Introduction: The Best Papers of ISSTA 2008

Barbara G. Ryder and Andreas Zeller

Abstract—We present the best papers of the International Symposium on Software Testing and Analysis (ISSTA) 2008.

In this special section of the IEEE Transactions on Software Engineering, we have compiled six papers selected from the International Symposium on Software Testing and Analysis (ISSTA) 2008.

What is ISSTA about? It is the leading research conference in software testing and analysis, bringing together academics, industrial researchers, and practitioners to exchange new ideas, problems, and experience. Its 2008 incarnation was held in Seattle, Washington; out of 101 submissions, the program committee accepted 26 papers for the conference.

For this special section of the IEEE Transactions on Software Engineering, we have selected six papers that represent the best of what the conference had to offer, covering a wide variety of topics, from static analysis via fault localization to hybrid testing and analysis approaches.

The paper “An Experience in Testing the Security of Real-world Electronic Voting Systems” by Davide Balzarotti, Greg Banks, Marco Cova, Viktoria Felmetsgger, Richard Kemmerer, William Robertson, Fredrik Valeur, and Giovanni Vigna discusses the testing of voting systems—the machines that help in the voting process of democratic societies. In their testing, they identified major flaws and implemented a number of attacks which allowed them to take complete control of the examined voting systems; the paper describes the methodology, the findings, and the lessons learned.

Errors are also common in Web applications, and seriously impact usability and reliability. In “Finding Bugs in Dynamic Web Applications,” Shay Artzi, Adam Kiezun, Julian Dolby, Frank Tip, Danny Dig, Amit Paradkar, and Michael D. Ernst present a dynamic test generation technique for the domain of dynamic Web applications. The approach combines concrete and symbolic execution with model checking to automatically generate tests and minimal bug reports. Their Apollo prototype revealed 302 faults in six PHP Web applications.

In “Proofs from Tests,” Nels E. Beckman, Aditya V. Nori, Sriram K. Rajamani, Robert J. Simmons, Sai Deep Tetali, and Aditya V. Thakur explore how to leverage test executions to progressively guide the construction of program proofs. Their approach simultaneously performs program testing and program abstraction, scales much better than previous approaches, and has been applied to verify properties of 69 Windows Vista drivers.

In “Racer: Effective Race Detection Using Aspect,” Eric Bodden and Klaus Havelund address the problem of detecting concurrent programming errors such as data races. Their approach uses a language extension to the aspect-oriented programming language AspectJ to monitor program events where locks are granted or handed back, and where shared values are accessed. Applied to the NASA K9 Rover Executive, the approach detected 11 previously unknown data races, without false positives.

The paper “The Probabilistic Program Dependence Graph and Its Application to Fault Diagnosis” by George K. Baah, Andy Podgurski, and Mary Jean Harrold introduces a new model for a program’s internal behavior, called the probabilistic program dependence graph (PPDG). PPDGs extending traditional dependences with estimates of statistical dependences between node states, based on the established framework of probabilistic graphical models. As a first application of PPDGs, the authors show that PPDGs can facilitate fault localization and fault comprehension.

Last but not least, “Learning a Metric for Code Readability” by Raymond Buse and Westley Weimer explores the concept of code readability and investigates its relation to software quality. With data collected from 120 human annotators, they derive associations between a simple set of local code features and human notions of readability; from those features, they construct an automated readability measure.

With their originality, their depth, and their fearless combination and extension of existing techniques, these papers represent the state of the art in software testing and analysis, and manifest the vibrant dynamics as well as the tremendous pace of the research in the field. We thank the authors for providing these contributions, and our anonymous reviewers for their detailed and constructive comments. Enjoy the read!
Barbara G. Ryder received the PhD degree in computer science from Rutgers University in 1982, and served on the faculty there from 1982-2008. She is head of the Department of Computer Science at Virginia Tech, where she holds the J. Byron Maupin Professorship in Engineering. She also worked in the 1970s at AT&T Bell Laboratories in Murray Hill, New Jersey. Dr. Ryder’s research interests lie in static and dynamic program analyses for object-oriented systems, focusing on usage in practical software tools for ensuring the quality and security of industrial-strength applications. She became a fellow of the ACM in 1998, received the ACM President’s Award in 2008, was selected as a CRA-W Distinguished Professor in 2004, and received the ACM SIGPLAN Distinguished Service Award in 2001. She has been an active leader in the ACM (e.g., Secretary-Treasurer 2009-2010, ACM Council 2000-2008, general chair, FCRC 2003, chair ACM SIGPLAN (1995-1997)) and has served as a member of the Board of Directors of the Computer Research Association (1998-2001). She has served as an editorial board member of the ACM Transactions on Programming Languages and Systems, IEEE Transactions on Software Engineering, IEEE Software, Software, Practice and Experience, and Science of Computing. She was general chair of ISSTA 2008.

Andreas Zeller received the PhD degree in computer science from TU Braunschweig, Germany, in 1999, and has served on the faculty of Saarland University, Saarbrücken, Germany, since 2001, where he is now a professor of software engineering. His research interests lie in the analysis of programs and processes, especially the analysis of why programs fail to work as they should. In 2009, he received the ACM SIGSOFT Impact Paper Award for his work on delta debugging as the most influential software engineering paper of 1999. His book Why Programs Fail received the 2005 Software Productivity Award as one of the three most productivity-boosting books of the year. Dr. Zeller has served on the editorial boards of the ACM Transactions on Software Engineering and Methodology and Springer Journal on Empirical Software Engineering. He was program chair of ISSTA 2008.