit new requirements in their work context. Once more, the MSP's what-if capabilities enable stakeholders to find requirements candidates. These requirements candidates are discussed with an analyst in due course.

Another issue is to elaborate usage-scenarios which combine context-aware and conventional approaches. For instance, EasyWinWin based requirement negotiations could probably start with an initial face-to-face kick-off meeting, where a limited group of key stakeholders agrees on an initial set of high-level requirements. Selected stakeholders refine these requirements by directly observing future system users in their work environment. Finally, a larger group of stakeholders (e.g., selected end users) is invited to re-negotiate these results in a distributed negotiation.

After the definition of promising context-aware usage-scenarios we will finally carry out additional *empirical evaluation studies* using the mobile RE tools. These studies will analyze the quality of the acquired requirements in detail.

5. Conclusion and Further Work

This paper presented our research in the area of mobile RE tools. The first steps to realize our vision have been the analysis and selection of existing RE approaches and tools and the development of mobile RE tools complementing them. Furthermore, we carried out concept and usability evaluation studies. We are currently discussing context-aware usage-scenarios which are important for defining final empirical evaluation studies. The author believes that mobile RE tools will complement rather than replace traditional approaches and that the combination of context-aware and conventional elicitation and negotiation approaches has the potential to improve the quality of requirements.

6. Acknowledgements

The author would like to thank Prof. Neil Maiden from the Centre for HCI Design, City University London for ongoing discussions and support for this research. This research is also funded, in part, by a grant from the British Council and the UK EPSRC's GRN10707 SIMP project.

7. References

[1] Damian, D., Challenges in Requirements Engineering. 2000, Calgary: Department of Computer Science, The University of Calgary.

[2] Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., and Van de Velde, W. Advanced Interaction in Context. in Proc. of First International Symposium on Handheld and Ubiquitous Computing (HUC99). 1999. Karlsruhe, Germany: Springer-Verlag.

[3] Viller, S. and Sommerville, I. Social Analysis in the Requirements Engineering Process: From Ethnography to Method. in Proc. IEEE International Symposium on Requirements Engineering (RE '99). 1999. Limerick, Ireland: IEEE Computer Society Press.

[4] Maiden, N.A.M., Systematic Scenario Walkthroughs with ART-SCENE, in Scenarios, Stories and Use Cases, Alexander, I. and Maiden, N.A.M., Editors. 2004, to be publishes by John Wiley.

[5] Boehm, B.W., Bose, P., Horowitz, E., and Lee, M.J. Software requirements negotiation and renegotiation aids. in Proceedings of the 17th international conference on Software engineering. 1995. Seattle, Washington, United States.

[6] Grünbacher, P. and Boehm, B.W. EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation. in Proceedings Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9). 2001. Vienna, Austria.

[7] Grünbacher, P. and Braunsberger, P., Tool Support for Distributed Requirements Negotiation, in Cooperative methods and tools for distributed software processes, Cimititle, A., De Lucia, A., and Gall, H., Editors. 2003, FrancoAngeli: Milano, Italy. p. 56-66.

[8] Seyff, N., Grünbacher, P., Maiden, N.A.M., and Tosar, A. RE Tools Go Mobile. in Proceedings of the 26th IEEE International Conference on Software Engineering (Research Demo). 2004: IEEE Computer Society Press.

[9] Beyer, H. and Holtblatt, K., Contextual Design: Defining Consumer-Centered Systems. 1998: Morgan-Kauffman.

[10] Seyff, N., Hoyer, C., Kroiher, E., and Grünbacher, P. Enhancing GSS-based Requirements Negotiation with Distributed and Mobile Tools. in to appear: Proceedings 3rd International Workshop on Distributed and Mobile Collaboration (DMC 2005), within 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2005). 2005: IEEE Computer Society Press.