In this issue

Rachel Harrison¹

Published online: 5 May 2017

© Springer Science+Business Media New York 2017

In this issue, we have nine regular research papers. The first four papers are concerned with various aspects of software testing and diagnostics whereas the next four are all related through the themes of design and architecture. The final paper in this issue is concerned with the customization of enterprise resource planning systems.

In “Test-Driven Development in scientific software: a survey”, Aziz Nanthamornphong and Jeffrey Carver present the results of a survey of scientific developers who use test-driven development. They report that test-driven development does help scientific developers improve software quality but that it requires significant investment initially in terms of effort.

Continuing with the testing theme, the paper “H-Switch Cover: a new test criterion to generate test case from finite state machines”, by Érica Ferreira de Souza, Valdivino Alexandre de Santiago Júnior, and Nandamudi Lankalapalli Vijaykumar, presents a new test criterion for model-based test case generation. The criterion is called H-Switch Cover and is based on the Switch Cover test criterion with additional heuristics. The H-Switch Cover test criterion was found to offer improved cost and efficiency when compared to two other test criteria.

In “Testing of model-driven development applications”, Beatriz Marín, Carlos Gallardo, Diego Quiroga, Giovanni Giachetti, and Estefanía Serral present techniques that automatically generate executable test cases for software developed with the model-driven development paradigm in order to save time and effort during testing. The techniques were applied to an industrial project and were found to be very efficient when compared to traditional testing techniques.

Diagnostics are an important part of the testing activity. The paper “Analysis of service diagnosis improvement through increased monitoring granularity” by Cuiting Chen, Hans-Gerhard Gross, and Andy Zaidman analyzes the trade-offs between the diagnostic improvements that can be gained through increased monitoring and the overheads caused by the introduction of greater monitoring in the context of spectrum-based fault localization for service-oriented systems. The authors find that monitoring at the service implementation level has a relatively low overhead, but this is dependent on the number of monitors deployed.

✉ Rachel Harrison
rachel.harrison@brookes.ac.uk

¹ Department of Computing and Communication Technologies, Oxford Brookes University, Oxford OX33 1HX, UK

Turning our attention to design, the paper “Multi-view refactoring of class and activity diagrams using a multi-objective evolutionary algorithm” by Usman Mansoor, Marouane Kessentini, Manuel Wimmer, and Kalyanmoy Deb proposes a framework that enables software designers to apply refactoring during design and so analyze trade-offs between the quality of class and activity diagrams. The authors performed a statistical evaluation using four open-source systems to determine the efficiency of the approach and found it to compare well.

In “Quantitative risk analysis of safety-critical embedded systems”, Yinling Liu, Guohua Shen, Zhiqiu Huang, and Zhibin Yang propose a method for quantitative risk analysis using an architecture design language, AADL. Their method can automatically analyze an AADL model to obtain a risk-based failure mode effect analysis table in order to analyze both the severity and cost of risks. The method provides both formal specification of errors and automatic safety analysis.

Continuing with the design theme, the paper “Multi-objective code-smells detection using good and bad design examples” by Usman Mansoor, Marouane Kessentini, Bruce R. Maxim, and Kalyanmoy Deb describes how to use multi-objective genetic programming to find a combination of metrics that maximizes the detection of code smells whilst minimizing the detection of well-designed code. The authors evaluated their technique on seven large open-source systems and found that it performed well according to both the precision and recall metrics.

In “Availability analysis of software architecture decomposition alternatives for local recovery”, Hasan Sözer, Mariëlle Stoelinga, Hichem Boudali, and Mehmet Akşit present a number of techniques to predict the availability of systems for local recovery. The techniques are designed to work at the architectural level. For validation, the authors applied their techniques to the MPlayer open-source software and found that the estimated availabilities were very similar to the actual measured availabilities.

The final paper in this issue is concerned with enterprise resource planning systems. The paper “Impact of customization over software quality in ERP projects: an empirical study” by Sudhaman Parthasarathy and Srinarayan Sharma analyzes the impact of the customization of enterprise resource planning systems on software quality. The authors report that an enterprise resource planning team should customize each business process in order to safeguard the quality of the enterprise resource planning system.

I hope you enjoy reading this issue. As always, please send any comments on this issue to me at rachel.harrison@brookes.ac.uk.