

Guest Editors' Introduction: 2000 International Symposium on Software Testing and Analysis

Mary Jean Harrold, *Member, IEEE Computer Society*, and Antonia Bertolino

1 INTRODUCTION

THIS special section contains five papers from the 2000 International Symposium on Software Testing and Analysis (ISSTA 2000). As we enter the third millennium, we face many new and challenging problems in providing efficient techniques and tools that help software engineers develop high-quality software; research in software testing and analysis addresses many of these problems. ISSTA 2000, which was held in Portland, Oregon, in August 2000, brought together academics, industrial researchers, and practitioners to exchange new ideas, problems, and experiences related to how testing and analysis can reduce the incidence of software failures and improve the overall quality of the software.

The program for ISSTA 2000 consisted of research papers, invited speakers, and a joint session with collocated Third Workshop on Formal Methods in Software Practice (Third FMSP). The Program Committee selected 17 regular papers and four short papers from the 73 submitted papers. These papers address many important testing and analysis issues related to object-oriented software, component-based systems, real-time systems, and database applications. The program also had a number of papers that report empirical studies on testing and analysis techniques. The ISSTA 2000 program also consisted of two invited presentations on state-of-the-art and future directions: one on finite-state verification by L.A. Clark and the other on testing software components by C.H. Wittenberg. In the opening address, Jon Pincus presented his experiences in developing and deploying software tools and, in the joint address with Third FMSP, D. Dill discussed model checking for Java programs. The ISSTA Proceedings were published as *ACM SIGSOFT Software Engineering Notes*, vol. 25, no. 5, September 2000.

Following the symposium, the committee selected seven papers which represented the best of ISSTA 2000. Revised versions of these papers underwent a second rigorous review process that involved reviewers external to the program committee. Of these, we accepted five papers for this special section.

- M.J. Harrold is with the College of Computing, Georgia Institute of Technology, 801 Atlantic Dr., Atlanta, GA 30332-0280.
E-mail: harrold@cc.gatech.edu.
- A. Bertolino is with the Istituto di Elaborazione della Informazione, Area della Ricerca CNR di Pisa-San Cataldo, 56100 Pisa, Italy.
E-mail: bertolino@iei.pi.cnr.it.

For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 115157.

2 THE ARTICLES

The selected papers span a range of topics in software analysis and testing. The first paper is "Improving the Precision of INCA by Eliminating Solutions with Spurious Cycles" by S.F. Seigel and G.S. Avrunin. INCA is a finite-state verification tool that can check properties of concurrent systems with very large state spaces. In exchange for greater tractability, INCA accepts the possibility of spurious results—reports about property violations that cannot actually occur. Basically, INCA models a concurrent system as a collection of communicating Finite State Automata (FSAs), produces a system of equations and inequalities specifying the execution flows over the FSAs that would violate the property to be checked, and looks for integer solutions to this system by using standard integer linear programming methods. If a solution is found, this could be a violation of the property, but, in some cases, it could also correspond to no real execution. In the paper, the authors explain the two possible causes for these inconclusive results and propose a technique for preventing one of those causes: when spurious solutions correspond to cycles not connected to the initial state and, hence, not executable. Some preliminary experiments to assess the cost of applying the method are also described.

In the paper "Verisim: Formal Analysis of Network Simulations," K. Bhargavan, C.A. Gunter, M. Kim, I. Lee, D. Obradovic, O. Sokolsky, and M. Viswanthan present a tool suite called Verisim that facilitates the formal analysis of performance and correctness properties of network protocols, using simulations. Verisim is obtained by the integration of a popular network simulator tool, NS, with the trace-checker component of the Java MaC (Monitoring and Checking) framework. The authors illustrate the application of Verisim to the AODV case study, that is, the Ad Hoc On-Demand Distance Vector routing protocol used in packet radio networks.

The paper "Requirements-Based Monitors for Real-Time Systems," by D.K. Peters and D.L. Parnas provides a formal definition of a monitor—a system that observes the behavior of another system (the target system) and checks whether it conforms to the requirements. The monitor can be used during testing (as an oracle) or in operation and the observation can be done concurrently with the target system execution or a posteriori on an execution record. The behavior of the target system is represented by two sets of environmental quantities, the monitored quantities and the controlled quantities, respectively, both of which can be

modeled as functions of time. Thus, monitors are suitable in particular to analyze the behavior of real-time systems. The authors discuss in detail how to design and implement monitors for real-time systems and identify some necessary conditions under which the monitors are proven to be useful for the analysis task at hand.

The fourth paper, "Test Case Prioritization: A Family of Empirical Studies" by S. Elbaum, A. Malishevsky, and G. Rothermel, reports and discusses the empirical results obtained from a set of controlled experiments and case studies, aimed at investigating the usage of prioritization techniques in regression testing. This paper builds on previous empirical work which showed how the application of some defined prioritization techniques can significantly improve the fault detection rate. The present study refines and complements the earlier findings by addressing several additional questions concerning the generality or vice versa, the specificity, of prioritization with regard to the software version, the level of granularity at which prioritization is applied (e.g., source code statements or function), and the incorporation of predictors of fault proneness into the prioritization technique. The empirical observations provide several interesting implications for both practitioners and researchers and, most importantly, demonstrate that the selection of the most appropriate prioritization technique or even the decision to adopt one (as opposed to random ordering) depends on many factors and is not obvious at all.

Finally, in "Simplifying and Isolating Failure-Inducing Input," A. Zeller and R. Hildebrandt present two versions of an algorithm for simplifying and isolating the circumstances that caused an observed failure. More precisely, the minimizing Delta Debugging algorithm (or ddmin) takes a complex bug report and automatically extracts from it a minimal test case such that every part of it is relevant in reproducing the reported failure. This is clearly useful in facilitating debugging and in grouping bug reports that differ only in insignificant details. A generalization of ddmin is also proposed, called the general Delta Debugging algorithm, or dd, which can find the failure-inducing input more efficiently as a difference between a successful test case and a failing one. Examples illustrate the application of the two algorithms to real-world case studies.

ISSTA is a biennial event. We invite you to attend the next symposium, which will be held in Rome in July 2002 (collocated with the Workshop on Software Performance (WOSP)). Information about ISSTA 2002 can be found at <http://www.iei.pi.cnr.it/ISSTA2002>.

ACKNOWLEDGMENTS

The program of the ISSTA'00 is the result of hard work by many dedicated people. The program committee members reviewed a large number of papers during a relatively short reviewing period, provided thoughtful and thorough reviews, participated in the discussion of the papers at the program committee meeting, selected the papers for the symposium, and helped in the creation of the final program. In addition to her many tasks as general chair for both ISSTA 2000 and the third FMSP, Debra Richardson provided much support and assistance in the reviewing

process, the program committee meeting, and the creation of the final program. As program chair for the third FMSP, Mats Heimdahl helped in coordinating the many joint activities of ISSTA 2000 and the third FMSP. Finally, we thank the *IEEE Transactions on Software Engineering* Editor-in-Chief Anneliese Amschler Andrews and the IEEE Computer Society staff for their patience, support, and cooperation during the review process.



Mary Jean Harrold received the BS and MA degrees in mathematics from Marshall University and the MS and PhD degrees in computer science from the University of Pittsburgh. She is currently an associate professor in the College of Computing at the Georgia Institute of Technology, Atlanta. Her research interests include the development of efficient techniques and tools that will automate, or partially automate, development, testing, and maintenance tasks.

Her research to date has involved program-analysis-based software engineering, with an emphasis on regression testing, analysis and testing of imperative and object-oriented software, and development of software tools. Her recent research has focused on the investigation of the scalability issues of these techniques through algorithm development and empirical evaluation. She is a recipient of the US National Science Foundation's National Young Investigator Award. Dr. Harrold serves on the editorial board of the *IEEE Transactions on Software Engineering*, *ACM Transactions on Programming Languages and Systems*, and *Journal of Empirical Software Engineering*. She served as the program chair for the ACM International Symposium on Software Testing and Analysis (July 2000) and the program cochair of the 23rd International Conference on Software Engineering (May 2001). She is a member of the Computing Research Association's Committee on the Status of Women in Computing and she directs the committee's Distributed Mentor Project. She is a member of the IEEE Computer Society and the ACM.



Antonia Bertolino graduated cum laude in electronic engineering (Laurea degree) from the University of Pisa, Italy, in 1985. Since 1986, she has been a researcher with the Institute for Information Processing of the Italian National Research Council (CNR) in Pisa. Her research interests are in software engineering, especially software testing and software dependability. Currently, she is investigating approaches for rigorous integration test strategies,

for architecture and UML-based test approaches, and for improving the cost/effectiveness of testing techniques, as well as methods for the evaluation of software reliability. She is responsible for the recently established Pisatel Laboratory, for joint research and training between Ericsson Lab Italy and CNR. Since 1994, she has been an associate editor of the *Journal of Systems and Software*, and, since 2000, of the *IEEE Transactions on Software Engineering*. She is the general chair of the forthcoming ACM Symposium on Software Testing and Analysis ISSTA 2002 (Rome, July 2002) and was the general chair of the Second International Conference on Achieving Quality in Software (AQUIS'93), held in Venice (Italy) in October 1993. She has been a member of the international program committees of several conferences and symposia, including ISSTA, Joint ESEC-FSE, ICSE, SEKE, Safecomp, and Quality Week. She has been the key area specialist for the software testing knowledge area of the Stone Man phase of the ACM/IEEE project *Guide to the SWEBOK* (Software Engineering Body of Knowledge). She has (co)authored more than 40 papers in international journals and conferences.

► For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.