Electronic Communications of the EASST
Volume 8 (2008)

Proceedings of the Third International ERCIM Symposium on Software Evolution (Software Evolution 2007)

PREFACE

Tom Mens, Kim Mens, Ellen Van Paesschen, and Maja D'Hondt

4 pages

Guest Editors: Tom Mens, Ellen Van Paesschen, Kim Mens, Maja D'Hondt
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

PREFACE

Tom Mens[1], Kim Mens[2], Ellen Van Paesschen[3], and Maja D'Hondt[3]

[1] Université de Mons-Hainaut, Avenue du Champ de Mars 6, 7000 Mons, Belgique
[2] Université catholique de Louvain, Place Sainte-Barbe 2, 1348 Louvain-la-Neuve, Belgique
[3] LIFL/INRIA, Cité Scientifique, 59655 Villeneuve d'Ascq, France

Abstract: This volume of Electronic Communications of the EASST contains a selection of the best submissions to the third edition of the International ERCIM Symposium on Software Evolution, co-located with ICSM in Paris in October 2007. The event was organised by the ERCIM Working Group on Software Evolution, gathering researchers from all over the world to identify and discuss recent advancements and emerging trends in the state-of-the-art in research and practice on Software Evolution.

Keywords: software evolution, ERCIM

1 Introduction

Software evolution refers to the phenomenon that software intensive systems constantly change, and *have to* change, in order to remain useful and usable. The research domain of software evolution is well-established and covers a wide diversity of different research topics [MD08]. Many conferences and workshops are devoted to software evolution, and this Volume contains the postproceedings of one of them.

On Friday 5 October 2007, the third edition of the International Symposium on Software Evolution was held, under the auspices of the ERCIM Working Group on Software Evolution. This year, the symposium was organised in Paris, in co-location with the 23rd IEEE International Conference on Software Maintenance (ICSM 2007). The organisers were Tom Mens (director of the ERCIM Working Group on Software Evolution), Kim Mens (lecturer at UCL, Belgium), Maja D'hondt (chargé de Recherches at INRIA) and Ellen Van Paesschen (ERCIM postdoctoral research fellow).

The aim of the event was to gather people from both academia and industry to identify and discuss recent advancements and emerging trends in the state-of-the-art on research and practice in software evolution. As in previous years, the event was very successful with a total of 32 submissions, three of which were tool demonstrations. Fifteen submissions and two tool demonstrations were accepted for presentation, and 11 of those were selected for inclusion in the current postproceedings, published as part of the Electronic Communications of EASST. This corresponds to an overall acceptance ratio of 34.4 %. An international programme committee of experts in the field was in charge of ensuring the quality of the accepted and published papers. 37 persons, coming from 12 different countries attended the symposium.

The presentations covered a wide variety of research topics, ranging from research advances in software refactoring, open source software evolution, reengineering of legacy code and software product lines, to evolution problems and issues in component-based, service-oriented, aspect-

oriented and model-driven software development. The two tool demonstrations focused on the support of feature modelling and maintaining design regularities in evolving code.

Last year's postproceedings were published in Elsevier's Electronic Notes in Theoretical Computer Science [MDD07].

2 About the papers in this Volume

The following papers have been selected for inclusion in this Volume of Electronic Communications of the EASST because of their quality. First, there is a series of papers in the Volume that provide empirical studies on the evolution of open source software systems:

- The article *On the Resilience of Classes to Change* by Rajesh Vasa et al. studies the evolution of several public domain open source object-oriented (Java) software systems. More specifically, the rate and amount of change in individual classes is investigated. It turns out that most of the effort during evolution is spent on introducing new classes; only a small percentage of existing classes is modified, and these modifications tend to be small.

- In the article *Package evolvability and its relationship with refactoring*, Asma Mubarak et al. study the evolution of seven open source systems in Java at package level, and try to investigate change trends in overall changes applied to these systems, complemented with empirical evidence in refactoring data for these systems.

- Karl Beecher et al. report on the *Evolutionary success of open source software: an investigation into exogenous drivers*. In particular, the chosen *forge* (i.e., the online repository in which the open source software products are made available to the community) seems to have an important impact on the relative success and change rate of a software project. These findings are corroborated by comparing about 50 projects from Debian with about 50 projects from SourceForge.

Two articles in this Volume are related to model-driven engineering, and deal with aspects like model refactoring and model-code synchronisation:

- The article *Refactoring of UML models using AGG* by Alessandro Folli and Tom Mens reports on an experiment to implement UML model refactorings using the AGG graph transformation tool, and provides recommendations on how AGG and graph transformation in general may be improved to facilitate such support.

- László Angyal et al. present *Novel Techniques for Model-Code Synchronization* that can be used to better support model-based iterative software development through efficient model-code change propagation. A bi-directional synchronization technique is proposed based on three-way abstract syntax tree differencing and merging, and a case study is provided to illustrate the technique.

Two more articles in this Volume address aspects of migration and reverse engineering:

- The article *Lightweight Visualisations of COBOL Code for Supporting Migration to SOA* by Joris Van Geet and Serge Demeyer studies how legacy systems, programmed in COBOL, can be migrated to a service-oriented architecture in a stepwise fashion. To this extent, existing reverse engineering and visualisation techniques are used throughout the process. Some preliminary results of applying the approach on an industrial case study are discussed.

- The *Supporting Reengineering Scenarios with Fetch: an Experience Report* by Bart Du Bois et al. introduces *Fetch*, a tool chain for reengineering. This tool chain has been applied in an industrial setting, and the results of this experience report are presented.

The remaining articles in this Volume are quite diverse, and treat topics like build systems, acceptance testing, and software product line evolution.

- Filippo Ricca et al. report on *The Use of Executable FIT Tables to support Maintenance and Evolution Tasks*. They carried out an experiment with master students to investigate the usefulness of executable acceptance tests for software evolution activities. The acceptance tests were developed with the Framework for Integrated Test (FIT). The initial results seem to indicate that evolution activities can be performed better (in the sense that the correctness of the evolved code is improved) in roughly the same amount of time as without using the technique.

- Bram Adams et al. studied *The evolution of the Linux build system*. Just as source code is subject to continuous evolution, so is the build system. The challenge is to keep it fast, even though its complexity continues to grow over time. This study is quite interesting in that the laws of software evolution also seem to hold for the build system. In addition, it turns out that there is some inevitable amount of co-evolution between the source code and its build system.

- Klaus Schmid and Holger Eichelberger propose *A requirements-based taxonomy of software product line evolution*. From the viewpoint of requirements engineering, a taxonomy of evolution operations for software product lines is proposed. The presented taxonomy can be used as a reference for product line evolution support that goes beyond existing levels of support.

Acknowledgements: The symposium was sponsored by the following organisations:

- The *European Research Consortium for Informatics and Mathematics* (ERCIM) www.ercim.org

- The *INRIA Futurs* research center in France www.inria.fr/futurs

- The *Université des Sciences et Technologies de Lille* (USTL) in France www.univ-lille1.fr

- The *Fonds de la Recherche Scientifique* - FNRS in Belgium
  www.frs-fnrs.be
- The *Interuniversity Attraction Poles Programme* - Belgian State Belgian Science Policy.
  prog.vub.ac.be/moves

Bibliography

[MD08] T. Mens, S. Demeyer. *Software Evolution*. Springer, 2008.

[MDD07] T. Mens, M. D'Hondt, L. Duchien (eds.). *Proceedings of the ERCIM Worshop on Software Evolution 2006*. Volume 166. Elsevier Science, January 2007.