John Fitzgerald
Ian J. Hayes
Andrzej Tarlecki (Eds.)

LNCS 3582

FM 2005: Formal Methods

International Symposium of Formal Methods Europe
Newcastle, UK, July 2005
Proceedings

Springer

John Fitzgerald Ian J. Hayes
Andrzej Tarlecki (Eds.)

FM 2005: Formal Methods

International Symposium of Formal Methods Europe
Newcastle, UK, July 18-22, 2005
Proceedings

Springer

Volume Editors

John Fitzgerald
University of Newcastle upon Tyne
Centre for Software Reliability
Newcastle upon Tyne, NE1 7RU, UK
E-mail: john.fitzgerald@ncl.ac.uk

Ian J. Hayes
University of Queensland
School of Information Technology and Electrical Engineering
Brisbane, QLD 4072, Australia
E-mail: Ian.Hayes@itee.uq.edu.au

Andrzej Tarlecki
Warsaw University
Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warszawa, Poland
E-mail: tarlecki@mimuw.edu.pl

Library of Congress Control Number: 2005928720

CR Subject Classification (1998): D.2, F.3, D.3, D.1, J.1, K.6, F.4

ISSN 0302-9743
ISBN-10 3-540-27882-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-27882-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11526841 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science

For information about Vols. 1–3481

please contact your bookseller or Springer

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3562: J. Mira, J.R. Álvarez (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Part II. XXIV, 636 pages. 2005.

Vol. 3561: J. Mira, J.R. Álvarez (Eds.), Mechanisms, Symbols, and Models Underlying Cognition, Part I. XXIV, 532 pages. 2005.

Vol. 3560: V.K. Prasanna, S. Iyengar, P.G. Spirakis, M. Welsh (Eds.), Distributed Computing in Sensor Systems. XV, 423 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005. (Subseries LNAI).

Vol. 3557: H. Gilbert, H. Handschuh (Eds.), Fast Software Encryption. XI, 443 pages. 2005.

Vol. 3556: H. Baumeister, M. Marchesi, M. Holcombe (Eds.), Extreme Programming and Agile Processes in Software Engineering. XIV, 332 pages. 2005.

Vol. 3555: T. Vardanega, A. Wellings (Eds.), Reliable Software Technology – Ada-Europe 2005. XV, 273 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005. (Subseries LNAI).

Vol. 3553: T.D. Hämäläinen, A.D. Pimentel, J. Takala, S. Vassiliadis (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 476 pages. 2005.

Vol. 3552: H. de Meer, N. Bhatti (Eds.), Quality of Service – IWQoS 2005. XVIII, 400 pages. 2005.

Vol. 3551: T. Härder, W. Lehner (Eds.), Data Management in a Connected World. XIX, 371 pages. 2005.

Vol. 3548: K. Julisch, C. Kruegel (Eds.), Intrusion and Malware Detection and Vulnerability Assessment. X, 241 pages. 2005.

Vol. 3547: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. XIII, 588 pages. 2005.

Vol. 3543: L. Kutvonen, N. Alonistioti (Eds.), Distributed Applications and Interoperable Systems. XI, 235 pages. 2005.

Vol. 3541: N.C. Oza, R. Polikar, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. XII, 430 pages. 2005.

Vol. 3540: H. Kalviainen, J. Parkkinen, A. Kaarna (Eds.), Image Analysis. XXII, 1270 pages. 2005.

Vol. 3537: A. Apostolico, M. Crochemore, K. Park (Eds.), Combinatorial Pattern Matching. XI, 444 pages. 2005.

Vol. 3536: G. Ciardo, P. Darondeau (Eds.), Applications and Theory of Petri Nets 2005. XI, 470 pages. 2005.

Vol. 3535: M. Steffen, G. Zavattaro (Eds.), Formal Methods for Open Object-Based Distributed Systems. X, 323 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005. (Subseries LNAI).

Vol. 3532: A. Gómez-Pérez, J. Euzenat (Eds.), The Semantic Web: Research and Applications. XV, 728 pages. 2005.

Vol. 3531: J. Ioannidis, A. Keromytis, M. Yung (Eds.), Applied Cryptography and Network Security. XI, 530 pages. 2005.

Vol. 3530: A. Prinz, R. Reed, J. Reed (Eds.), SDL 2005: Model Driven. XI, 361 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005. (Subseries LNAI).

Vol. 3527: R. Morrison, F. Oquendo (Eds.), Software Architecture. XII, 263 pages. 2005.

Vol. 3526: S.B. Cooper, B. Löwe, L. Torenvliet (Eds.), New Computational Paradigms. XVII, 574 pages. 2005.

Vol. 3525: A.E. Abdallah, C.B. Jones, J.W. Sanders (Eds.), Communicating Sequential Processes. XIV, 321 pages. 2005.

Vol. 3524: R. Barták, M. Milano (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 320 pages. 2005.

Vol. 3523: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part II. XXVI, 733 pages. 2005.

Vol. 3522: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis, Part I. XXVI, 703 pages. 2005.

Vol. 3521: N. Megiddo, Y. Xu, B. Zhu (Eds.), Algorithmic Applications in Management. XIII, 484 pages. 2005.

Vol. 3520: O. Pastor, J. Falcão e Cunha (Eds.), Advanced Information Systems Engineering. XVI, 584 pages. 2005.

Vol. 3519: H. Li, P. J. Olver, G. Sommer (Eds.), Computer Algebra and Geometric Algebra with Applications. IX, 449 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005. (Subseries LNAI).

Vol. 3517: H.S. Baird, D.P. Lopresti (Eds.), Human Interactive Proofs. IX, 143 pages. 2005.

Vol. 3516: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part III. LXIII, 1143 pages. 2005.

Vol. 3515: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part II. LXIII, 1101 pages. 2005.

Vol. 3514: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part I. LXIII, 1089 pages. 2005.

Vol. 3513: A. Montoyo, R. Muñoz, E. Métais (Eds.), Natural Language Processing and Information Systems. XII, 408 pages. 2005.

Vol. 3512: J. Cabestany, A. Prieto, F. Sandoval (Eds.), Computational Intelligence and Bioinspired Systems. XXV, 1260 pages. 2005.

Vol. 3511: U.K. Wiil (Ed.), Metainformatics. VIII, 221 pages. 2005.

Vol. 3510: T. Braun, G. Carle, Y. Koucheryavy, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communications. XIV, 366 pages. 2005.

Vol. 3509: M. Jünger, V. Kaibel (Eds.), Integer Programming and Combinatorial Optimization. XI, 484 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005. (Subseries LNAI).

Vol. 3507: F. Crestani, I. Ruthven (Eds.), Information Context: Nature, Impact, and Role. XIII, 253 pages. 2005.

Vol. 3506: C. Park, S. Chee (Eds.), Information Security and Cryptology – ICISC 2004. XIV, 490 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V. A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005. (Subseries LNAI).

Vol. 3504: A.F. Frangi, P.I. Radeva, A. Santos, M. Hernandez (Eds.), Functional Imaging and Modeling of the Heart. XV, 489 pages. 2005.

Vol. 3503: S.E. Nikoletseas (Ed.), Experimental and Efficient Algorithms. XV, 624 pages. 2005.

Vol. 3502: F. Khendek, R. Dssouli (Eds.), Testing of Communicating Systems. X, 381 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005. (Subseries LNAI).

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005. (Subseries LNBI).

Vol. 3499: A. Pelc, M. Raynal (Eds.), Structural Information and Communication Complexity. X, 323 pages. 2005.

Vol. 3498: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part III. XLIX, 1077 pages. 2005.

Vol. 3497: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part II. XLIX, 947 pages. 2005.

Vol. 3496: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part II. L, 1055 pages. 2005.

Vol. 3495: P. Kantor, G. Muresan, F. Roberts, D.D. Zeng, F.-Y. Wang, H. Chen, R.C. Merkle (Eds.), Intelligence and Security Informatics. XVIII, 674 pages. 2005.

Vol. 3494: R. Cramer (Ed.), Advances in Cryptology – EUROCRYPT 2005. XIV, 576 pages. 2005.

Vol. 3493: N. Fuhr, M. Lalmas, S. Malik, Z. Szlávik (Eds.), Advances in XML Information Retrieval. XI, 438 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005. (Subseries LNAI).

Vol. 3489: G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 358 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005. (Subseries LNAI).

Vol. 3486: T. Helleseth, D. Sarwate, H.-Y. Song, K. Yang (Eds.), Sequences and Their Applications - SETA 2004. XII, 451 pages. 2005.

Vol. 3483: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part IV. LXV, 1362 pages. 2005.

Vol. 3482: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part III. LXV, 1340 pages. 2005.

¥641.92元

Preface

This volume contains the proceedings of Formal Methods 2005, the 13th International Symposium on Formal Methods held in Newcastle upon Tyne, UK, during July 18–22, 2005. Formal Methods Europe (FME, www.fmeurope.org) is an independent association which aims to stimulate the use of, and research on, formal methods for system development. FME conferences began with a VDM Europe symposium in 1987. Since then, the meetings have grown and have been held about once every 18 months. Throughout the years the symposia have been notably successful in bringing together researchers, tool developers, vendors, and users, both from academia and from industry. Formal Methods 2005 confirms this success.

We received 130 submissions to the main conference, from all over the world. Each submission was carefully refereed by at least three reviewers. Then, after an intensive, in-depth discussion, the Program Committee selected 31 papers for presentation at the conference. They form the bulk of this volume. We would like to thank all the Program Committee members and the referees for their excellent and efficient work.

Apart from the selected contributions, the Committee invited three keynote lectures from Mathai Joseph, Marie-Claude Gaudel and Chris Johnson. You will find the abstracts/papers for their keynote lectures in this volume as well.

An innovation for the FM 2005 program was a panel discussion on the history of formal methods, with Jean-Raymond Abrial, Dines Bjørner, Jim Horning and Cliff Jones as panelists. Unfortunately, it was not possible to reflect this event in the current volume, but you will find the material documenting it elsewhere (see the conference Web page).

An Industry Day was organized by the Formal Techniques Industrial Association (ForTIA) alongside the main symposium. This was directly related to the main theme of the FM symposia: the use of well-founded formal methods in the industrial practice of software design, development and maintenance. We have therefore included abstracts of the invited presentations in this volume as well.

The main FM 2005 conference was accompanied by 9 workshops and 11 tutorials.

The electronic submission, refereeing and Program Committee discussions would not have been possible without software support. We worked with the OCS system developed at the University of Dortmund — our thanks to the staff there for their support.

Finally, we would like to thank all those who helped to create and run the symposium in Newcastle, and in particular Claire Smith, Jon Warwick, Joan Atkinson, Sarah Davidson, Nigel Jefferson, Joey Coleman, Jeremy Bryans, Neil Henderson and Juan Bicarregui for their help in bringing the program, and these proceedings, together.

July 2005 John Fitzgerald, Ian Hayes, Andrzej Tarlecki

Organization

FM 2005 was organized by the Centre for Software Reliability at the University of Newcastle upon Tyne (www.csr.ncl.ac.uk) and Formal Methods Europe. We are grateful for the support of the University of Newcastle and its School of Computing Science. Within Formal Methods Europe, we are particularly grateful to Kees Pronk and Stefania Gnesi for their help with budgeting and organization. We also gladly acknowledge direct sponsorship from SAP Research and the British Computer Society Specialist Group on Formal Aspects of Computing Science (BCS-FACS).

Conference Chairs

General Chair	John S. Fitzgerald, University of Newcastle, UK
Program Co-chairs	Ian Hayes, University of Queensland, Australia
	Andrzej Tarlecki, Warsaw University, Poland
Conference Organizer	Claire Smith, University of Newcastle, UK
Finance Chair	Jon Warwick, University of Newcastle, UK
Tools Exhibition Chair	Joan Atkinson, University of Newcastle, UK
Workshops Chair	Juan Bicarregui, Rutherford Appleton Laboratory, UK
Tutorials Chair	Neil Henderson, University of Newcastle, UK

Program Committee

Bernhard Aichernig, UNU-IIST, Macau, China
Keijiro Araki, Kyushu University, Japan
Juan Bicarregui, Rutherford Appleton Laboratory, UK
Michel Bidoit, LSV, CNRS and ENS de Cachan, France
Ed Brinksma, University of Twente, The Netherlands
Luca Cardelli, Microsoft Research, UK
Ernie Cohen, Microsoft, USA
Jin Song Dong, National University of Singapore, Singapore
José Luiz Fiadeiro, University of Leicester, UK
John S. Fitzgerald, Centre for Software Reliability, UK
Stefania Gnesi, CNR, Italy
Anthony Hall, UK
Anne E. Haxthausen, Technical University of Denmark, Denmark
Ian Hayes, University of Queensland, Australia (Co-chair)
Thomas A. Henzinger, EPFL and University of California, Berkeley, USA
He Jifeng, UNU-IIST, Macau, China
Cliff Jones, University of Newcastle, UK

Shaoying Liu, Hosei University, Japan
Mícheál Mac an Airchinnigh, Trinity College Dublin, Ireland
Tom Maibaum, McMaster University, Canada
Dino Mandrioli, Politecnico di Milano, Italy
Tobias Nipkow, Technische Universität München, Germany
José Oliveira, Universidade do Minho, Portugal
Sam Owre, CRI, USA
Alexander Petrenko, ISPRAS, Russia
Nico Plat, West Consulting, The Netherlands
Ken Robinson, University of New South Wales, Australia
Mark Saaltink, ORA Canada, Canada
Shin Sahara, JFITS, Japan
Steve Schneider, University of Surrey, UK
Kaisa Sere, Åbo Akademi, Finland
Ketil Stølen, SINTEF, Norway
Andrzej Tarlecki, Warsaw University, Poland (Co-chair)
Mark Utting, Waikato University, New Zealand
Marcel Verhoef, Chess IT and Radboud University, Nijmegen, Netherlands
Alan Wassyng, McMaster University, Canada
Martin Wirsing, Ludwig-Maximilians-Universität, München, Germany

Referees

Carlos Bacelar Almeida
Paulo Sergio Almeida
Matthias Anlauff
Alvaro Arenas
Alexei Barantsev
Luis Barbosa
Leonor Barroca
Hubert Baumeister
Marek A. Bednarczyk
Maurice ter Beek
Axel Belinfante
Dirk Beyer
Machiel van der Bijl
Henrik Bohnenkamp
Pontus Boström
Ahmed Bouajjani
Patricia Bouyer
Folker den Braber
Laura Brandan Briones
Phil Brooke
Roberto Bruni
Hans Bruun
Gyrd Brændeland
Bettina Buth
Jens Bæk Jørgensen
Jacques Carette
David Carrington
Arindam Chakrabarti
Michel Chaudron
Chunqing Chen
Jacek Chrząszcz
David Clark
Joey Coleman
Phil Cook
Véronique Cortier
Jorge Cuellar
Roberto Delicata
Dubravka Ilic
Bruno Dutertre
Neil Evans
Alessandro Fantechi
Gianluigi Ferrari
Paul Fischer
Oana Florescu
Martin Fränzle
Laurent Fribourg
Carlo Furia
Peter Gorm Larsen
Jean Goubault-Larrecq
Adriaan de Groot
Stefan Gruner
Moritz Hammer
Ping Hao
Neil Henderson
Martijn Hendriks
Thai Son Hoang
Martin Hofmann
Jozef Hooman
Dang Van Hung
Wilson Ifill
Ryszard Janicki
Tomasz Janowski
Einar Broch Johnsen
Wolfram Kahl
Alexander Kamkin
Ridha Khedri

Victor Khomenko
Alexander Knapp
Erwin van der Koogh
Evgeny Kornykhin
Piotr Kosiuczenko
Fred Kröger
Steve Kremer
Victor Kuliamin
Alexander Kurz
Linas Laibinis
Christian Lange
Rom Langerak
Franiçois Laroussinie
Diego Latella
Timo Latvala
Christian Lengauer
Yuan Fang Li
Quan Long
Mass Soldal Lund
Volkmar Lotz
Hans Henrik Løvengreen
Tom Lysemose
Qaisar Ahmad Malik
Tiziana Margaria
Nicolas Markey
Mieke Massink
Brian Matthews
Franco Mazzanti
Alistair McEwan
Robert Meolic
Stephan Merz
Ali Mesbah
Tim Miller
Leonardo de Moura
Henry Muccini
Damian Niwiński
David von Oheim
Nickolay Pakulin
Jun Pang
Dirk Pattinson
Jan Peleska
Luigia Petre
Laure Petrucci
Nir Piterman
David Pitt
Matteo Pradella
Kees Pronk
Axel Rauschmayer
Atle Refsdal
Brian Ritchie
Markus Roggenbach
Judith Rossebø
Matteo Rossi
Ragnhild Kobro Runde
John Rushby
Theo Ruys
Denis Sabatier
Hassen Saidi
Thomas Santen
Bernhard Schätz
Norbert Schirmer
Aleksy Schubert
Fredrik Seehusen
Emil Sekerinski
Natarajan Shankar
Mike Shields
Graeme Smith
Monika Solanki
Bjørnar Solhaug
Jorge Sousa Pinto
Simao Melo de Sousa
Pieter van der Spek
Paola Spoletini
Mariëlle Stoelinga
Asuman Suenbuel
Jun Sun
Helen Treharne
Jan Tretmans
Leonidas Tsiopoulos
Irek Ulidowski
Neeraj Verma
Joost Visser
Peter Visser
Fredrik Vraalsen
Marina Waldén
Burkhart Wolff
Lu Yan
Yuwen Yang

Sponsors

Table of Contents

Keynote Talks

Formal Aids for the Growth of Software Systems
Mathai Joseph 1

Formal Methods and Testing: Hypotheses, and Correctness Approximations
Marie-Claude Gaudel 2

The Natural History of Bugs: Using Formal Methods to Analyse Software Related Failures in Space Missions
C.W. Johnson 9

Object Orientation

Modular Verification of Static Class Invariants
K. Rustan M. Leino, Peter Müller 26

Decoupling in Object Orientation
Ioannis T. Kassios 43

Controlling Object Allocation Using Creation Guards
Cees Pierik, Dave Clarke, Frank S. de Boer 59

Symbolic Animation of JML Specifications
Fabrice Bouquet, Frédéric Dadeau, Bruno Legeard, Mark Utting 75

Resource Analysis and Verification

Certified Memory Usage Analysis
David Cachera, Thomas Jensen, David Pichardie, Gerardo Schneider 91

Compositional Specification and Analysis of Cost-Based Properties in Probabilistic Programs
Orieta Celiku, Annabelle McIver 107

Formally Defining and Verifying Master/Slave Speculative Parallelization
Pierre Salverda, Grigore Roşu, Craig Zilles 123

Timing and Testing

Systematic Implementation of Real-Time Models
Martin De Wulf, Laurent Doyen, Jean-François Raskin 139

Timing Tolerances in Safety-Critical Software
Alan Wassyng, Mark Lawford, Xiayong Hu 157

Timed Testing with TorX
Henrik Bohnenkamp, Axel Belinfante 173

Automatic Verification and Conformance Testing for Validating Safety Properties of Reactive Systems
Vlad Rusu, Hervé Marchand, Thierry Jéron 189

CSP, B and Circus

Adding Conflict and Confusion to CSP
Christie Bolton 205

Combining CSP and B for Specification and Property Verification
Michael Butler, Michael Leuschel 221

Operational Semantics for Model Checking Circus
Jim Woodcock, Ana Cavalcanti, Leonardo Freitas 237

Control Law Diagrams in *Circus*
Ana Cavalcanti, Phil Clayton, Colin O'Halloran 253

Security

Verification of a Signature Architecture with HOL-Z
David Basin, Hironobu Kuruma, Kazuo Takaragi, Burkhart Wolff ... 269

End-to-End Integrated Security and Performance Analysis on the DEGAS Choreographer Platform
Mikael Buchholtz, Stephen Gilmore, Valentin Haenel, Carlo Montangero ... 286

Formal Verification of Security Properties of Smart Card Embedded Source Code
June Andronick, Boutheina Chetali, Christine Paulin-Mohring 302

Networks and Processes

A Formal Model of Addressing for Interoperating Networks
Pamela Zave 318

An Approach to Unfolding Asynchronous Communication Protocols
Yu Lei, S. Purushothaman Iyer 334

Semantics of BPEL4WS-Like Fault and Compensation Handling
Qiu Zongyan, Wang Shuling, Pu Geguang, Zhao Xiangpeng 350

Abstraction, Retrenchment and Rewriting

On Some Galois Connection Based Abstractions for the Mu-Calculus
Dragan Bošnački 366

Retrenching the Purse: Finite Sequence Numbers, and the Tower Pattern
Richard Banach, Michael Poppleton, Czeslaw Jeske, Susan Stepney 382

Strategic Term Rewriting and Its Application to a VDM-SL to SQL Conversion
T.L. Alves, P.F. Silva, J. Visser, J.N. Oliveira 399

Scenarios and Modeling Languages

Synthesis of Distributed Processes from Scenario-Based Specifications
Jun Sun, Jin Song Dong 415

Verifying Scenario-Based Aspect Specifications
Emilia Katz, Shmuel Katz 432

An MDA Approach Towards Integrating Formal and Informal Modeling Languages
Soon-Kyeong Kim, Damian Burger, David Carrington 448

Model Checking

Model-Checking of Specifications Integrating Processes, Data and Time
Jochen Hoenicke, Patrick Maier 465

Automatic Symmetry Detection for Model Checking Using Computational Group Theory
A.F. Donaldson, A. Miller 481

On Partitioning and Symbolic Model Checking
Subramanian Iyer, Debashis Sahoo, E. Allen Emerson, Jawahar Jain 497

Dynamic Component Substitutability Analysis
Natasha Sharygina, Sagar Chaki, Edmund Clarke, Nishant Sinha 512

Industry Day: Abstracts of Invited Talks

Floating-Point Verification
John Harrison 529

Preliminary Results of a Case Study: Model Checking for Advanced Automotive Applications
Stefan Eisler, Christian Scheidler, Bernhard Josko, Guido Sandmann, Joachim Stroop 533

Model-Based Testing in Practice
Alexander Pretschner 537

Testing Concurrent Object-Oriented Systems with Spec Explorer
Colin Campbell, Wolfgang Grieskamp, Lev Nachmanson, Wolfram Schulte, Nikolai Tillmann, Margus Veanes 542

ASD Case Notes: Costs and Benefits of Applying Formal Methods to Industrial Control Software
Guy H. Broadfoot 548

The Informal Nature of Systems Engineering
Gerrit Muller 552

Author Index 557

Formal Aids for the Growth of Software Systems

Mathai Joseph

Tata Research Development & Design Centre,
A Division of Tata Consultancy Services
mathai.joseph@tcs.com

Abstract. The use of formal techniques has for a long time been focused on relatively small and complex applications. The hardware domain lends itself well to this and it has therefore been the target of some of the most significant applications of formal techniques. The software applications that have typically been considered were for small, safety-critical systems.

This restricted focus was understandable and necessary while formal techniques were evolving and practical considerations limited the size of the system that could be specified and verified. However, there are now compelling demands for the use of more precise techniques for a variety of large-scale applications, ranging from smart cards to financial systems.

So there are now new reasons to extend the use of formal methods for all phases of software development: from requirements and software modeling to coding and testing. Problems of scale still remain so it is important to focus the use of formal techniques in areas where their impact will be most important.

Different formal techniques can be used for solving different problems. For example, use of model-checking during requirements modeling can identify incomplete or inconsistent specifications, while use of transformational techniques can be very effective for software modeling and enable generation of code directly from models. Program analysis techniques can be used to generate tests that will greatly improve functional coverage during testing.

The use of formal techniques continues during *software maintenance* through the following kinds of activities:

a. **Remedial**: correction of errors discovered during use;
b. **Adaptive**: making changes to cater to changes in the operating environment;
c. **Enhancing**: adding new features or capabilities; and
d. **Improving**: making the software more robust and easier to maintain.

It is estimated that the cost of software maintenance amounts to as much as 90% of the life-cycle cost of a software system. While this calls for major improvements in maintenance techniques, changes in software development methods can also help to reduce the need for, and therefore the cost of, making remedial improvements (i.e. bug fixing).

In this talk, I will describe the use of formal techniques for different areas of the software life-cycle and relate this to evidence obtained through the analysis of a large number of actual software development and maintenance projects.

J.S. Fitzgerald, I.J. Hayes, and A. Tarlecki (Eds.): FM 2005, LNCS 3582, p. 1, 2005

Formal Methods and Testing: Hypotheses, and Correctness Approximations

Marie-Claude Gaudel

LRI, Paris-Sud University & CNRS, Orsay, France
mcg@lri.fr

Abstract. It has been recognised for a while that formal specifications can bring much to software testing. Numerous methods have been proposed for the derivation of test cases from various kinds of formal specifications, their submission, and verdict. All these methods rely upon some hypotheses on the system under test that formalise the gap between the success of a test campaign and the correctness of the system under test.

1 Introduction

It has been recognised for a while that formal specifications and models can bring much to software testing [16], [10]. In this extended abstract, we first precisely introduce the distinction between specification testing, model checking, and implementation testing based on formal specifications. Then we focus on the specificities of the latter one.

Actually, embedding implementation testing within a formal framework is far from being obvious. One tests a system. A system is a dynamic entity. It raises tricky issues such as observability and controllability, and sometimes specific physical constraints. A system is not a formula, even if it can be partially described as such. Thus, testing is very different from program proving, even if it is related. Similarly, testing is different from model checking, where verifications are performed on a known model: when testing, the model corresponding to the system under test is unknown (if it was known, testing would not be necessary...) and it is sometimes difficult to observe in what state it is [20], [22]. These points have been successfully circumvented in several testing methods based on formal specifications (or models) that use various and diverse techniques such as graph theory, symbolic evaluation, proof techniques, constraint solving, static analysis or model checking.

Explicitly or not, all these methods rely upon hypotheses on the system under test. They provide some approximation of correctness that is correlated to these hypotheses. In this talk we recall the notions of testability hypotheses and selection hypotheses that were introduced in [4], and we show how they have been used or could be used on various kinds of formal methods. We also address the issues of observation and control of the system under test.

J.S. Fitzgerald, I.J. Hayes, and A. Tarlecki (Eds.): FM 2005, LNCS 3582, pp. 2–8, 2005.

2 Testing Specifications, Checking Models, or Testing Implementations?

Before starting some discussion on formal methods and testing, it is necessary to introduce some terminology. Unfortunately, there is no consensus on these issues among the various research communities working in the area of software.

There is not even an agreement on the meanings of the words "validation" and "verification" [7] [3]. Similarly, the word "testing" is often used with different meanings.

Looking in a dictionary, one gets definitions such as:

"subjecting somebody or something to challenging difficulties"

In the case of software and formal methods, the "somebody or something" and the "challenging difficulties" are sometimes understood in different ways.

In most cases, the entity under test is a system, and the "challenging difficulties" are inputs, or sequences of inputs, aiming at revealing some dysfunctions [4], [8], [12] [13], [15], etc. In such cases, formal descriptions of the system are mainly used as guidelines for the selection of (sequences of) inputs and for the verdict. We focus on these approaches in Sections 3, 4 and 5.

2.1 Debugging or Testing Formal Specifications

In some other cases, testing is understood as debugging of formal descriptions or models. The formal description is the subject of the test. The challenges are either properties to be satisfied or refuted [14], or inputs for some simulation of the future system, based on the formal description [19], [11].

As the main characteristic of formal specifications is the ability of reasoning, theorem proving is used either to prove that a required property is a consequence of the specification, or to refute a property that corresponds to a forbidden situation. The choice of such challenges is far from being simple. It requires a very good expertise in the application domain. As the specification may be wrong, is probably a good idea to make this choice as independent of it as possible [2], even if some positive experiments have been performed on mutation of formal specifications [6].

2.2 Checking Models ... or Testing Them?

Model checking is similar in purpose: it aims at finding faults in so-called models of software systems. These models are behavioural (Kripke Structures, Finite Automata, Finite State Machine, Labelled Transition Systems, or even program control graphs), with a finite (but often huge) number of states labelled by atomic propositions. A model checker checks properties written in some temporal logic via an exhaustive exploration of the model, or some equivalent technique. Here also, the choice of the temporal properties to be checked is far from being obvious.

Model checking could be seen as a special kind of testing where the subject is a model and the challenges are temporal properties. Actually, there is some evolution in this direction. Due to the state explosion problem new techniques have been proposed that somewhat give up exhaustiveness: for instance, bounded model checking [5]