Guest Editors’ Introduction: 2003 International Conference on Software Engineering

Laura K. Dillon, Senior Member, IEEE, and Walter F. Tichy, Member, IEEE Computer Society

1 INTRODUCTION

Managing the complexity of modern software systems is without question a grand challenge. It is therefore fitting that, inspired by the view of Mount Hood in Portland, Oregon, the theme for the Silver Anniversary of the International Conference on Software Engineering (ICSE 2003) was “Scaling New Heights.” This special section features three papers based on work first presented at this meeting of the premier conference for software engineering.

Each morning of the ICSE 2003 conference started with a topical keynote address by one of the outstanding invited speakers, Bertrand Meyer, JoAnne McGrath Cohoon, and Eugene Spafford. These talks kicked off an exciting program which included 42 technical papers, 16 experience reports, 11 education papers, and a variety of minitutorials, panels, and formal research demonstrations. Some notable highlights of this program were: three minitutorials on new and promising software engineering technologies in the Frontiers of Software Practice track, one each by Clemens Szyperski, Douglas Schmidt, and Richard Kemmerer; a minitutorial on how to write a good research paper in software engineering by Mary Shaw; and participation from the automotive community in a session on automotive software engineering. Throughout the conference, there were exhibits, posters, and informal research demonstrations, as well as an onsite newsletter describing memorable moments, humor, and fascinating facts.

Prior to and immediately following the main ICSE 2003 program, there were 17 tutorials on a variety of topics and 15 workshops that offered a forum for interaction. There were also three special events: the Pioneers Symposium, the New Software Engineering Faculty Symposium, and the Doctoral Symposium. Finally, the week included five workshops that offered a forum for interaction. There were 17 tutorials on a variety of topics and 15 workshops that offered a forum for interaction.

Following the conference, the program committee selected four papers as representing the best of ICSE 2003. Authors were invited to submit revised and expanded versions of the conference papers for this special issue. Three of the papers were revised and submitted. These submissions underwent a second rigorous review process that involved reviewers external to the program committee.

2 THE ARTICLES

The first paper in this special issue concerns development of software. Specifically, the paper, “Scaling Step-Wise Refinement” by D. Batory, J.N. Sarvela, and A. Rauschmayer extends the authors’ earlier work on synthesizing code via feature refinements to synthesize disparate artifacts associated with a software product, including code, Makefiles, design documents, and so on. In the authors’ earlier work, features are defined by collections of classes and class fragments, which are composed with classes that implement a software product using subclassing. This paper generalizes the notion of feature composition to apply to collections of artifacts and artifact fragments in different notations. The authors’ AHEAD model defines an algebra with which to define applications as nested sets of equations. The paper shows that software can have an elegant and concise mathematical structure.

The next two papers concern validation of software. In particular, the second paper, “Fragment Class Analysis for Testing of Polymorphism in Java Software” by A. Rountev, A. Milanova, and B.G. Ryder addresses the problem of computing coverage criteria in testing polymorphism. For example, criteria may require coverage of all possible classes of a receiver object at a call site, or of all possible methods that could be invoked at a call site. The paper proposes to use static class analyses to compute collections of bindings to cover in satisfying these criteria. However, existing static class analyses operate on only whole programs, whereas testing techniques must be able to handle incomplete programs. The paper therefore describes a general method for adapting whole-program class analyses to work on program fragments and provides a proof that the resulting fragment class analyses are correct. Further, it evaluates the precision of several such analyses and identifies analyses that are good candidates for use in coverage tools.

Finally, the paper “Modular Verification of Software Components in C” by S. Chaki, E. Clarke, A. Groce, S. Jha, and H. Veith describes a compositional methodology for
verifying C programs. The methodology marries different lines of work to great benefit: assume-guarantee reasoning to decompose the verification along procedure boundaries; predicate abstraction to build labeled transition systems from procedures; theorem proving to drive predicate abstraction; a formulation of weak simulation as a Boolean satisfiability problem to verify that a procedure conforms to its specification; and counterexample guided abstraction refinement to suggest improved abstractions, when needed. The paper describes the authors’ MAGIC tool, which supports this methodology and reports on results using MAGIC.

3 Future ICSE Conferences

ICSE is an annual event sponsored by ACM SIGSOFT and IEEE Computer Society. ICSE 2004, which is taking place in the ancient Scottish city of Edinburgh, will have passed by the time of this printing. We hope you were able to participate in it. In any case, we invite you to attend the next conference, ICSE 2005, to be held in St. Louis, Missouri, 15-21 May 2005. Information about ICSE 2005 can be found at http://www.cs.wustl.edu/icse05/Home/index.shtml.

Acknowledgments

Many people devoted considerable time, energy, and hard work to help organize ICSE 2003. We would like to thank everyone who served on the organizing committee. Special thanks are in order for Margaret Burnett who did an excellent job on publicity, Gregg Rothermel who kept the Web pages up-to-date, and Jo Atlee and Stefan Leue who oversaw the ICSE workshops and colocated events. We thank the program committee members for preparing high quality reviews and devoting two long days to the program committee meeting. The record number of 324 submissions meant that each program committee member wrote 23 reviews in the space of five weeks. We also thank Holger Hopp (University of Karlsruhe) and Richard van de Stadt (Borbala Conference Services), who kept the submission and reviewing server operating smoothly (and even recovered from a fire), and Mary Kate Rada and Kristine Weiand, from IEEE Conference Services, and Anne Jacobs, from IEEE publishing, who provided valuable support. Finally, the authors thank the IEEE Transactions on Software Engineering Editor-in-Chief John Knight and the IEEE Computer Society staff for their support and cooperation during the review process.

Laura K. Dillon received the PhD degree in computer science from the University of Massachusetts in 1984. Subsequently, she was on the faculties of Computer and Information Science and Electrical and Computer Engineering at the University of Massachusetts, Amherst (1984–1985), and Computer Science at the University of California, Santa Barbara (1985-1998). In 1997, she joined the Department of Computer Science and Engineering at Michigan State University. She is completing her first year as chair of that department. She has been active in the IEEE and ACM since 1980, serving over the years in various editorial capacities and on numerous program committees, including as a program cochair of the 2003 International Conference on Software Engineering. Currently, she is a member of the executive committee of the ACM Special Interest Group in Software Engineering. She is also a member of Computer Professionals for Social Responsibility (CPSR), American Association of University Women (AAUW), and Women in Engineering Programs and Advocate Network (WEPAN). Her research interests center on formal methods for specification and analysis of concurrent software systems, programming languages, and software engineering. She is a senior member of the IEEE.

Walter F. Tichy received the MS and PhD degrees in computer science from Carnegie Mellon University, Pittsburgh, Pennsylvania in 1976 and 1980, respectively. He has been professor of computer science at the University Karlsruhe, Germany, since 1986. Previously, he was senior scientist at Carnegie Group, Inc., in Pittsburgh, Pennsylvania, and served six years on the faculty of Computer Science at Purdue University in West Lafayette, Indiana. His primary research interests are software engineering and parallelism. He is currently directing research on a variety of topics, including empirical software engineering, autonomic computing, software configuration management, cluster computing, optimizing compilers for parallel computers, and optoelectronic interconnects. He has consulted widely for industry. He is the director of the Forschungszentrum Informatik, a technology transfer institute. He is a cofounder of ParTec AG, a company specializing in software for computer clusters. He was a program cochair for the 25th International Conference on Software Engineering (2003). Dr. Tichy is a member of ACM, GI, and the IEEE Computer Society.