4-2005

Guest Editor's Introduction: IEEE 2004 International Symposium on Software Testing and Analysis

Gregg Rothermel
University of Nebraska-Lincoln, grothermel2@unl.edu

Guest Editor's Introduction: IEEE 2004 International Symposium on Software Testing and Analysis

Gregg Rothermel, *Member*, *IEEE*

1 Introduction

The International Symposium on Software Testing and Analysis (ISSTA) is the leading research conference in software testing and analysis. 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 2004 was held in at the Parker House Hotel in Boston, Massachusetts, July 11-14, 2004. On this occasion the conference was co-located with the 16th Computer-Aided Verification Conference (CAV 2004), with the two conferences sharing a day of sessions. The conference was also preceded by two workshops: the Workshop on Testing, Analysis, and Verification of Web Services and the Workshop on Empirical Research in Software Testing.

The number of submitted papers to ISSTA, and the ratio of that number to accepted papers, are a good indication of ISSTA's health as a conference, and technical reputation. For ISSTA 2004, 93 research papers (a near record) and eight tools 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 26 research papers and two tools papers.

This issue of *IEEE Transactions on Software Engineering* presents a special section containing papers based on five of the best papers of ISSTA 2004. These papers were part of a set of seven papers selected by the ISSTA Program Committee from among the papers presented at the conference. The selected set, following revisions and enhancements by the authors, went through the standard *TSE* review process involving at least three anonymous reviewers per paper, overseen by myself as guest editor (or in one case on a paper on which I had a conflict, by John Knight). The result is a selection of five excellent papers.

2 The Papers

The first paper, "Exploiting Purity for Atomicity," by C. Flanagan, S.N. Freund, and S. Qadeer, presents an approach for verifying the atomicity (a useful noninterference property of multithreaded programs, that supports the use of various analysis and validation activities) of code blocks that are not reducible—and, hence, for which atomicity could not previously be established. The approach exploits "purity," a property of a code block that neither reads nor writes state when terminating normally, and allows the code to be treated as a "no-op" by the atomicity analysis. The approach is illustrated on several interesting examples where atomicity could not previously be verified.

The second paper, "Robustness Testing of Java Server Applications," by C. Fu, A. Milanova, B.G. Ryder, and D.G. Wonnacott, presents an approach to help testers exercise error recovery code in Java systems—code that is often left untested due to difficulties that arise in executing it. The approach relies on exception-catch link analysis and additional algorithms to identify and prune a set of exception-catch links to be tested; this provides a coverage criterion and also guides a fault-injection process that simulates faults, allowing testing of exception handling code. The paper reports promising results via a study in which the approach is applied to several nontrivial Java systems.

The third paper, "Profiling Deployed Software: Assessing Strategies and Testing Opportunities," by S. Elbaum and M. Diep, empirically investigates the potential for using data gathered from fielded software systems to assist in analysis and testing activities. The empirical work involves a controlled experiment in which several profiling techniques are employed on a large deployed system, and the profiling data is used to direct several different client uses of that data. Both the profiling techniques and the client uses are evaluated to assess cost-benefit trade-offs. The results suggest that the use of profile data from the field can be beneficial, while also revealing several tradeoffs related to that use.

The fourth paper, "Software Assurance by Bounded Exhaustive Testing," by D. Coppit, J. Yang, S. Khurshid, W. Le, and K. Sullivan, presents the results of a case study in which bounded exhaustive testing (which tests a system using all valid program inputs up to specified size bounds) is applied to a large, real application, using Alloy and TestEra to generate all nonisomorphic inputs up to a given "size." The results show that the approach scales effectively

• *The author is with the Department of Computer Science and Engineering, 360 Avery Hall, University of Nebraska—Lincoln, Lincoln, NE 68588. E-mail: grother@cse.unl.edu.*

For information on obtaining reprints of this article, please send e-mail to: tse@computer.org.

to the system investigated, producing meaningful test inputs that reveal nontrivial faults.

The fifth paper, "Predicting the Location and Number of Faults in Large Software Systems," by T.J. Ostrand, E.J. Weyuker, and R.M. Bell, builds on data gathered from 26 fielded releases of two large-scale, industrial software systems to construct a statistical model for predicting fault counts and fault densities for files the systems are composed of. The model is found to be quite accurate relative to the actual faults that had been found in these systems. The fact that this relatively simple approach can be used to select files for retesting is interesting, especially given the large mass of system data on which the results are based.

ACKNOWLEDGMENTS

This special issue is the result of a great deal of work by many people—most of them volunteers. The members of the ISSTA 2004 program committee helped form the ISSTA program through their efforts in reviewing and at the program committee meeting, and they helped select the papers that appear in this issue. The anonymous reviewers generously shared their time and efforts. John Knight, the Editor-in-Chief for *IEEE Transactions on Software Engineering*, oversaw the reviewing process for a paper on which I had a conflict. George Avrunin, the general chair for ISSTA 2004, organized the conference, provided indispensible input on program matters, and helped with the selection process for the papers appearing in this edition. To all of you, my warmest thanks.

Gregg Rothermel received the PhD degree in computer science from Clemson University, the MS degree in computer science from the State University of New York at Albany, and the BA degree in Philosophy from Reed College. He is currently a professor and Jensen Chair of Software Engineering in the Department of Computer Science and Engineering at University of Nebraska—Lincoln. He has also spent several years in industry as a software engineer and as Vice President of Quality Assurance and Quality Control for Palette Systems, Inc., a manufacturer of CAD/CAM software. Dr. Rothermel's research interests include program analysis and its uses in software testing and maintenance, and end-user software engineering, with a particular emphasis on controlled experimentation. Dr. Rothermel received the US National Science Foundation Faculty Early CAREER Award in 1996 for his research on regression testing. He is also a cofounder of and coprincipal investigator with the EUSES Consortium. Dr. Rothermel is an associate editor-in-chief for *IEEE Transactions on Software Engineering* and a program cochair for the 29th International Conference on Software Engineering, and he has served in the past as the program chair for the International Symposium on Software Testing and Analysis, and Chair of the Steering Committee for the International Conference on Software Maintenance. He is a member of the IEEE, ACM, ACM SIGSoft, and ACM SIGPlan.