Introduction to the REFSQ 2011 special issue

Daniel M. Berry · Xavier Franch

Published online: 5 February 2012
© Springer-Verlag London Limited 2012

This special issue of *Requirements Engineering Journal* contains the four best papers among the nineteen papers presented at the 17th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2011), held in Essen, Germany, during March 28–31, 2011. We, the guest editors of this special issue were the co-chairs of the conference's program committee.

Since the beginning of computing, long before the first REFSQ took place in 1994, requirements engineering (RE) has always been a major factor determining the quality of software-intensive, computer-based systems and services. From REFSQ's beginnings as a workshop, the REFSQ working conference series has steadily established itself as a leading international forum in which to discuss RE in its many relations to computer-based system quality. REFSQ seeks reports of novel ideas and techniques that enhance RE processes and artifacts as well as reflections on current research and industrial practice about and in RE. Probably, the most appreciated characteristic of a REFSQ working conference is its format in which, unlike most conferences and workshops, the discussion following a paper's presentation is as long as the presentation itself. One intended benefit of this discussion is helping the authors improve their papers for later submission to an archival journal such as this one.

We selected the four best papers, starting from recommendations sent to us after the conference by conference attendees, who had read the papers, heard the presentations, and attended the discussions. We had assigned ourselves to attend every session; whenever there were parallel sessions, each of us was at a different session. Therefore, at least one of us had heard every presentation at the conference. We considered also the original reviews and the contents of the discussion about each accepted paper both before and during the program committee meeting. Soon after the conference, we announced the four best papers and invited the authors of each to submit their paper to this special issue to be subjected a standard *Requirements Engineering Journal* reviewing process with *no guarantee of acceptance*.

To be accepted to this special issue, each conference paper had to be enhanced by at least 40%, with material that had not appeared in the conference paper. We instructed authors to, at the very least, deal with issues raised by reviewers for which there was no room in the page-restricted conference papers. We suggested that they consider the questions and issues that came up during the post-presentation question-and-answer and discussion sessions and during informal conversations during breaks and meals. Of course, the authors could bring in any new research that they might have done since sending in the final version of the conference paper some 3 months before the conference.

Each paper was reviewed by three reviewers with instructions to apply *Requirements Engineering Journal* standards. In the interest of speedier reviews of only 1 month, we invited the reviewers of each new paper's conference version to review the new paper. To replace the decliners among these invitations, we found some other experts not on the program committee who were willing to

D. M. Berry (✉)
Cheriton School of Computer Science,
University of Waterloo, Waterloo,
ON N2L 3G1, Canada
e-mail: dberry@uwaterloo.ca

X. Franch
Universitat Politècnica de Catalunya,
08034 Barcelona, Spain
e-mail: franch@essi.upc.edu

read and review the papers within 1 month. We thank all of the reviewers for their rapid but thorough reviews.

REFSQ 2011 saw an innovation that is not relevant to the best papers. However, we are so proud of this innovation that we wish to brag about it a bit in this introduction. REFSQ 2011 had two new events, both proposed by Jörg Dörr, in a new Empirical Track. One of these was the Empirical Studies at REFSQ event, organized by Jörg Dörr, in which one study of 11 submitted was selected to be conducted during a session of the first day of the conference. The second of these was the Empirical Research Fair, organized by Brian Berenbach and Nazim Madhavji. For this Fair, all 12 submitted proposals for empirical research to be done in industrial settings were selected for presentation during one session during the Industry Track day. The hope was that researchers who want to conduct the research can be matched with companies that want to provide the industrial setting and subjects for the research in exchange for first access to the results.

The four best papers are summarized:

“Task Descriptions versus Use Cases” by Soren Lauesen and Mohammad A. Kuhail describes an empirical comparison of task descriptions and use cases in their effectiveness in specifying requirements for a computer-based system, particularly those dealing with customers’ needs and problems to be handled by the new system, while also covering the solution space. The comparison was done by having each of 15 teams, about one third of them composed of researchers and two thirds of them composed of professionals, specify the requirements for one system, “to support a hotline”. Eight teams used use cases and seven used task descriptions. The discussion in the paper shows clearly how task descriptions fared better than use cases in the three comparison criteria: how well a requirements specification deals with existing problems; how well it supports comparison of potential solutions; and how well it helps avoid incorrect solutions.

“Automotive Behavioral Requirements expressed in a Specification Pattern System—A Case Study at Bosch” by Amalinda Post, Igor Menzel, Jochen Hoenicke, and Andreas Podelski describes a study to examine over 289 requirements for automotive systems at Bosch Corporation for the purpose of determining if the requirements are adequately expressible in Konrad and Cheng’s SPS (Specification Pattern System), a grammar for a restricted form of English whose sentences can be automatically converted into mathematical logic expressions. SPS is

intended to be used to allow practitioners, who are domain experts but not mathematicians, to write formal mathematical requirements specifications in what appears similar to natural English. The study found that with the addition of three new patterns, SPS was adequate for expressing all 289 requirements, thus encouraging a more general use of formal methods in the automotive industry.

“Enhancing Security Requirements Engineering by Organisational Learning” by Kurt Schneider, Eric Knauss, Siv Houmb, Shareeful Islam, and Jan Jürjens describes an approach with which an organization can learn from its and others’ experiences to identify and then deal with security-relevant requirements, even when the organization’s requirements engineers lack expertise in security. The organization’s learning is supported by the establishment of organizational security experience resources and a socio-technical network that learns how to both update and use the resources. Among the resources are heuristic assistant tools including a Bayesian classifier that learns to classify requirements as security relevant after being trained by data from past experiences. Thus, the organization, the members of the organizations, and the tools slowly teach themselves and each other about security as they gain experience using the resources and writing requirements specifications.

“Industry Needs and Research Directions in Requirements Engineering for Embedded Systems” by Ernst Sikora, Bastian Tenbergen, and Klaus Pohl reports the results of extensive interviews of industrial RE practitioners and of data mining to determine practitioners’ actual needs with respect to using natural languages versus using requirements models, dealing with highly complex systems, assuring quality of requirements specifications, moving from RE to architecture design, and relating RE with safety engineering. This study was an attempt to learn from practitioners what they *really do* and what they *really want* from RE research. Among the conclusions are that support is needed for model-based RE, decomposing systems into abstraction layers during RE, possibly automated transitions between simultaneously developed requirements specifications and architecture design, and identifying security-relevant requirements and integrating safety models into requirements models.

We hope that you will enjoy reading these papers as much as we have and will make a point of attending future incarnations of the REFSQ working conference.