Kung-Kiu Lau
Richard Banach (Eds.)

LNCS 3785

Formal Methods and Software Engineering

7th International Conference
on Formal Engineering Methods, ICFEM 2005
Manchester, UK, November 2005, Proceedings

Springer

Kung-Kiu Lau Richard Banach (Eds.)

Formal Methods and Software Engineering

7th International Conference
on Formal Engineering Methods, ICFEM 2005
Manchester, UK, November 1-4, 2005
Proceedings

Springer

Volume Editors

Kung-Kiu Lau
Richard Banach
University of Manchester
School of Computer Science
Oxford Road, Manchester M13 9PL, UK
E-mail: {kung-kiu,banach}@cs.man.ac.uk

Library of Congress Control Number: 2005934587

CR Subject Classification (1998): D.2.4, D.2, D.3, F.3

ISSN 0302-9743
ISBN-10 3-540-29797-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29797-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11576280 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3785

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

Editorial Board

Preface

This volume contains papers presented at the 7th International Conference on Formal Engineering Methods (ICFEM 2005), 1–4 November 2005, Manchester, UK.

Formal engineering methods are changing the way that systems are developed. With language and tool support, these methods are being used for semi-automatic code generation, and for the automatic abstraction and checking of implementations. In the future, they will be used at every stage of development: requirements, specification, design, implementation, testing, and documentation.

The aim of ICFEM 2005 was to bring together those interested in the application of formal engineering methods to computer systems. Researchers and practitioners, from industry, academia, and government, were encouraged to attend, and to help advance the state of the art.

The conference was supported by sponsorships from Microsoft Research, USA, the Software Engineers Association of Japan, the University of Manchester, Manchester City Council, Formal Methods Europe (FME) and the British Computer Society Formal Aspects of Computing Specialist Group (BCS-FACS). We wish to thank these sponsors for their generosity.

The final programme consisted of 3 invited talks and 30 technical papers selected from a total of 74 submissions. The invited speakers were: Anthony Hall, independent consultant, UK; Egon Börger, University of Pisa, Italy; John Rushby, SRI, USA. Their talks were sponsored by BCS-FACS, Microsoft Research and FME respectively. We wish to thank the invited speakers for their inspiring talks.

Our heartfelt thanks go to all the members of the Programme Committee for their hard and conscientious work in reviewing and selecting the papers at various stages. I would also like to thank all the additional reviewers for their efforts and professionalism.

For organizing ICFEM 2005, we would like to thank the workshops and tutorials chair Mike Poppleton, the publicity chair Kenji Taguchi, the local organization chair Dave Lester, and the web-masters Elton Ballhysa and Faris Taweel. Their efforts were pivotal for the success of ICFEM 2005.

Finally, we would like to thank all the authors who submitted papers and all the conference attendees.

September 2005
Manchester

Kung-Kiu Lau and Richard Banach

Organization

Conference Chair

Richard Banach	University of Manchester, UK

Programme Chair

Kung-Kiu Lau	University of Manchester, UK

Programme Committee

Farhad Arbab	CWI and Leiden University, The Netherlands; University of Waterloo, Canada
Richard Banach	University of Manchester, UK
Luis Soares Barbosa	Minho University, Portugal
Mike Barnett	Microsoft Research, USA
Eerke Boiten	University of Kent, UK
Tommaso Bolognesi	CNR-ISTI, Italy
Marcello Bonsangue	Leiden University, The Netherlands
Jonathan P. Bowen	London South Bank University, UK
Manfred Broy	Technische Universität München, Germany
Bettina Buth	HAW Hamburg, Germany
Ana Cavalcanti	University of York, UK
Michel Charpentier	University of New Hampshire, USA
Jim Davies	University of Oxford, UK
Jin Song Dong	National University of Singapore, Singapore
Kai Engelhardt	University of New South Wales and NICTA, Australia
Colin Fidge	Queensland University of Technology, Australia
Mamoun Filali Amine	Université Paul Sabatier, France
John Fitzgerald	University of Newcastle upon Tyne, UK
Marc Frappier	Université de Sherbrooke, Canada
Dimitra Giannakopoulou	USRA/NASA Ames, USA
Chris George	United Nations University, China
Wolfgang Grieskamp	Microsoft Research, USA
Lindsay Groves	Victoria University of Wellington, New Zealand
Henri Habrias	Université de Nantes, France
Andrew Ireland	Heriot-Watt University, UK
Thomas Jensen	IRISA/CNRS, France
Soon-Kyeong Kim	University of Queensland, Australia
Steve King	University of York, UK
Rom Langerak	University of Twente, The Netherlands
James Larus	Microsoft Research, USA
Kung-Kiu Lau	University of Manchester, UK

Mark Lawford	McMaster University, Canada
Yves Ledru	LSR/IMAG, Domaine Universitaire, France
Peter A. Lindsay	University of Queensland, Australia
Shaoying Liu	Hosei University, Japan
Zhiming Liu	UNU-IIST, China
Brendan Mahony	Defence Science and Technology Organisation, Australia
Tiziana Margaria	University of Göttingen, Germany
Brad Martin	US Department of Defense, USA
Dominique Mery	Université Henri Poincaré Nancy 1, France
Huaikou Miao	Shanghai University, China
Alexandre Mota	Federal University of Pernambuco, Brazil
David Naumann	Stevens Institute of Technology, USA
Richard Paige	University of York, UK
Iman Poernomo	King's College London, UK
Fiona Polack	University of York, UK
Michael Poppleton	University of Southampton, UK
Steve Reeves	University of Waikato, New Zealand
Ken Robinson	University of New South Wales, Australia
Abhik Roychoudhury	National University of Singapore, Singapore
Harald Ruess	SRI International, USA
Motoshi Saeki	Tokyo Institute of Technology, Japan
Thomas Santen	Technische Universität Berlin, Germany
Klaus-Dieter Schewe	Massey University, New Zealand
Wolfram Schulte	Microsoft Research, USA
Kaisa Sere	Åbo Akademi University, Finland
Paul Strooper	University of Queensland, Australia
Asuman Suenbuel	SAP Research, USA
Paul A. Swatman	University of South Australia, Australia
Kenji Taguchi	National Institute of Informatics, Tokyo, Japan
Sofiene Tahar	Concordia University, Canada
Tetsuo Tamai	University of Tokyo, Japan
T.H. Tse	University of Hong Kong, China
Margus Veanes	Microsoft Research, USA
Charles Wallace	Michigan Technological University, USA
Farn Wang	National Taiwan University, Taiwan
Wang Yi	Uppsala University, Sweden
Jim Woodcock	University of York, UK

Additional Referees

Pascal André	Jean-Paul Bodeveix	Manuela Xavier
Marcelo Arenas	Jeremy Bryans	Robert Colvin
Christian Attiogbé	Michael Butler	Hugo ter Doest

Yuzhang Feng
David Faitelson
Benoit Fraikin
Amjad Gawanmeh
Frédéric Gervais
Irina Mariuca Gheorghita
Ben Gorry
Jens Grabowski
Olga Grinchtein
Juan Guillen-Scholten
Neil Henderson
Lutz Kettner
Bas Luttik
Tom Maibaum
Frank Marschall
Tim McComb
Sun Meng
Haja Moinudeen
Shin Nakamima
O. Nasr
Lemai Nguyen
Corina Pasareanu
Pascal Poizat
M. Rached
Rodrigo Ramos
Pritam Roy
Johann Schumann
Emil Sekerinski
Qin Shengchao
Rakesh Shukla
Colin Snook
Maria Sorea
Graham Steel
Bernhard Steffen
Jing Sun
Willem Visser
Nabil Wageeh
Marina Waldén
Geoffrey Watson
James Welch
Mohamed Zaki
Sergiy Zlatkin

Lecture Notes in Computer Science

For information about Vols. 1–3677

please contact your bookseller or Springer

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3775: J. Schoenwaelder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3772: M. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3770: J. Akoka, S.W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J.v.d. Heuvel, M. Kolp, J. Trujillo, C. Kop, H.C. Mayr (Eds.), Perspectives in Conceptual Modeling. XXII, 476 pages. 2005.

Vol. 3766: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 231 pages. 2005.

Vol. 3765: Y. Liu, T. Jiang, C. Zhang (Eds.), Computer Vision for Biomedical Image Applications. X, 563 pages. 2005.

Vol. 3762: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM Workshops. XXXI, 1228 pages. 2005.

Vol. 3761: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part II. XXVII, 653 pages. 2005.

Vol. 3760: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part I. XXVII, 921 pages. 2005.

Vol. 3759: G. Chen, Y. Pan, M. Guo, J. Lu (Eds.), Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. XIII, 669 pages. 2005.

Vol. 3758: Y. Pan, D. Chen, M. Guo, J. Cao, J. Dongarra (Eds.), Parallel and Distributed Processing and Applications. XXIII, 1162 pages. 2005.

Vol. 3756: J. Cao, W. Nejdl, M. Xu (Eds.), Advanced Parallel Processing Technologies. XIV, 526 pages. 2005.

Vol. 3754: J. Dalmau Royo, G. Hasegawa (Eds.), Management of Multimedia Networks and Services. XII, 384 pages. 2005.

Vol. 3752: N. Paragios, O. Faugeras, T. Chan, C. Schnoerr (Eds.), Variational, Geometric, and Level Set Methods in Computer Vision. XI, 369 pages. 2005.

Vol. 3751: T. Magedanz, E.R. M. Madeira, P. Dini (Eds.), Operations and Management in IP-Based Networks. X, 213 pages. 2005.

Vol. 3750: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part II. XL, 1018 pages. 2005.

Vol. 3749: J.S. Duncan, G. Gerig (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005, Part I. XXXIX, 942 pages. 2005.

Vol. 3747: C.A. Maziero, J.G. Silva, A.M.S. Andrade, F.M.d. Assis Silva (Eds.), Dependable Computing. XV, 267 pages. 2005.

Vol. 3746: P. Bozanis, E.N. Houstis (Eds.), Advances in Informatics. XIX, 879 pages. 2005.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martin-Sanchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005. (Subseries LNBI).

Vol. 3744: T. Magedanz, A. Karmouch, S. Pierre, I. Venieris (Eds.), Mobility Aware Technologies and Applications. XIV, 418 pages. 2005.

Vol. 3740: T. Srikanthan, J. Xue, C.-H. Chang (Eds.), Advances in Computer Systems Architecture. XVII, 833 pages. 2005.

Vol. 3739: W. Fan, Z. Wu, J. Yang (Eds.), Advances in Web-Age Information Management. XXIV, 930 pages. 2005.

Vol. 3738: V.R. Syrotiuk, E. Chávez (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XI, 360 pages. 2005.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005. (Subseries LNAI).

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005. (Subseries LNAI).

Vol. 3733: P. Yolum, T. Güngör, F. Gürgen, C. Özturan (Eds.), Computer and Information Sciences - ISCIS 2005. XXI, 973 pages. 2005.

Vol. 3731: F. Wang (Ed.), Formal Techniques for Networked and Distributed Systems - FORTE 2005. XII, 558 pages. 2005.

Vol. 3728: V. Paliouras, J. Vounckx, D. Verkest (Eds.), Integrated Circuit and System Design. XV, 753 pages. 2005.

Vol. 3726: L.T. Yang, O.F. Rana, B. Di Martino, J. Dongarra (Eds.), High Performance Computing and Communcations. XXVI, 1116 pages. 2005.

Vol. 3725: D. Borrione, W. Paul (Eds.), Correct Hardware Design and Verification Methods. XII, 412 pages. 2005.

Vol. 3724: P. Fraigniaud (Ed.), Distributed Computing. XIV, 520 pages. 2005.

Vol. 3723: W. Zhao, S. Gong, X. Tang (Eds.), Analysis and Modelling of Faces and Gestures. XI, 4234 pages. 2005.

Vol. 3722: D. Van Hung, M. Wirsing (Eds.), Theoretical Aspects of Computing – ICTAC 2005. XIV, 614 pages. 2005.

Vol. 3721: A. Jorge, L. Torgo, P. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005. (Subseries LNAI).

Vol. 3720: J. Gama, R. Camacho, P. Brazdil, A. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005. (Subseries LNAI).

Vol. 3719: M. Hobbs, A.M. Goscinski, W. Zhou (Eds.), Distributed and Parallel Computing. XI, 448 pages. 2005.

Vol. 3718: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XII, 502 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005. (Subseries LNAI).

Vol. 3716: L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, O. Pastor (Eds.), Conceptual Modeling – ER 2005. XVI, 498 pages. 2005.

Vol. 3715: E. Dawson, S. Vaudenay (Eds.), Progress in Cryptology – Mycrypt 2005. XI, 329 pages. 2005.

Vol. 3714: H. Obbink, K. Pohl (Eds.), Software Product Lines. XIII, 235 pages. 2005.

Vol. 3713: L. Briand, C. Williams (Eds.), Model Driven Engineering Languages and Systems. XV, 722 pages. 2005.

Vol. 3712: R. Reussner, J. Mayer, J.A. Stafford, S. Overhage, S. Becker, P.J. Schroeder (Eds.), Quality of Software Architectures and Software Quality. XIII, 289 pages. 2005.

Vol. 3711: F. Kishino, Y. Kitamura, H. Kato, N. Nagata (Eds.), Entertainment Computing - ICEC 2005. XXIV, 540 pages. 2005.

Vol. 3710: M. Barni, I. Cox, T. Kalker, H.J. Kim (Eds.), Digital Watermarking. XII, 485 pages. 2005.

Vol. 3709: P. van Beek (Ed.), Principles and Practice of Constraint Programming - CP 2005. XX, 887 pages. 2005.

Vol. 3708: J. Blanc-Talon, W. Philips, D.C. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems. XXII, 725 pages. 2005.

Vol. 3707: D.A. Peled, Y.-K. Tsay (Eds.), Automated Technology for Verification and Analysis. XII, 506 pages. 2005.

Vol. 3706: H. Fuks, S. Lukosch, A.C. Salgado (Eds.), Groupware: Design, Implementation, and Use. XII, 378 pages. 2005.

Vol. 3704: M. De Gregorio, V. Di Maio, M. Frucci, C. Musio (Eds.), Brain, Vision, and Artificial Intelligence. XV, 556 pages. 2005.

Vol. 3703: F. Fages, S. Soliman (Eds.), Principles and Practice of Semantic Web Reasoning. VIII, 163 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005. (Subseries LNAI).

Vol. 3701: M. Coppo, E. Lodi, G. M. Pinna (Eds.), Theoretical Computer Science. XI, 411 pages. 2005.

Vol. 3699: C.S. Calude, M.J. Dinneen, G. Păun, M. J. Pérez-Jiménez, G. Rozenberg (Eds.), Unconventional Computation. XI, 267 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005. (Subseries LNAI).

Vol. 3697: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Part II. XXXII, 1045 pages. 2005.

Vol. 3696: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations – ICANN 2005, Part I. XXXI, 703 pages. 2005.

Vol. 3695: M.R. Berthold, R. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005. (Subseries LNBI).

Vol. 3694: M. Malek, E. Nett, N. Suri (Eds.), Service Availability. VIII, 213 pages. 2005.

Vol. 3693: A.G. Cohn, D.M. Mark (Eds.), Spatial Information Theory. XII, 493 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatics. X, 436 pages. 2005. (Subseries LNBI).

Vol. 3691: A. Gagalowicz, W. Philips (Eds.), Computer Analysis of Images and Patterns. XIX, 865 pages. 2005.

Vol. 3690: M. Pěchouček, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005. (Subseries LNAI).

Vol. 3689: G.G. Lee, A. Yamada, H. Meng, S.H. Myaeng (Eds.), Information Retrieval Technology. XVII, 735 pages. 2005.

Vol. 3688: R. Winther, B.A. Gran, G. Dahll (Eds.), Computer Safety, Reliability, and Security. XI, 405 pages. 2005.

Vol. 3687: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Image Analysis, Part II. XXV, 809 pages. 2005.

Vol. 3686: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Part I. XXVI, 689 pages. 2005.

Vol. 3685: V. Gorodetsky, I. Kotenko, V. Skormin (Eds.), Computer Network Security. XIV, 480 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005. (Subseries LNAI).

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005. (Subseries LNAI).

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005. (Subseries LNAI).

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005. (Subseries LNAI).

Vol. 3680: C. Priami, A. Zelikovsky (Eds.), Transactions on Computational Systems Biology II. IX, 153 pages. 2005. (Subseries LNBI).

Vol. 3679: S.d.C. di Vimercati, P. Syverson, D. Gollmann (Eds.), Computer Security – ESORICS 2005. XI, 509 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005. (Subseries LNBI).

Table of Contents

Invited Talks

Specification

Modelling

Security

Communication

Development

Testing

Verification

Tools

Realising the Benefits of Formal Methods

Anthony Hall

22 Hayward Road, Oxford OX2 8LW, UK
anthony@anthonyhall.org

I keep six honest serving-men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.

Rudyard Kipling

Abstract. The web site for this conference states that: "The challenge now is to achieve general acceptance of formal methods as a part of industrial development of high quality systems, particularly trusted systems." We are all going to be discussing How to achieve this, but before that we should maybe ask the other questions: What are the real benefits of formal methods and Why should we care about them? When and Where should we expect to use them, and Who should be involved? I will suggest some answers to those questions and then describe some ways that the benefits are being realised in practice, and what I think needs to happen for them to become more widespread.

1 What Have Formal Methods Ever Done for Us?

Formal methods consist of writing formal descriptions, analyzing those descriptions and in some cases producing new descriptions – for example refinements – from them. In what way is this a useful activity? First, experience shows that the very act of writing the formal description is of benefit: it forces the writer to ask all sorts of questions that would otherwise be postponed until coding. Of course, that's no help if the problem is so simple that one can write the code straight away, but in the vast majority of systems the code is far too big and detailed to be a useful description of the system for any human purpose. A formal specification, on the other hand, is a description that is abstract, precise and in some senses complete. The abstraction allows a human reader to understand the big picture; the precision forces ambiguities to be questioned and removed; and the completeness means that all aspects of behaviour - for example error cases - are described and understood.

Second, the formality of the description allows us to carry out rigorous analysis. By looking at a single description one can determine useful properties such as consistency or deadlock-freedom. By writing different descriptions from different points of view one can determine important properties such as satisfaction of high level requirements or correctness of a proposed design.

K.-K. Lau and R. Banach (Eds.): ICFEM 2005, LNCS 3785, pp. 1–4, 2005.

There are, however, stronger claims sometimes made for formal methods that are not, in my opinion, justified. The whole notion of proof as qualitatively superior to other analysis methods seems to me wrong: proof is no more a guarantee of correctness than testing, and in many cases far less of one. Furthermore, formal methods are descriptive and analytic: they are not creative. There is no such thing as a formal design process, only formal ways of describing and analyzing designs. So we must combine formal methods with other approaches if we actually want to build a real system.

2 Why Bother?

There sometimes seems to be a belief that formal methods are somehow morally better than other approaches to software development, and that they can lead to the holy grail of zero defect software. This is nonsense, and the fact that it's so obviously untrue is part of the reason for the strong backlash against formal methods. What is true, however, is that formal methods contribute to demonstrably cost-effective development of software with very low defect rates. It is economically perverse to try to develop such software without using them.

Furthermore, formal methods provide, for free, the kind of evidence that is needed in heavily regulated industries such as aviation. They demonstrate responsible engineering and give solid reasons for trust in the product. As more and more industries demand such trust, formal methods become increasingly attractive.

In trying to realise the benefits, therefore, we should be looking at cost-effective methods that address the major risks and that provide tangible evidence of trustworthiness. That is not the same as looking for perfection or proving every single piece of code.

3 When Do Formal Methods Bring Benefit?

It is well known that the early activities in the lifecycle are the most important. It follows that the most effective use of formal methods is at these early stages: requirements analysis, specification, high-level design. In contrast, a lot of research in formal methods has concentrated on low-level design and programming. The early use of formal methods does pose challenges: we need better notations and tools to address large scale specification issues.

As well as concentrating on the early lifecycle, formal methods need to be used from the start of each activity, not as a check at the end. We should concentrate, I believe, on correct construction rather than post-hoc analysis. Lots of experience with analysis tools tells us that it is far easier and more effective to demonstrate the correctness of a well constructed program than to analyse a poorly constructed one to find the numerous flaws that it contains. However, there is a real human problem in persuading people to think carefully rather than adopting the classic hack and test approach to programming.

4 Where Are They Best Used?

Formal methods traditionally live in a ghetto where they are applied to critical parts of critical systems. While I don't believe that they will ever be widely applied to fast-moving software such as web pages where the occasional failure is tolerated or even expected, there is an increasing amount of software where failure is becoming unacceptable and costly, and we need to extend the reach of formal methods to a wide range of systems such as banks, cars, telecommunications and domestic appliances.

Even where they have been used, formal methods have often been seen as a specialist activity divorced from the main development. Although there are some successful projects that have followed this approach, it is not a viable approach for most organisations. I believe strongly that formal methods will only be accepted when they clearly add value to mainstream development and verification activities.

5 Who Uses Formal Methods?

It follows from the previous point that everyone on a project needs to come into contact with formal methods. This is clearly a challenge: current formal notations are notoriously opaque, and formal methods tools are almost all hard to use. We need two things to happen.

First, there needs to be a change in attitude among developers, to accept that like engineers in any other discipline they need to use relevant mathematics as a daily part of their job.

Second, we need to make the mathematics more relevant and palatable, and integrate it better into the other less frightening notations that people are used to.

6 How Can We Realise the Benefits?

The previous sections have provided a pretty challenging list of issues for formal methods practitioners and researchers. In this section I will describe one process, Correctness by Construction (CbyC), which I believe starts to address these issues. CbyC aims to be a Lean Development process: there are, remarkably, commonalities between its philosophy and that of many so-called Agile processes. In particular it is strongly risk-driven, demands that all activities add value to the final product, and is based on tight feedback at every stage. Its big difference is the use of the most rigorous practical notation for each artifact, giving the maximum opportunity for analysis.

CbyC has some successful projects under its belt, but there is a long way to go before it or anything like it is a mainstream process. I will conclude by looking at the challenges for formal methods researchers and tool developers if they are going to support a practical process on a large scale in industry. Here are some examples of questions that need to be answered: