Lecture Notes in Computer Science 4454

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Yuri Gurevich Bertrand Meyer (Eds.)

Tests and Proofs

First International Conference, TAP 2007
Zurich, Switzerland, February 12-13, 2007
Revised Papers

Springer

Volume Editors

Yuri Gurevich
Microsoft Research
Redmond, WA 98052, USA
E-mail: gurevich@microsoft.com

Bertrand Meyer
ETH Zurich
8092 Zurich, Switzerland
E-mail: Bertrand.Meyer@inf.ethz.ch

Library of Congress Control Number: 2007931908

CR Subject Classification (1998): D.2.4-5, F.3, D.4, C.4, K.4.4, C.2

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-73769-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73769-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12095476 06/3180 5 4 3 2 1 0

Preface

To prove the correctness of a program is to demonstrate, through impeccable mathematical techniques, that it has no bugs. To test a program is to run it with the expectation of discovering bugs.

These two paths to software reliability seem to diverge from the very start: if you have proved your program correct, it is fruitless to comb it for bugs; and if you are testing it, that surely must be a sign that you have given up on any hope to prove its correctness.

Accordingly, proofs and tests have, since the onset of software engineering research, been pursued by distinct communities using different kinds of techniques and tools. Dijkstra's famous pronouncement that tests can only show the presence of errors — in retrospect, perhaps one of the best advertisements one can imagine for testing, as if "only" finding bugs were not already a momentous achievement! — didn't help make testing popular with provers, or proofs attractive to testers.

And yet the development of both approaches leads to the discovery of common issues and to the realization that each may need the other. The emergence of model checking was one of the first signs that apparent contradiction may yield to complementarity; in the past few years an increasing number of research efforts have encountered the need for combining proofs and tests, dropping earlier dogmatic views of incompatibility and taking instead the best of what each of these software engineering domains has to offer.

TAP — Tests And Proofs — results from an effort to present and discuss some of the most interesting of today's research projects at the convergence of proofs and tests. The first event of its kind, TAP 2007 was held at ETH Zurich on February, 12–13 2007. The conference demonstrated that this is indeed a vibrant topic with exciting developments and the potential for much further growth and cross-fertilization between the ideas pursued by many groups.

We hope that you will agree that TAP 2007 advanced the understanding of two equally promising approaches to software quality, and that you will find in the results, collected in this volume, a source of insight inspiration, and new challenges.

The success of TAP was the result of contributions by many people. We are particularly grateful to the authors who submitted excellent papers; to the keynote speakers, Yuri Gurevich, Jonathan Ostroff and Yannis Smaragdakis; to the Program Committee members and outside referees who made it possible to conduct an effective process leading to a selection of high-quality papers.

The conference was sponsored by IFIP; we are particularly grateful to the support of IFIP Working Group WG2.3 on Programming Methodology (through its Chairperson, Pamela Zave, and all the other members who supported the idea of IFIP sponsorship) as well as TC2 (the Technical Committee on Programming, especially its Chair Robert Meersman and its then secretary Judith Bishop). ETH Zurich provided excellent facilities and impeccable organization.

The financial support of Microsoft Research was particularly useful and is gratefully acknowledged.

The organization, including the preparation of these proceedings, was made possible by the work of the Organizing Committee: Ilinca Ciupa, Manuel Oriol, Andreas Leitner, Claudia Günthart, and Lisa Liu without whom the conference could not have taken place.

Yuri Gurevich
Bertrand Meyer

Organization

Committees

Conference Chair

Bertrand Meyer, ETH Zurich, Switzerland and Eiffel Software, California, USA

Program Chair

Yuri Gurevich, Microsoft Research, USA

Program Committee

Chandrasekhar Boyapati, University of Michigan, USA
Ed Clarke, Carnegie Mellon University, USA
Michael Ernst, MIT CSAIL, USA
Kokichi Futatsugi, JAIST, Japan
Tom Henzinger, EPFL, Switzerland
Daniel Kroening, ETH Zurich, Switzerland
Gary T. Leavens, Iowa State University, USA
Bertrand Meyer, ETH Zurich, Switzerland
Peter Müller, ETH Zurich, Switzerland
Huaikou Miao, Shanghai University, China
Jeff Offutt, George Mason University, USA
Jonathan Ostroff, York University, Canada
Benjamin Pierce, University of Pennsylvania, USA
Wolfram Schulte, Microsoft Research, USA
Yannis Smaragdakis, University of Oregon, USA
Tao Xie, North Carolina State University, USA
T.H. Tse, University of Hong Kong, China

External Referees

Gerard Basler
Nicolas Blanc
Arindam Chakrabarti
Yuri Chebiriak
Adam Darvas
Weiqiang Kong
Masaki Nakamura
Martin Nordio
Kazuhiro Ogata
Joseph Ruskiewicz
Faraz Torshizi
Jianwen Xiang

Organizing Committee

Lisa (Ling) Liu, ETH Zurich, Switzerland
Ilinca Ciupa, ETH Zurich, Switzerland
Andreas Leitner, ETH Zurich, Switzerland
Claudia Günthart, ETH Zurich, Switzerland
Manuel Oriol, ETH Zurich, Switzerland

Sponsors

ETH Zurich
IFIP
Microsoft Research

Table of Contents