Guest Editors’ Introduction: International Symposium on Software Testing and Analysis

Sebastian Elbaum and David S. Rosenblum

1 INTRODUCTION

THE International Symposium on Software Testing and Analysis (ISSTA) is the premier forum for the presentation of research results on leading edge issues related to software testing and analysis. Every year, ISSTA brings together academics, industrial researchers, and practitioners to present and discuss the most promising approaches for using testing and analysis to assess and improve software and the processes by which it is engineered.

ISSTA 2007 was held at the Brunei Gallery of the University of London’s School of Oriental and African Studies (SOAS) in the beautiful city of London, United Kingdom, 9-12 July 2007. The conference was preceded by two workshops: the Workshop on Parallel and Distributed Systems: Testing and Debugging (PADTAD) and the Workshop on Advances in Model-Based Testing (A-MOST). For ISSTA 2007, 101 research papers were submitted. Submissions were reviewed by at least three reviewers and discussed in a Program Committee meeting. The result was a high-quality technical program with 22 accepted research papers that covered a variety of topics, including fault isolation, testing, model checking, and various hybrid analysis approaches.

This special section of the IEEE Transactions on Software Engineering contains papers based on five of the best papers of ISSTA 2007. These papers were accepted from a set of six papers we invited for this special section based on selections made by the ISSTA Program Committee from among the papers presented at the conference. The papers, incorporating revisions and enhancements by the authors, went through the standard TSE review process involving at least three anonymous reviewers per paper, overseen by us as guest editors. The result is a selection of five excellent papers.

2 THE PAPERS

The first paper, “Discovering Neglected Conditions in Software by Mining Dependence Graphs” by Ray-Yaung Chang, Andy Podgurski, and Jiong Yang, presents an approach that uses static program analysis and data mining techniques to extract rules from code whose violation may indicate missing paths. The paper discusses the representations and mechanisms supporting the approach and presents a series of studies assessing the quality of the mined rules and the nature of the detected violations.

The second paper, “Delta Execution for Efficient State-Space Exploration of Object-Oriented Programs” by Marcelo d’Amorim, Steven Lauterburg, and Darko Marinov, introduces a technique to speed up state-space exploration, a key activity in model-checking and automated test generation. The technique identifies paths’ overlaps through a tailored state representation to enable their simultaneous exploration. The technique has been implemented on top of two model checkers and the results indicate that it can provide important efficiency improvements.

The third paper, “Interface Grammars for Modular Software Model Checking” by Graham Hughes and Tevfik Bultan, proposes an approach to support modular model-checking through the use of grammars. The paper presents a framework to specify these grammars so that they can be used to generate stubs to model the components that form the environment of the module being checked. A case study performed on clients of the Enterprise Java Beans Persistence API shows the enabling potential of the proposed approach.

The fourth paper, “Constructing Interaction Test Suites for Highly-Configurable Systems in the Presence of Constraints: A Greedy Approach” by Myra B. Cohen, Matthew B. Dwyer, and Jiangfan Shi, addresses the problem of testing systems in the presence of large but constrained configuration spaces. The paper introduces algorithms that integrate traditional combinatorial integration testing strategies with SAT solver technology to provide support for the modeling and solving of constraints. An empirical study across several artifacts shows the cost-effectiveness gained through the approach.

The fifth paper, “Static Specification Mining Using Automata-Based Abstractions” by Sharon Shoham, Eran Yahav, Stephen J. Fink, and Marco Pistoia, presents a static analysis approach for mining temporal API specifications...
from client code. The paper presents the approach to deduce typestate-style properties for objects as their state evolves throughout sequences of library calls and provides details on the mechanisms used to more precisely model aliasing and event sequences. The effectiveness of the approach is assessed on a series of Java libraries.

ACKNOWLEDGMENTS

This special section is the result of a great deal of effort from many volunteers in our community. The members of the ISSTA 2007 Program Committee helped form the ISSTA program through their efforts in reviewing and later discussing the submitted papers at the program committee meeting and they helped select the papers that appear in this section. The anonymous reviewers generously shared their time and efforts. Jo Atlee helped us to manage the whole process as the Associate Editor liaison of this special section. To all of you, thank you.

Sebastian Elbaum
David S. Rosenblum
Guest Editors

Sebastian Elbaum received the PhD degree in computer science from the University of Idaho and the degree in systems engineering from the Universidad Catolica de Cordoba, Argentina. He is an associate professor at the University of Nebraska-Lincoln. His research interests include program analysis and testing, deployed software analysis, end-user software engineering, and empirical software engineering. He received the US National Science Foundation Faculty Early CAREER Award in 2004 for his research on the utilization of field data to test highly configurable and rapidly evolving pervasive systems. He was the program chair for ISSTA 2007 and program cochair for ESEM 2008. He has served as a program committee member for several international conferences, including ICSE and FSE.

David S. Rosenblum received the PhD degree from Stanford University. He is a professor of software systems at University College London. His current research focuses on scalability analysis of software systems, testing and analysis of ubiquitous computing systems, dynamic service discovery in service-oriented architectures, and probabilistic modeling and analysis of software systems. He served as general chair of ISSTA 2007 and is the current chair of the ISSTA Steering Committee and he has served in numerous additional organizational roles for technical meetings, including program cochair of ICSE 2004, program chair of SIGSOFT FSE 2000, and chair of the ICSE Steering Committee. He is the vice chair of ACM SIGSOFT and holds a Wolfson Research Merit Award from the Royal Society in the United Kingdom. He is a fellow of the IEEE, the British Computer Society, and the Institution of Engineering and Technology.