Lecture Notes in Computer Science 4171

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

Editorial Board

Bertrand Meyer Jim Woodcock (Eds.)

Verified Software: Theories, Tools, Experiments

First IFIP TC 2/WG 2.3 Conference, VSTTE 2005
Zurich, Switzerland, October 10-13, 2005
Revised Selected Papers and Discussions

Springer

Volume Editors

Bertrand Meyer
ETH Zurich, Department of Computer Science
Clausiusstr. 59, 8092 Zurich, Switzerland
E-mail: bertrand.meyer@inf.ethz.ch

Jim Woodcock
University of York, Department of Computer Science
Heslington, York YO10 5DD, UK
E-mail: jim@cs.york.ac.uk

Library of Congress Control Number: 2008930409

CR Subject Classification (1998): D.1.0, D.2, D.2.1, D.2.6, D.2.13, D.3.1, D.4.5, F.3.1, F.4.1, F.4.3

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-69147-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-69147-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12282821 06/3180 5 4 3 2 1 0

Preface

A Step Towards Verified Software

Worries about the reliability of software are as old as software itself; techniques for allaying these worries predate even James King's 1969 thesis on "A program verifier." What gives the whole topic a new urgency is the conjunction of three phenomena: the blitz-like spread of software-rich systems to control ever more facets of our world and our lives; our growing impatience with deficiencies; and the development—proceeding more slowly, alas, than the other two trends—of techniques to ensure and verify software quality.

In 2002 Tony Hoare, one of the most distinguished contributors to these advances over the past four decades, came to the conclusion that piecemeal efforts are no longer sufficient and proposed a "Grand Challenge" intended to achieve, over 15 years, the production of a *verifying compiler*: a tool that while processing programs would also guarantee their adherence to specified properties of correctness, robustness, safety, security and other desirable properties. As Hoare sees it, this endeavor is not a mere research project, as might normally be carried out by one team or a small consortium of teams, but a momentous endeavor, comparable in its scope to the successful mission to send a man to the moon or to the sequencing of the human genome. It requires a cogent agenda with both scientific and engineering components; support from the scientific community and from governments; and the collaboration of many groups, spread across many countries and including many specialties, for the advancement of a common goal.

VSTTE — the first conference on Verified Software: Theories, Tools, Experiments —(http://vstte.ethz.ch) was the launching event for this undertaking. Held at ETH Zurich in October of 2005, the conference gathered about 100 participants, the elite of software verification from a dozen countries. The conference was by invitation only. Although of course not every top expert could be invited, and not all who were invited could come, the attendee list makes up a sample of the best work currently being pursued in software verification. The present volume is the record of this memorable event.

We have not just gathered the position papers contributed by most of the participants. Attendees of scientific conferences know well that the benefit does not just come from formal paper presentations; question-and-answer sessions after talks are another prime source of insights, especially with participants as distinguished as those of VSTTE. In an effort to increase the attraction of this book for readers who were not in Zurich, we recorded the discussions and include the transcripts here (with the exclusion of some sessions with no associated paper). To preserve the spontaneity of these exchanges, we have kept the editing to a minimum, essentially removing a few superfluous comments. (Because the process took some time, it also serves as our excuse for the delay between the conference and the present publication.) We hope that you will enjoy the added value of this material, and perhaps even that you will at times feel like you are actually attending a live VSTTE session, including the occasional inaudible comment.

From "Exclusive ors" to Conjunction

The field of software verification has long been marked by competition between approaches, often resulting in binary oppositions—view A vs. view B. The oppositions still exist, and help define a rough multi-dimensional classification of the techniques represented at VSTTE and elsewhere; in attempting this classification, however, we should note that the variants along each dimension are increasingly recognized as complementary rather than exclusive. Software verification is sufficiently hard that a Grand Challenge effort must glean help from every good idea.

The first distinction pits *management*-oriented approaches against those based on *technology*. VSTTE was technology-oriented and this volume emphasizes the latter kind, but you will find occasional references to project management, the political context, model-driven development, process models and the like. Bad management can be as effective as bad technology in destroying software quality. In the rest of this discussion we limit ourselves to technical aspects.

A thick line has long divided proponents of *constructive* approaches to software construction from people who understand "verification" in the same way Reagan did in his reaction to Gorbachev's arms-reduction promises: "Trust, but *verify*." This division can also be called "a priori vs. a posteriori": build the software right, in the Dijkstra tradition of *A Discipline of Programming*, continued today by such techniques as proof-supported refinement, well represented in this volume; or scrutinize the result, with the help of powerful tools, for possible quality violations. This is an example of where opposition tends nowadays to yield to convergence: we need the best techniques to produce software that will have no bugs; and the best techniques to find any bugs that, all these efforts notwithstanding, still slip through the cracks.

Also prominent in this volume is the contrast between authors who concern themselves mostly with *specification*, often to the point of dismissing the advances of modern programming languages, and those who work at the implementation level, using these very programming languages as their basic tool. Even that conflict is not as irreducible as it sounds, since effective approaches using specification languages must go all the way to implementation, and advances in programming languages bring them to a level of abstraction not so far removed from the realm of specification.

The next opposition is between *static* approaches, which work on the basis of the software text only, and *dynamic* ones which require its execution. Examples of the first category are static analysis and proofs; the primary example of the second is testing. Although VSTTE did not prominently feature work on testing, this approach has made significant progress in recent years, and researchers are increasingly recognizing what software practitioners have known for years: that testing is today, and will remain for a long time, a key part of almost any effort to produce quality software.

Among static approaches, you will note the contrast between *full proofs* of correctness, which for any significant software system must command the support of powerful computer-based proof systems (the authors of several of the most famous of these tools were present at VSTTE and contributed to this volume), and *static analysis*, a term that generally describes ascertaining more partial properties of the software. Static analysis uses tools too, sometimes even the same theorem provers.

This is one more distinction that tends to get blurred, as the successes of partial static analysis continue to expand the scope and ambition of the properties they assess.

The distinction between static and dynamic approaches might seem to defy such blurring; but that too would be a wrong guess. The testing community's long-time differences with the advocates of proofs are fading out; a successor conference series of VSTTE, Tests And Proofs (TAP, http://tap.ethz.ch), launched at ETH Zurich in 2007, explores this new convergence. Also significant is the growing popularity of *model checking*, a technique that has enjoyed considerable successes in recent years, first with hardware verification and now with software. Model checking is similar to exhaustive testing in that it attempts to explore all execution paths of a program; but it does so on a simplified version of the program (to make the search tractable) and uses advanced proof techniques to limit the search space and make the results significant. One may further contrast model checking with *abstract interpretation*, also used successfully in large industrial projects; here too, some of the key contributors to both approaches were active participants in the conference and this book.

Competition turning into cooperation, "exclusive or" replaced by "and," initial contradictions revealing complementarity the deeper one digs: this could be the theme of the present book. By exploring the authors' often contrasted contributions, you will discover both the multiplicity of viewpoints that exist today in the field of software verification and the numerous common themes and solutions that justify Tony Hoare's injunction: to join forces and all get to work together.

Acknowledgments. Many people made VSTTE and this volume possible. As Program Chair, Natarajan Shankar composed and orchestrated the event. We are of course particularly grateful to the authors, panelists and other participants who contributed their best work.

ETH Zurich (as part of its 150th anniversary celebrations) contributed a generous donation; so did Microsoft Research. We are particularly indebted to Professor Meinrad Eberle, coordinator of the ETH 150th anniversary, and Dr. Andrew Herbert, Managing Director of the Microsoft Research center in Cambridge.

ETH provided superb infrastructure and services, allowing in particular the recording of sessions as reflected in this book. The organizing team from the Chair of Software Engineering—Claudia Günthart, Bernd Schoeller, Ilinca Ciupa, Andreas Leitner, Manuel Oriol, Werner Dietl, Adam Darvas, Farhad Mehta, Luc de Louw and a number of student helpers—was essential to the preparation of the conference and its smooth operation, including the recordings. The grueling task of transcribing these recordings was performed with remarkable efficiency by Patrick Schönbach. Claudia Günthart helped put the book in its final form.

Our primary debt is to the two conference Chairs, Tony Hoare and Jay Misra, for not only conceiving the vision but also relentlessly pursuing its realization down to the most practical details, with an energy that never ceases to amaze us all. We hope that readers of this book will feel some of that energy, and share some of the excitement that pervaded every session of the VSTTE conference.

February 2008

Bertrand Meyer
Jim Woodcock

VSTTE Chairs and Committees

Joint General Chairs

Tony Hoare, Microsoft Research, UK
Jayadev Misra, University of Texas at Austin, USA

Program Chair

Natarajan Shankar, SRI, USA

Steering Committee

Manfred Broy, TU München, Germany
Butler Lampson, Microsoft Research, UK
Mathai Joseph, Tata Research Development and Design Centre, India
Gilles Kahn, INRIA, France
J Moore, University of Texas at Austin, USA
Amir Pnueli, Weizmann Inst. of Science, Israel
Michel Sintzoff, U. Cath. de Louvain, Belgium
Zhou Chao Chen, Chinese Acad. of Sciences

Program Committee

Jean-Raymond Abrial, ETH Zurich, Switzerland
Dines Bjørner, Technical Univ. of Denmark
Patrick Cousot, École Normale Supérieure, France
Michael Ernst, Mass. Inst. of Technology, USA
David Evans, Univ. of Virginia, USA
David Gries, Cornell University, USA
Orna Grumberg, Israel Inst. of Technology
Masami Hagiya, Univ. of Tokyo, Japan
Ian Hayes, Univ. of Queensland, Australia
Gerard Holzmann, Jet Propulsion Laboratory, USA
Cliff Jones, Univ. of Newcastle, UK
Greg Morrisett, Harvard Univ., UK
George Necula, UC Berkeley, USA
Greg Nelson, HP Systems Research Center, USA
Tobias Nipkow, TU München, Germany
Colin O'Halloran, QinetiQ, UK
Ernst-Rüdiger Olderog, Univ. Oldenburg, Germany
Mooly Sagiv, Tel-Aviv University, Israel

He Jifeng, United Nations Univ., Macao
Connie Heitmeyer, Naval Research Laboratory, USA
Carolyn Talcott, SRI International, USA
Jian Zhang, Academia Sinica, China

Publication Chair

Jim Woodcock, University of York, UK

Publication Committee

David Gries, Cornell University, USA
Cliff Jones, Newcastle, UK
Bertrand Meyer, ETH Zurich, Switzerland
Ernst-Rüdiger Olderog, Univ. Oldenburg, Germany

Organization Chair

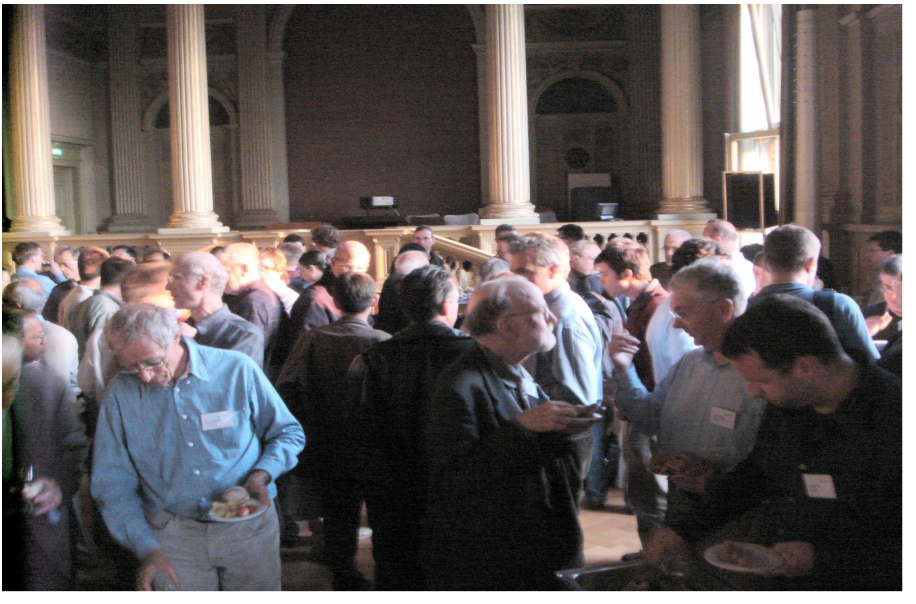
Bertrand Meyer, ETH Zurich, Switzerland

Organization Committee

Bernd Schoeller, ETH Zurich, Switzerland
Claudia Günthart, ETH Zurich, Switzerland

Tony Hoare, opening session

Rajeev Joshi, Joseph Kiniry, Greg Nelson, Natarajan Shankar, Jay Misra, Tony Hoare, closing panel



Welcome party in the historic Semper-Aula of the ETH

Panagiotis Manolios, Wolfgang Paul, Amir Pnueli

Cliff Jones, Connie Heitmeyer

Anthony Hall, Rod Chapman
in the "VSTTE Express" (chartered train to Üetliberg for conference dinner)

Jay Misra, Niklaus Wirth, Tony Hoare

Photographs by Bertrand Meyer

Participants

Jean-Raymond Abrial, ETH Zurich, Switzerland
Rajeev Alur, University of Pennsylvania, USA
Myla Archer, Naval Research Laboratory, USA
Ralph Back, Abo Akademi University, Finland
Thomas Ball, Microsoft Research, USA
David Basin, ETH Zurich, Switzerland
Yves Bertot, INRIA, France
Ramesh Bharadwaj, Naval Research Laboratory, USA
Egon Börger, University of Pisa, Italy
Manfred Broy, TU Munich, Germany
Tevfik Bultan, University of California, USA
Michael Butler, University of Southampton, UK
Supratik Chakraborty, Indian Institute of Technology, India
Patrice Chalin, Concordia University, Canada
Roderick Chapman, Praxis High Integrity Systems, UK
Marsha Chechik, University of Toronto, Canada
Alessandro Coglio, Kestrel Institute, USA
Patrick Cousot, École Normale Supérieure, France
Michael Ernst, MIT, USA
Kathi Fisler, Worcester Polytechnic Institute, USA

Kokichi Futatsugi, JAIST, Japan
Allen Goldberg, Kestrel Technology LLC, USA
Cordell Green, Kestrel Institute, USA
Arie Gurfinkel, University of Toronto, Canada
Masami Hagiya, University of Tokyo, Japan
Anthony Hall, Oxford, UK
Stefan Hallerstede, ETH Zurich, Switzerland
Klaus Havelund, Kestrel Technology LLC, USA
Eric Hehner, University of Toronto, Canada
Constance Heitmeyer, Naval Research Laboratory, USA
Michael G. Hinchey, NASA, USA
Thai Son Hoang, ETH Zurich, Switzerland
Tony Hoare, Microsoft Research, UK
Gerard J. Holzmann, NASA Jet Propulsion Laboratory, USA
Peter Vincent Homeier, NSA, USA
Andrew Ireland, Heriot-Watt University, UK
Bart Jacobs, Radboud University Nijmegen, Netherlands
He Jifeng, UNU-IIST, China
Cliff Jones, Newcastle University, UK
Mathai Joseph, Tata Consultancy Services, India
Rajeev Joshi, NASA/JPL Laboratory for Reliable Software, USA
Joseph Kiniry, University College Dublin, Ireland
Daniel Kröning, ETH Zurich, Switzerland
Patrick Lam, Massachusetts Institute of Technology, USA
Gary T. Leavens, Iowa State University, USA
Rustan Leino, Microsoft Research, USA
Zhiming Liu, UNU-IIST, Macao, China
Panagiotis Manolios, Georgia Tech, USA
Tiziana Margaria, University of Göttingen, Germany
Edu Metz, Nokia Research Center, USA
Bertrand Meyer, ETH Zurich, Switzerland
Jayadev Misra, University of Texas at Austin, USA
J. Strother Moore, University of Texas, USA
Peter Müller, ETH Zurich, Switzerland
David A. Naumann, Stevens Institute of Technology, USA
Greg Nelson, USA
Colin O'Halloran, QinetiQ, UK
Peter O'Hearn, Queen Mary, University of London, UK
Ernst-Rüdiger Olderog, University of Oldenburg, Germany
Manuel Oriol, ETH Zurich, Switzerland
Wolfgang Paul, University of Saarbrücken, Germany
Amir Pnueli, Weizmann Institute, Israel, and New York University, USA
Sanjiva Prasad, Indian Institute of Technology, India
Ganesan Ramalingam, IBM Research, India
Thomas Reps, University of Wisconsin-Madison, USA
Tamara Rezk, INRIA, France
Martin Rinard, Massachusetts Institute of Technology, USA
Willem Paul de Roever, University of Kiel, Germany
Grigore Rosu, University of Illinois, USA
Harald Ruess, SRI International, USA
John Rushby, SRI International, USA
Shmuel Sagiv, Tel-Aviv University, Israel
Peter Schmitt, University of Karlsruhe, Germany
Florian Schneider, ETH Zurich, Switzerland
Wolfram Schulte, Microsoft Research, USA
Natarajan Shankar, SRI International, USA
Natasha Sharygina, Carnegie Mellon University, USA

Michel Sintzoff, Université Catholique de Louvain, Belgium
Douglas R. Smith, Kestrel Institute, USA
Graham Steel, University of Edinburgh, UK
Bernhard Steffen, University Dortmund, Germany
Ofer Strichman, Technion, Israel
Aaron Stump, Washington University, USA
Carolyn Talcott, SRI International, USA
Cesare Tinelli, University of Iowa, USA
Mark Utting, University of Waikato, New Zealand
Eric Van Wyk, University of Minnesota, USA
Helmut Veith, TU Munich, Germany
Arnaud Venet, Kestrel Technology, USA
Laurent Voisin, ETH Zurich, Switzerland
Georg Weissbacher, ETH Zurich, Switzerland
Niklaus Wirth, ETH Zurich, Switzerland
Jim Woodcock, University of York, UK
Stephan Zdancewic, University of Pennsylvania, USA
Naijun Zhan, Chinese Academy of Sciences
Jian Zhang, Chinese Academy of Sciences
Jianjun Zhao, Fukuoka Institute of Technology, Japan

Table of Contents

Verifying Object-Oriented Programming

Programming Language and Methodology Aspects

Components

Static Analysis

Design, Analysis and Tools

Formal Techniques

Position Papers