Guest Editors’ Introduction: 2001 International Conference on Software Maintenance

Gerardo Canfora, Member, IEEE, and Anneliese A. Andrews

1 INTRODUCTION

This special section of the IEEE Transactions on Software Engineering contains three papers from the 2001 International Conference on Software Maintenance (ICSM 2001). The pressure on evolving software systems to adapt to rapidly changing business scenarios poses new challenges to researchers and practitioners in software maintenance. With “Systems and Software Evolution in the Era of the Internet” as its theme, ICSM 2001, which was held in Florence, Italy, in November 2001, provided a truly international forum for researchers, developers, and users interested in maintenance issues for both existing and emerging software systems.

The technical program of ICSM 2001 included 68 papers and five short papers, selected out of 161 submissions. Each paper was reviewed by at least three members of the program committee. The review process consisted of two steps: The authors of submitted papers were provided with the comments of the reviewers before the PC meeting and were given the opportunity to submit a rebuttal. This enriched the information available to the members of the program committee for making their decisions. The program also included a Doctoral Symposium, which allowed PhD students to present and discuss their research objectives, methods, and achievements. Two industrial sessions provided accounts of the practical application of software maintenance techniques, tools, and processes. Finally, distinguished speakers David L. Parnas, Dieter Rombach, and Dr. Ivar Jacobson provided keynotes on the state-of-the-art and future directions of software maintenance.

Following the conference, a selection of eight papers representing the best papers of ICSM 2001 was nominated for possible inclusion in this special section. The authors were invited to submit extended versions of their conference papers. Extended papers underwent a second rigorous review process that resulted in the selection of three excellent papers for this special issue.

2 THE ARTICLES

The papers in this special section address three key problems of software maintenance, namely, regression testing, program comprehension, and legacy system reengineering.

The first paper, “Test-Suite Reduction and Prioritization for Modified Condition/Decision Coverage,” by J.A. Jones and M.J. Harrold, deals with regression testing of high-assurance software. Testing evolving software typically entails that a test-suite be developed for the initial version of the system and reused to test subsequent versions. The size of the test suite tends to grow over time as new test cases are added either to establish confidence that changes are correct or to ensure that unchanged portions of the system have not been affected. Of course, the increasing size of the test suite also means that the cost of running it on the modified system increases. The paper proposes two new algorithms for test-suite reduction and one new algorithm for test-suite prioritization that apply to modified condition/decision coverage. Empirical studies assess the effectiveness of the test-suite reduction algorithms and evaluate the performance of the test-suite prioritization algorithm.

In the second paper, “Locating Features in Source Code,” T. Eisenbarth, R. Koschke, and D. Simon discuss a novel approach to support scenario-driven, goal-oriented program comprehension. The proposed approach combines static and dynamic analysis techniques to localize the implementation of a certain feature in the code. First, concept analysis is performed for analyzing the execution traces for different usage scenarios and building a concept lattice. The lattice helps the user to understand which code components contribute more specifically to implement a given set of features. Then, a static analysis uses the identified, feature-specific code components to locate additional contributing components by navigating the dependency graph. The approach is evaluated in two case studies that demonstrate that combining static and dynamic analysis reduces the search space drastically.

Legacy systems can pose significant challenges for software maintenance. In fact, while they may be crucial systems for the business they support, they are often hard to modify and difficult to evolve to meet new and constantly changing business requirements. The paper “Iterative Reengineering of Legacy Systems,” by A. Bianchi, D. Caivano, V. Marengo, and G. Visaggio, describes an approach to reengineer legacy systems in an iterative manner. The iterative reengineering process guarantees...
that the underlying system continues to operate during reengineering and preserves the maintainers’ and users’ familiarity with the system, by making only small, gradual changes during each iteration. This requires the coexistence of legacy components, components undergoing reengineering, reengineered components, and new components added to satisfy new requirements. The proposed method is experimentally applied to an old industrial legacy system supporting distributors of chemical products.

ACKNOWLEDGMENTS

The program of the 2001 International Conference on Software Maintenance is the result of hard work by many dedicated people. We thank the authors of all submitted papers and the members of the program committee who reviewed them in a relatively short period, providing thoughtful and thorough comments. We are also indebted to Paolo Nesi: In addition to his many tasks as the General Chair of ICSM 2001, he participated actively in the reviewing process, the program committee meeting, and the creation of the final program. Finally, we appreciate the hard work of the additional reviewers who helped to make this special section possible.

Gerardo Canfora received the Laurea degree in electronic engineering from the University of Naples, Federico II, Italy, in 1989. He is currently a full professor of computer science at the Faculty of Engineering and the Director of the Research Centre on Software Technology (RCOST) of the University of Sannio in Benevento, Italy. From 1990 to 1991, he was with the Italian National Research Council (CNR). During 1992, he was at the Department of Informatica e Sistemistica of the University of Naples, Federico II, Italy. From 1992 to 1993, he was a visiting researcher at the Centre for Software Maintenance of the University of Durham, UK. In 1993, he joined the Faculty of Engineering of the University of Sannio in Benevento, Italy. He has served on the program committees of a number of international conferences. He was a program cochair of the 1997 International Workshop on Program Comprehension and of the 2001 International Conference on Software Maintenance and he is the general chair of the 2003 European Conference on Software Maintenance and Reengineering. His research interests include software maintenance, program comprehension, reverse engineering, reuse, reengineering, migration, workflow management, and document management. He serves on the Editorial Board of the IEEE Transactions on Software Engineering. He is a member of the IEEE and the IEEE Computer Society.

Anneliese A. Andrews received the Dipl. Inf. degree from the Technical University in Karlsruhe in 1976. She obtained the Master’s and PhD degrees in computer science from Duke University in 1978 and 1979, respectively. She holds the Huie-Rogers Endowed Chair in Software Engineering at Washington State University. She also serves as the associate director of the School of Electrical Engineering and Computer Science. Dr. Andrews is the author of more than 150 publications in software testing, maintenance, design, and performance modeling. She has served on numerous program committees, conference steering and organization committees, and editorial boards.