

Guest Editors Introduction: Special Issue on Innovative Automated Software Engineering Tools – Part #2

John Grundy and John Hosking¹

Introduction

This is the second part of our Special Issue on Innovative Automated Software Engineering tools. Over 30 papers were submitted to the special issue, demonstrating the key role that tools have in the ASE research and practice communities. The five accepted papers appearing in this second part of the special issue underwent rigorous refereeing, major and minor revisions and then were selected for their scientific contributions. Coincidentally the papers cover an interesting range of ASE tool application areas, including model-based testing, embedded software engineering, symbolic execution and model checking, process-centred environments, and software evolution. Several of these innovative tools adopt service-based and compositional approaches to tool integration. Most have been validated not only on academic problems but also with significant industrial applications and software teams.

We would like to thank very much the efforts of all authors of papers both appearing in the special issue, but also those whose work we were unable to accept. We also would like to thank the large number of referees from the automated software engineering community who we had to draw upon very heavily in this process. Finally, we thank Anand David from Springer for excellent assistance with editorial system management and production work, and Bob Hall for having the faith in us to Guest Edit this special issue.

Papers in Part 2 of the Special Issue on Innovative ASE Tools

Filho, Hasling, Budnik and McKenna present a tool for model-based testing, tedeso, used within Siemens corporation. They describe the motivation and design of an innovative tool to integrate into the development process to support workflow-driven, service-oriented model-based testing. This service-oriented architecture of the tedeso framework is a novel approach to extensible tool frameworks, as well as delivering the model-based testing tool functionality. The authors recount key issues in adopting such a service-oriented architecture and associated workflow for their toolset. The

¹ J. Grundy

Centre for Computing & Engineering Software Systems, Swinburne University of Technology, Melbourne, Australia
email: jgrundy@swin.edu.au

J. Hosking

College of Engineering & Computer Science, Australian National University, Canberra, Australia
email: john.hosking@anu.edu.au

tool has been adopted by a number of Siemens business units and the authors recount experiences of the users of their platform.

Voelter, Ratiu, Kolb and Schaetz describe mbeddr, a novel automated tool for engineering embedded software systems. mbeddr is a language workbench built upon the C programming language that provides embedded software system engineers a range of domain-specific capabilities tailored to their particular needs. The tool provides rich C code annotation, requirements tracing, model checking, debugging and integrated transformation support. mbeddr is targeted at small to medium enterprises with particular embedded software development needs. They report on a detailed evaluation of their toolset deployed in practice and the advantages of the language workbench approach.

Pasareanu, Visser, Bushnell, Geldenhuys, Mehltz and Rungta describe Symbolic PathFinder, an innovative automated tool combining symbolic execution with model checking. SPF generates test cases for Java programs by analyzing their byte code and executing the programs with symbolic inputs. Constraints generated on expressions in the programs are solved with a range of constraint solvers, and model checkers are used to explore symbolic program executions. The tool architecture and implementation is described along with reports of experiences using the tool in a range of contexts, including academia, industry and NASA.

Maciel, Gomes, Magalhães, Silva and Queiroz present MoDErNE, a model-driven, process-centred software engineering environment. Their tool incorporates support both for process-centric software engineering task management and co-ordination along with model-driven engineering facilities. They argue for a more process-centric support platform for MDD including formalized transformation specification and chaining. MoDErNE has been evaluated on several significantly sized problems, several of which are reported as case studies.

Ghezzi and Gall present SOFAS, a service-oriented tool approach to software evolution analysis, including bug data, version control information and issue tracking. They introduce a framework supporting aggregation and storage of evolution information and integrating a variety of diverse analysis tools using this unified data source. A set of analysis tools uses a REST communication infrastructure and the platform includes semi-automated analysis tool service composition. Two different applications of the platform are described to illustrate its tool service composition approach and its flexibility.

We do hope that you enjoy this second installment on Innovative ASE tools!