

Lecture Notes in Computer Science

2931

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Alexandre Petrenko Andreas Ulrich (Eds.)

Formal Approaches to Software Testing

Third International Workshop on
Formal Approaches to Testing of Software, FATES 2003
Montreal, Quebec, Canada, October 6th, 2003
Revised Papers



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Alexandre Petrenko
Centre de recherche informatique de Montréal (CRIM)
550 Sherbrooke West, Suite 100, Montreal, H3A 1B9, Canada
E-mail: Petrenko@crim.ca

Andreas Ulrich
Siemens AG, Corporate Technology CT SE 1
Otto-Hahn-Ring 6, 81730 Munich, Germany
E-mail: andreas.ulrich@siemens.com

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at <<http://dnb.ddb.de>>.

CR Subject Classification (1998): D.2.5, D.2.4, D.2, D.3.1, F.3.1

ISSN 0302-9743

ISBN 3-540-20894-1 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media
springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10984468 06/3142 5 4 3 2 1 0

Preface

Formal methods provide system designers with the possibility to analyze system models and reason about them with mathematical precision and rigor. The use of formal methods is not restricted to the early development phases of a system, though. The different testing phases can also benefit from them to ease the production and application of effective and efficient tests. Many still regard formal methods and testing as an odd combination. Formal methods traditionally aim at verifying and proving correctness (a typical academic activity), while testing shows only the presence of errors (this is what practitioners do). Nonetheless, there is an increasing interest in the use of formal methods in software testing. It is expected that formal approaches are about to make a major impact on emerging testing technologies and practices. Testing proves to be a good starting point for introducing formal methods in the software development process.

This volume contains the papers presented at the 3rd Workshop on Formal Approaches to Testing of Software, FATES 2003, that was in affiliation with the IEEE/ACM Conference on Automated Software Engineering (ASE 2003). This year, FATES received 43 submissions. Each submission was reviewed by at least three independent reviewers from the program committee with the help of additional reviewers. Based on their evaluations, 18 papers submitted by authors from 13 different countries were selected for presentation at the workshop. The papers present different approaches to using formal methods in software testing. One of the main themes is the generation of an efficient and effective set of test cases from a formal description. Different models and formalisms are used, such as finite state machines, input/output transition systems, timed automata, UML, and Abstract State Machines. An increasing number of test methodologies (re)uses techniques from model checking. The prospects for using formal methods to improve software quality and reduce the cost of software testing are encouraging. But more efforts are needed, both in developing new theories and making existing methods applicable to the current practice of software development projects. Without doubt, coming FATES workshops will continue to contribute to the growing and evolving research activities in this field.

We wish to express our gratitude to the authors for their valuable contributions. We thank the program committee and the additional reviewers for their support in the paper selection process. Last but not least, we thank May Haydar who helped in organizing the proceedings and all persons from the Centre de Recherche Informatique de Montréal and the organizing committee of ASE 2003 who were involved in arranging local matters.

Program Committee

Marie-Claude Gaudel	Université Paris-Sud, France
Jens Grabowski	Georg-August-Universität Göttingen, Germany
Dick Hamlet	Portland State University, USA
Robert M. Hierons	Brunel University, UK
Thierry Jéron	IRISA, France
David Lee	Bell Labs Research, China
Brian Nielsen	Aalborg University, Denmark
Jeff Offutt	George Mason University, USA
Alexandre Petrenko	CRIM, Canada
Jan Tretmans	Universiteit Nijmegen, The Netherlands
Andreas Ulrich	Siemens AG, Germany
Antti Valmari	Tampereen Teknillinen Yliopisto, Finland
Carsten Weise	Ericsson, Germany
Martin Woodward	Liverpool University, UK

Additional Reviewers

Serge Boroday	Keqin Li
Rene de Vries	Andrej Pietschker
Lars Frantzen	Vlad Rusu
Hesham Hallal	Valery Tschaen
May Haydar	Machiel van der Bijl
Ruibing Hao	Dong Wang
Jia Le Huo	Gavin Yang
Claude Jard	Xia Yin
Ahmed Khoumsi	

Table of Contents

Program Testing and Analysis

Black-Box Testing of Grey-Box Behavior	1
<i>Benjamin Tyler and Neelam Soundarajan</i>	
On Checking Whether a Predicate Definitely Holds	15
<i>Alper Sen and Vijay K. Garg</i>	
Using a Software Testing Technique to Improve Theorem Proving	30
<i>Reiner Hähnle and Angela Wallenburg</i>	
Auto-generating Test Sequences Using Model Checkers: A Case Study	42
<i>Mats P.E. Heimdahl, Sanjai Rayadurgam, Willem Visser, George Devaraj, and Jimin Gao</i>	
Mutually Enhancing Test Generation and Specification Inference	60
<i>Tao Xie and David Notkin</i>	
JMLAutoTest: A Novel Automated Testing Framework Based on JML and JUnit	70
<i>Guoqing Xu and Zongyuang Yang</i>	

Test Theory and Test Derivation Algorithms

Compositional Testing with IOCO	86
<i>Machiel van der Bijl, Arend Rensink, and Jan Tretmans</i>	
Defining Observation Objectives for Reactive and Distributed Systems	101
<i>Timo Aaltonen</i>	
Time-Optimal Real-Time Test Case Generation Using UPPAAL	114
<i>Anders Hessel, Kim G. Larsen, Brian Nielsen, Paul Pettersson, and Arne Skou</i>	
Test Cases Generation for Nondeterministic Real-Time Systems	131
<i>Ahmed Khoumsi, Thierry Jéron, and Hervé Marchand</i>	
Property Oriented Test Case Generation	147
<i>Jean-Claude Fernandez, Laurent Mounier, and Cyril Pachon</i>	
Computing Unique Input/Output Sequences Using Genetic Algorithms ...	164
<i>Qiang Guo, Robert M. Hierons, Mark Harman, and Karnig Derderian</i>	
Automatic Generation of Test Purposes for Testing Distributed Systems ..	178
<i>Olaf Henniger, Miao Lu, and Hasan Ural</i>	

Test Methods and Test Tools

Interaction Testing in an Embedded System
Using Hardware Fault Injection and Program Mutation 192
Ahyoung Sung and Byoungju Choi

Automatic Conformance Testing of Internet Applications 205
Harm M.A. van Beek and Sjouke Mauw

A Use Case Driven Testing Process: Towards a Formal Approach
Based on UML Collaboration Diagrams 223
Mourad Badri, Linda Badri, and Marius Naha

VISWAS and on Diagnosability with IEEE Std P1522
and UML2.0 Testing Profile 236
Sita Ramakrishnan

Towards a Tool Environment for Model-Based Testing with AsmL 252
*Mike Barnett, Wolfgang Grieskamp, Lev Nachmanson,
Wolfram Schulte, Nikolai Tillmann, and Margus Veanes*

Author Index 267