Guest Editorial

Situational Requirements Engineering Processes: Reflecting on Method Engineering and Requirements Practice

Pär J Ågerfalk1* and Jolita Ralytė2
1 Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland; and Methodology Exploration Lab, Department of Informatics (ESI), Örebro University, SE-701 82 Örebro, Sweden
2 Centre Universitaire d’Informatique, Université de Genève, 24 rue du Général Dufour, CH-1211 Genève 4, Switzerland

KEY WORDS: method engineering; software process improvement; requirements engineering

1. INTRODUCTION

Software engineering situations vary considerably from project to project. As a consequence, a number of different situation-specific factors significantly influence the requirements engineering (RE) process. These factors include project objective, application domain, features of the product to be developed, stakeholders involved, and various technological conditions and constraints. At the same time, published process models and software specification and design methods are, by their very nature, general descriptions applicable in a wide range of situations. Consequently, projects require tailored methods and tools to support the actual RE process at hand.

The need for process (or method) tailoring has been long acknowledged in the software process and method engineering domains (Basili and Rombach 1987, Kumar and Wellke 1992, Rolland et al. 1999, Fitzgerald et al. 2003) and RE is certainly no exception (Davis 1982, Sawyer et al. 1997, Firesmith 2002). Although some attention has been paid to domain-specific RE processes, the more general question of how best to construct, select, and adapt RE methods for specific projects remains largely unexplored.

In recognition of this, the IFIP WG8.1 Task Group on Method Engineering1 organized the First International Workshop on Situational Requirements Engineering Processes: Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes (SREP’05) in conjunction with the Thirteenth IEEE Requirements Engineering Conference (RE’05) in Paris, France.

*Correspondence to: Pär J Ågerfalk, Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland; and Methodology Exploration Lab, Department of Informatics (ESI), Örebro University, SE-701 82 Örebro, Sweden
†E-mail: par.agerfalk@ul.ie

Copyright © 2006 John Wiley & Sons, Ltd.
2005. This special issue comprises a selection of revised and extended papers from that workshop.

Several of the articles in this special issue are founded in the tradition of method engineering. Before introducing the articles, we will therefore set the scene by elaborating how method engineering can contribute to software process practice in general, and to RE practice in particular.

2. METHOD ENGINEERING AND THE SOFTWARE PROCESS

Method (or methodology) engineering (ME) and software process improvement (SPI) are two different but related approaches aiming for more effective and efficient information systems and software engineering practices. Although there is no agreed clear-cut distinction between ‘method’ and ‘process’ (Serour and Henderson-Sellers 2004), most SPI initiatives focus on the actual development process and on empirical metrics while ME approaches emphasize coherent and cohesive method representations as a basis for tailoring situation-specific processes. SPI approaches often centre on maturity models such as CMMI and on improvement methods such as IDEAL (Cattaneo et al. 2001). ME, on the other hand, is generally understood as a way of ‘applying the discipline of engineering to design, construct and adapt methods, techniques and tools for the development of information systems’ (Brinkkemper 1996, p. 276). Interestingly, SPI tends to focus on assessment, with little guidance on how to actually improve the software process (Henderson-Sellers and Serour 2005). As pointed out by Dömges and Pohl (1998), continuous process improvement requires four practices to be in place: defining explicitly the methods used, letting these method definitions guide process performance, recording process execution experience, and feeding this experience back to the method definitions. The focus of much ME research has been on supporting incremental, model-based definition of project-specific methods for information systems and software engineering. In this sense, the role of ME is to support the proper software process guidance and incorporation of experience to facilitate improved guidance for future efforts.

Traditionally, the way to proceed has been to describe methods and processes in metamodels to be used as a basis for computer-supported instantiation of situational methods, typically through the assembly of a number of ‘method fragments’ from different methods stored in a ‘method base’ (Lyytinen and Welke 1999, Brinkkemper et al. 1999). Recently, however, ME research and practice have extended beyond the traditional assembly-based approach to address a variety of issues including method requirements specification (Gupta and Prakash 2001), method configuration, (Karlsson and Ågerfalk 2004) and roadmap-driven approaches (Ralyté et al. 2003, Mirbel and Ralyté 2006). Recent works (Rossi et al. 2004, Lundell and Lings 2004, Ågerfalk and Fitzgerald 2006, Karlsson and Wistrand 2006, Mirbel and Ralyté 2006) also pay more attention to ‘method rationale’ (i.e. the reasons behind and arguments for the method) and the tension between method-in-concept (as described in method handbooks) and method-in-action (as enacted in actual engineering practice), which seem to provide an important link between ME and SPI.

As pointed out by Nuseibeh and Easterbrook (2000), RE is a multi-disciplinary activity in which a variety of techniques and tools are used throughout development in many different application domains. Given this dynamics, methods are a means to combine different techniques and notations, and ‘method engineering ... plays an important role in designing the RE process to be deployed for a particular problem or domain’. (Nuseibeh and Easterbrook 2000, p. 43). Methods and ME thus provide heuristics and guidelines for the requirements engineer to facilitate the selection and combination of appropriate notations and modelling techniques required in different domains and at different stages of the RE process. In the next section, we will introduce the specific approaches to this end that are included in this special issue.

3. ARTICLES IN THIS SPECIAL ISSUE

In the first article of this special issue, Chad Coulin, Didar Zowghi, and Abd-El-Kader Sahraoui present A situational method engineering approach to requirements elicitation workshops in the software development process. Requirements elicitation is a complex activity, which is critical to the success of any software development project. The article
shows how ME principles can be used to create a lightweight situational approach to the early phases of requirements elicitation. Specifically, the authors show how the OPEN process framework can be tailored to this end.

The second article is *Shifts in foci and priorities: different relevance of requirements to changing goals yields conflicting prioritizations and is viewpoint-dependent*. This article by Johan Hoorn, Mark Breuker, and Evelien Kok brings attention to what the authors term the ‘requirements-analysis rift’. This rift suggests that stakeholders are inclined to accept requirements even when they do not match their personal goals. The reason is that people tend to regard requirements as being business related and do not feel a personal commitment to them. Through three empirical studies, the article shows that the key to successful RE is grounding requirements in goals and paying attention to different viewpoints on requirements, goals, and their relationships.

The third article is concerned with the application of RE methods to the analysis of civil aviation security standards. The article, *Adopting a situational requirements engineering approach for the analysis of civil aviation security standards*, is authored by Régine Laleau, Sylvie Vignes, Yves Ledru, Michel Lemoine, Didier Bert, Véronique Donzeau-Gouge, Catherine Dubois, and Fabien Peureux. The authors show how they have used situational ME principles to engineer an RE method suited for this timely and, in many ways, critical domain.

The fourth article, by Toshihiko Tsumaki and Tetsuo Tamai, presents a *Framework for matching requirements elicitation techniques to project characteristics*. This practical framework is used to characterize different RE techniques as a basis for selecting those appropriate in a particular situation. An interesting and useful property of the framework is that it acknowledges that changes to an ongoing project as well as difficulties experienced in gathering requirements must reflect on the selection of RE techniques. It thus highlights that situational ME in the domain of RE is inherently iterative.

In the fifth article, Inge van de Weerd, Sjaak Brinkkemper, Jurriaan Souer, and Johan Versendaal set out to improve the development of web-based content management systems through a method engineering approach. The article, *A situational implementation method for web-based content management system-applications: method engineering and validation in practice*, presents a novel approach to developing this type of system and evaluates its usefulness in two case studies. Focussing on RE, the authors show the practical value of a contemporary approach to assembly-based situational ME.

In the sixth and final article, Mauri Leppänen presents a thorough overview of current work in ME and assesses the usefulness of the various approaches published to date. This *Conceptual evaluation of methods for engineering situational ISD methods* will provide much useful advice for anyone wanting to leverage ME as a way of improving their RE and software processes.

4. CONCLUSION

This special issue provides some insight into an evolving field of research, highlighting the need for systematic treatment of RE process tailoring. Although the concepts and studies covered are largely presented as specific to RE, general lessons for method and process tailoring and method engineering are certainly evident. We hope that the issue makes interesting reading!

ACKNOWLEDGEMENTS

We would like to extend our appreciation to the authors and to all the referees that worked with us on this special issue: Pekka Abrahamsson, Cesar Gonzalez-Perez, Fredrik Karlsson, John Krogstie, Kalle Lyytinen, Neil Maiden, Isabelle Mirbel, Norah Power, Ita Richardson, Colette Rolland, Matti Rossi, Motoshi Saeki, Camille Salinesi, Juha-Pekka Tolvonen. Thanks also to John Sturdy for proofreading this guest editorial.

The work on editing this special issue has been financially supported by the Science Foundation Ireland investigator programme B4-STEP (Building a Bi-directional Bridge Between Software Theory and Practice).

REFERENCES

In Advanced Topics in Database Research, Vol. 5., Siau K (ed.). Idea Group: Hershey, PA.


