Guest editors' introduction to the special section on exploring the boundaries of software test automation

Christof J. Budnik · W. K. Chan ·
Gregory M. Kapfhammer · Hong Zhu

Published online: 19 August 2011
© Springer Science+Business Media, LLC 2011

This special section includes five papers that are substantially extended and revised versions of the best papers presented at AST 2010. Since the first AST workshop held at ICSE 2006 in Shanghai, China, the research on the automation of software testing has developed significantly. The papers in this special section cover a wide range of topics related to software testing. They clearly reflect this trend of diversity in research on software test automation.

Test case generation is still one of the most active research topics in the research on software test automation. Two papers in this special section are devoted to test case generation. The paper by Mike Papadakis and Nicos Malevris entitled Automatically Performing Weak Mutation with the Aid of Symbolic Execution, Concolic and Search Based Testing combines symbolic execution, concolic testing and mutation testing techniques from a search-based approach to test case generation. In the paper entitled Enhancing Structural Software Coverage by Incrementally Computing Branch Executability, Mauro Baluda, Pietro Braione, Giovanni Denaro and Mauro Pezze propose a technique that not only generates test cases that execute uncovered branches, but also identifies infeasible branches so that they can be excluded from the calculation of the branch coverage.

C. J. Budnik (✉)
Siemens Corporation, Corporate Research, System Development Technologies,
755 College Road East, Princeton, NJ 08540, USA
e-mail: christof.budnik@siemens.com

W. K. Chan
City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
e-mail: wkchan@cs.cityu.edu.hk

G. M. Kapfhammer
Department of Computer Science, Allegheny College, Box Q, 520 North Main Street, Meadville,
PA 16335, USA
e-mail: gkapfham@allegheny.edu

H. Zhu
School of Technology, Oxford Brookes University, Wheatley Campus, Oxford, UK
e-mail: hzhu@brookes.ac.uk

Test automation is no longer limited to test case generation. Test oracles have become an active research topic of test automation. In this special section, we present a paper by Rene Just and Franz Schweiggert on Automating Unit and Integration Testing with Partial Oracles, which describes how an oracle can detect failures in a program's output.

In the paper entitled TORC: Test Plan Optimization by Requirements Clustering, Baris Güldali, Holger Funke, Stefan Sauer and Gregor Engels propose an automated method for deriving test plans from requirements expressed in natural language. Their method aims to reduce the cost of acceptance testing.

In the paper entitled Automating Performance Testing of Interactive Java Applications, Andrea Adamoli, Dmitrijs Zapanuks, Milan Jovic and Matthias Hauswirth report a case study on the feasibility of using five Java GUI capture-and-replay tools for GUI-based performance test automation. They discovered a new problem of using such GUI test tools for performance testing, that is, the temporal synchronization problem, which is of increasing importance for GUI applications that contain timer-driven activities.

The guest editors would like to take this opportunity to express their gratitude to the PC members and reviewers of AST 2010 for their excellent work in the review of the papers for the workshop and this journal special section and to the authors who contributed to AST 2010 and this special section. We would also like to thank the workshop participants who contributed in the discussions and charette sessions at AST 2010. Lastly, we would like to thank the Software Quality Journal for publishing this special section.