Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 166 (2007) 1–4
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Preface

ERCIM is a European research consortium that hosts a number of Working Groups on various topics within the field of mathematics and informatics. One of these thematic groups is the *ERCIM Working Group on Software Evolution*. On 6 and 7 april 2006, this working group held its annual two-day workshop. This international workshop was hosted by the Laboratoire d'Informatique Fondamentale de Lille (LIFL) and INRIA Futurs at the Université des Sciences et Technologies de Lille (USTL) in France. It was co-organised by Tom Mens (WG chair), together with Laurence Duchien and Maja DHondt (ERCIM Postdoctoral Fellow). It was financed by the *Institut de Recherche sur les Composants logiciels et matériels pour l'Information et la Communication Avancée* (IRCICA), the *Institut National de Recherche en Informatique et en Automatique* (INRIA), and the *Université des Sciences et Technologies de Lille* (USTL).

The main goal of the workshop was to report on the theoretical, practical and empirical research on software evolution carried out by the working group members, and to discuss new opportunities for collaboration. After the workshop, the best submissions were selected, after substantial revision, for inclusion in this special issue of Electronical Notes in Theoretical Computer Science.

The rest of this preface is structured as follows. Section 1 provides some more detailed information about the workshop itself, whereas Section 2 briefly discusses the papers that have been selected for this special issue, that can be seen as the official postproceedings of the workshop.

1 About the workshop

The workshop brought together 40 researchers coming from 10 different European countries (Belgium, Finland, France, Germany, The Netherlands, Norway, Spain, Sweden, Switzerland, United Kingdom). In total, 25 position papers were submitted to the workshop, all of which were peer-reviewed by an international programme committee consisting of 17 well-known researchers. Of all submissions, 11 were invited for a long presentation, and 6 for short presentation.

In addition to its scientific purpose, the workshop also hosted the annual steering committee meeting of the ERCIM Working Group on Software Evolution. During the latter meeting, we discussed the current status of the network (including over

1571-0661/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.entcs.2006.09.010

35 members originating from research institutes all over Europe, 17 of which belong to 10 different ERCIM partner institutes). We also planned our future activities. Last, but certainly not least, we discussed about concrete opportunities and plans for proposing new initiatives within the IST domain of the EU 7th Framework Programme, in particular within the strategic objective Adaptive Software Intensive Systems. Indeed, it appears that the need for supporting software adaptation and software evolution is becoming increasingly important within this strategic objective.

The workshop presentations covered a wide variety of research topics. Among others, the following topics were addressed, with the aim to provide either better formal support or better tool support: model-driven software evolution, aspect-oriented software evolution, component-based software evolution, architectural evolution, runtime software evolution, empirical analysis, software restructuring, and software quality measurement. In addition, Arie Van Deursen of Delft University of Technology in the Netherlands gave an invited talk on The Software Evolution Paradox: An Aspect Mining Perspective. During this talk, he explored the relation between software evolution and the exciting research domain of aspect-oriented software development.

We also discussed the need for common case studies in software evolution research, in order to be able to compare research results, or to be able to assess the complementarity of different techniques on the same case study. Jean-Marie Favre proposed to use the case study of a Conference Management System for this purpose. As an alternative, Serge Demeyer already suggested in earlier work to use a simulation of a Local Area Network as a case study for teaching refactoring techniques and tools in a classroom [2]. In fact, the need for a software evolution benchmark that provides a commonly agreed set of case studies that is freely available, and that can be used to compare approaches seems to be an important and recurring theme that needs to be addressed urgently [3][1].

2 About the papers in this issue

The following papers have been included in this issue of ENTCS because of their quality and because they provide a good overview of the topics that have been presented and discussed during the workshop.

- The first article in this issue, entitled *Formal model merging applied to class diagram integration* was written by Artur Boronat, José Á Carsí, Isidro Ramos and Patricio Letelier. It proposes algebraic description techniques in the Maude language to provide more generic automated support for model management in general, and model integration (or model merging) in particular.
- The experience report *Software Evolution from the Field: An Experience Report from the Squeak Maintainers*, written by Marcus Denker and Stéphane Ducasse, discusses the lessons learned during the maintenance and evolution of the open-source Squeak software development environment for the Smalltalk programming language.

- The article *Aspect-orientation for revitalising legacy business software* by Kris De Schutter and Bram Adams shows how the techniques of *aspect-oriented programming* and *logic metaprogramming* can be used to deal with the migration, restructuring and revitalisation of legacy business software written in C or Cobol.
- The article *SmPL: A Domain-Specific Language for Specifying Collateral Evolutions in Linux Device Drivers*, written by Yoann Padioleau, Julia L. Lawall and Gilles Muller, studies the problem of *collateral evolution.* This occurs when evolution of a software library entails modifications in all affected library clients. To address this problem they propose a transformation language, called SmPL, relying on so-called *semantic patches.* They illustrate their approach on the case study of Linux device drivers.
- *A Change-based approach to software evolution* was written by Romain Robbes and Michele Lanza. It addresses the limitations of current-day version control tools, and suggests to address the problem by integrating them into software development environments. This allows them to store much more fine-grained information, taking into account the change operations performed by the software developers.
- Romain Rouvoy and Philippe Merle wrote the article *Using microcomponents and design patterns to build evolutionary transaction services.* It addresses the limitations of existing transaction services by allowing them deal with evolving concerns. Based on the Fractal component model, a framework is proposed to construct such evolutionary transaction services.

In addition, one extra paper has been added to this special issue, because it was one of the most promising papers of the French-speaking workshop on software evolution (*Atelier sur l'évolution du logiciel*), held on 21 March 2006 in Nîmes, in conjunction with the French national conference *Langages et Modèles á Objets (LMO 2006).* The paper, entitled *A methodological approach to choose components in development and evolution processes* was written by Bart George, Régis Fleurquin and Salah Sadou. It addresses the problem of composing and substituting evolving software components.

3 Summary

Considered as a whole, this special issue of *Electronic Notes in Theoretical Computer Science* covers a wide variety of topics in the research domain of software evolution, both at the level of source code and software models. Formal support and tool support is proposed for the following activities: model-driven software evolution, software quality measurement and improvement, merging and integration, aspect-oriented software evolution, component-based and architectural software evolution, runtime software evolution, software restructuring, version management, empirical analysis and domain-specific languages.

Tom Mens
Maja D'Hondt

References

[1] Serge Demeyer, Tom Mens, and Michel Wermelinger. Towards a software evolution benchmark. In *Proc. Int'l Workshop on Principles of Software Evolution*, September 2001.

[2] Serge Demeyer, Filip Van Rysselberghe, Tudor Gîrba, Jacek Ratzinger, Radu Marinescu, Tom Mens, Bart Du Bois, Dirk Janssens, Stéphane Ducasse, Michele Lanza, Matthias Rieger, Harald Gall, and Mohammad El-Ramly. The LAN simulation: A refactoring teaching example. In *Proc. Int'l Workshop on Principles of Software Evolution (IWPSE 2005)*, 2005.

[3] S. E. Sim, S. Easterbrook, and R. C. Holt. Using benchmarking to advance research: A challenge to software engineering. In *Proc, 25th Int'l Conf. Software Engineering*, pages 74–83, Portland, Oregon, May 2003. IEEE Computer Society Press.