Eight Conference on Software Engineering Research and Practice in Sweden (SERPS'08)

Richard Torkar, Tony Gorschek and Robert Feldt
Department of Systems and Software Engineering
Blekinge Institute of Technology
SE-372 25, Ronneby, Sweden
e-mail: {rto|tgo|rfd}@bth.se

Abstract

The eight conference on software engineering research and practice in Sweden (SERPS'08) was held in Karlskrona, Sweden, on the 4th–5th of Nov. 2008. The aim with SERPS'08 is to bring researchers and industry practitioners together to discuss software engineering issues, problems, solutions and experiences, not necessarily from a Swedish perspective. During the conference a number of research and industry papers were presented and questions in connection to the presentations were discussed. This paper is a report on the discussions that took place, pointing towards needs and challenges as well as areas of interest in both academia and industry.

1 Introduction and Background

The eight conference on software engineering research and practice in Sweden [3] has during the years grown to be the foremost venue for software engineering (SE) researchers and practitioners in Sweden. One of the main aims with SERPS is to allow primarily junior researchers to publish early and currently ongoing work and thus receive feedback at an early stage from the software engineering community. This is also the main reason for SERPS being a copyleft conference where the authors keep the copyright so that they can improve on their work and later submit it to a journal or conference. In addition SERPS is an important meeting place for senior researchers and industry practitioners, thus the interaction between industry and academia is premiered.

SERPS'08 had 76 participants from academia and industry during two days, with five sessions discussing various software engineering areas. The topics of interest included (but were not limited to): Requirements engineering, technical product management, software design, verification and validation, project management, maintenance and evolution, quality management, process assessment and improvement, methods/tools/techniques (for software development) and, finally, methods for software engineering research.

After a peer review process, where a minimum of two reviewers provided feedback on each paper, twelve papers were accepted. During the conference, each paper was assigned a discussant. The discussant had the responsibility to prepare comments on the paper prior to the presentation, and start a discussion with the following aspects in mind: *i*) Scalability and industry relevance of research. *ii*) Novelty and innovation. *iii*) Two good things about the paper/ research. *iv*) Two things that could be improved.

In the rest of this paper we cover each session with an accompanying summary of the session and paper discussion. We then conclude the paper with a brief synopsis of a questionnaire answered by the conference participants.

2 Sessions

The conference consisted of five major sessions. Every session started with presentations, followed by a thirty minutes session discussion. The discussions aimed at exchanging experiences, views on challenges and solutions, and future ideas. Both groups, industry representatives and researchers, contributed greatly to the discussions. In short, the following presentations were given (the corresponding author's e-mail address is provided so that a copy of the paper can be requested, due to the copyleft nature of the conference):

- Components
  - J. Feljan, J. Carlson and M. Žagar. Realizing a domain specific component model with Java Beans
    `juraj.feljan@mdh.se`
  - L. Lednicki, J. Carlson and M. Žagar. Uniform treatment of hardware and software components
    `luka.lednicki@mdh.se`
- Implementation
  - M. Svahnberg and K. Henningsson. Consolidating different views of quality attribute relationships
    `mikael.svahnberg@bth.se`
  - K. Lind and R. Heldal. Estimation of real-time software code size using COSMIC FSM
    `kenneth.h.lind@se.saab.com`
  - T. Steijger and T. Gutzmann. Backporting Java 5 code to legacy environments
    `tstex06@student.vxu.se`
- Modeling
  - A. Borg, M. Patel and K. Sandahl. Modeling capacity requirements in large-scale telecommunication systems
    `andbo@ida.liu.se`
  - R. Heldal and M. Staron. Using action blocks to improve quality of use cases: An initial experiment
    `heldal@chalmers.se`
  - A. Petričić, I. Crnković and M. Žagar. Models transformation between UML and a domain specific language
    `ana.petricic@mdh.se`

- Industry
  - G. Wikstrand, J. K. Gorantla and W. Zhe. Fix-cache based regression test selection: Application at Ericsson Mobile Platforms
    greger.wikstrand@ericsson.com
- Empirical
  - C. Thörn, T. Gustafsson and K. Sandkuhl. Small-scale software engineering in Småland's SMEs: A survey of software engineering practices
    thch@jth.hj.se
  - C. Lindholm and M. Höst. Development of software for safety critical medical devices: An interview-based survey of state of practice
    christin.lindholm@cs.lth.se
  - K. Petersen and C. Wohlin. Issues and advantages of using agile and incremental practices: Industrial case study vs. state of the art
    kai.petersen@bth.se

The following subsections summarize the discussions and the conclusions that were drawn from the five sessions. In Subsection 2.6 a general summary of the discussions concerning industry and academy collaboration can be found since this was a topic that was touched upon in every session.

2.1 Components

This session focused on addressing the use and applicability of components. Three areas were emphasized:

- Component-based architectures
- Service-oriented components
- Compound models combining software and hardware models

Concerning component-based architectures, participants voiced concerns on the applicability of such architectures in various domains and a call for surveys examining the the usage of these architectures in different domains was made. Furthermore, issues on how a mix of components with service-oriented focus affects various functional and non-functional requirements and, additionally, how their interaction influences heterogeneous quality attributes, were lifted as central for future research in this area.

The session ended with a discussion on the feasibility and applicability of compound models containing both software and hardware components. There was no conclusion on the industry relevance in general, however, many participants pointed out that the boundary between hardware and software components is becoming transparent and that it is unlikely that compound models of this type will be of significant value in the future.

2.2 Implementation

This session focused mainly on the concept of function points [1]; their use in, and relevance to, industry. A number of participants questioned the actual benefit of using function points in industry. The discussion then revolved around the problem with applying and using metrics in general, and function points in particular.

In the end, the consensus was that: *i*) Engineers, in general, are not using existing databases, containing function point data, correctly. *ii*) Companies are looking for a clear correlation between code size and effort when there is none to be found in many cases, i.e. companies are looking for effort but get size estimates. *iii*) Engineers are many times confused about the many different estimation approaches that exist, and when they get a hard number they do not trust it, i.e. the more advanced estimation methods are black-box. *iv*) Quality attributes are difficult to measure with function points.

2.3 Modeling

The session on modeling discussed abstraction levels in modeling and the applicability of modeling in industry. Concerning the latter, some participants claimed very few companies used modeling extensively while others said that many companies used model-based development today. There was a clear difference of views in this question and a conclusion was that a larger type survey should be conducted in order to receive indications as to how much model-based development is used in industry (since knowing the extent could actually alter the research focus).

The general agreement, among industry participants and researchers, was that generating code from models, and preferably the reverse also, is hard and very cumbersome. Simply agreeing on what to, and what not to, put under revision control is in itself a difficult question.

To conclude, source code itself can be seen as a model, just at another abstraction level. A central question was what high-level software models can provide and to what extent can they enable and make software development easier. The participants would have liked to see more studies examining the benefits (significant differences) and possible pitfalls in using models on a higher abstraction level than source code. In the end, are models, at a high abstraction level, a good way to represent knowledge?

2.4 Industry

The discussion in the industry session focused on regression test prioritization strategies (the focus of the industry paper). Industry participants voiced concerns on the viability of implementing the research conducted in test case prioritization since results, according to industry representatives, are very hard to transfer to industry. The main issues brought forward in the discussion was that researchers in this field, in general, focus on the wrong things. Researchers do experiments that are either performed on toy examples or not of considerable size (the whole point of prioritizing test case execution is to decrease the actual test execution time). The view was that researchers only perform theoretical work with a heavy foundation in mathematics, which according to several participants hinders the transfer of research to industry as no empirical evaluations (validations) are performed, not even in later stages.

It was agreed upon that research in automatic regression testing, with accompanying test selection strategies, need to be empirically evaluated on large real world cases to a much higher extent to assure usability and usefulness of research results.

2.5 Empirical

This session quickly turned to the particular challenges faced by small- and medium-sized enterprises (SMEs) when developing software, and to process improvement issues in general. The consensus was that more research involving SMEs is needed, but that it is very hard for SMEs to participate actively in research due to limited time and resources. Some general conclusions were made by the discussants:

- Advanced tools are of very little interest for SMEs. The main reason being costs connected to procurement and training. To this end, SMEs usually focus on simple tools (helping in solving one particular problem) that, additionally, in many cases are Free or Open Source Software.
- Tools for communicating inside a project are seldom used since the communication overhead in general is low in SME projects due to project size.
- The magnitude of improvement when changing processes and/or adopting new tools is largely unknown. Research could help by offering models and methods for selecting the right change to make, introducing the change and measure the effect/improvement in order to ultimately provide an answer as to what extent a SME can and should perform changes and what effect these would have on e.g. cost or increased profitability.

Regarding the difficulty for SMEs to participate in research some participants believed that PhD students working inside a SME could be one way to enable the participation of SMEs in research. However, other voices were also raised asking in what way this could contribute to a PhD student's education, how funding would be obtained, and that, in the end, the research itself needs to be of sufficient height and quality which might be endangered by, what some referred to, a 'consultancy approach'.

2.6 Challenges of Software Engineering Research Technology Transfer in Sweden

This subsection is a summary of the discussions that took place during the conference sessions covering, in particular, technology transfer [2] and the transfer of research results to industry.

The discussions during the workshop had on occasion shown disagreements about the role of research. On the one hand there were participants who clearly felt that research needs to be more 'embedded' into industry and stop focusing on 'nonexistent' or 'selfinvented' problems. On the other hand, there were others who felt that it was time to ask industry what they could do for the researchers and that researchers should lift their eyes and focus on more revolutionary research and not on incremental steps. However, a majority of the participants clearly pointed out the need for more people to be involved with technology transfer issues. These people should come from industry and academia and, in the end, help bridge the gap and, additionally, work iteratively with long term objectives in mind while knowing how decision-making is conducted in industry.

3 Summary and Conclusions

The eight conference on software engineering research and practice in Sweden provided a lively forum for sharing positions, experiences, challenges, research results and ideas in five main areas: Software components, software implementation, software modeling, industry practice and empirical research. In addition, vivid discussions took place regarding the role of research in today's society.

A questionnaire, handed out at the end of the conference (65% answer frequency), clearly indicated that the concept of using appointed discussants for each paper was very appreciated as it provided better discussions. Moreover, a question 'Which software engineering areas do you believe are in most need of improving (from an industry perspective)?' resulted in that verification & validation and project and product management were ranked top-2 among the participants. Software architecture and requirements engineering followed closely in priority. Further, the two papers by Thörn et al. and Petersen and Wohlin were according to the questionnaire the most relevant from an industry perspective.

The conference was attended by experienced senior researchers and practitioners, as well as younger more junior participants (76 participants in total). Hence, it contributed to establishing a good understanding of current practices and problems across a number of companies and research topics, as well as give inspiration to up and coming researchers. SERPS'08 provided the opportunity to present research results and to discuss current state and the future of research in software engineering. To this end the conference provided valuable feedback to companies and researchers currently working on various projects.

In addition to the specific research opportunities described in this summary, a general need for the evolution of technology transfer was identified as a central concern. It is the hope and desire of the conference participants that the conference results presented in this report will contribute to shape forthcoming research in software engineering in Sweden and in the world and enhance collaboration between industry and academia as one feeds the other [4].

References

[1] A. J. Albrecht. Measuring Application Development Productivity. In *Proceedings of the IBM Application Development Symposium*, pages 83–92. IBM Press, October 1979.

[2] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin. A Model for Technology Transfer in Practice. *IEEE Software*, 23(6):88–95, 2006.

[3] The Eight Conference on Software Engineering Research and Practice in Sweden (SERPS'08). http://www.bth.se/serps08, January 2009.

[4] Swedish Network on Software Engineering (SESWEDEN). http://www.bth.se/sesweden, January 2009.