Editorial Board

David Hutchison
  Lancaster University, UK
Takeo Kanade
  Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
  University of Surrey, Guildford, UK
Jon M. Kleinberg
  Cornell University, Ithaca, NY, USA
Friedemann Mattern
  ETH Zurich, Switzerland
John C. Mitchell
  Stanford University, CA, USA
Moni Naor
  Weizmann Institute of Science, Rehovot, Israel
Oscar Nierstrasz
  University of Bern, Switzerland
C. Pandu Rangan
  Indian Institute of Technology, Madras, India
Bernhard Steffen
  University of Dortmund, Germany
Madhu Sudan
  Massachusetts Institute of Technology, MA, USA
Demetri Terzopoulos
  University of California, Los Angeles, CA, USA
Doug Tygar
  University of California, Berkeley, CA, USA
Moshe Y. Vardi
  Rice University, Houston, TX, USA
Gerhard Weikum
  Max-Planck Institute of Computer Science, Saarbruecken, Germany
Preface

Large and complex software systems provide the necessary infrastructure in all industries today. In order to construct such large systems in a systematic manner, the focus in the development methodologies has switched in the last two decades from functional issues to structural issues: both data and functions are encapsulated into software units which are integrated into large systems by means of various techniques supporting reusability and modifiability. This encapsulation principle is essential to both the object-oriented and the more recent component-based software engineering paradigms.

Formal methods have been applied successfully to the verification of medium-sized programs in protocol and hardware design. However, their application to the development of large systems requires more emphasis on specification, modeling and validation techniques supporting the concepts of reusability and modifiability, and their implementation in new extensions of existing programming languages like Java.

The fifth international symposium on Formal Methods for Components and Objects (FMCO 2006) was held in Amsterdam, The Netherlands, June 7–11, 2007. The program consisted of invited keynote lectures and tutorial lectures selected through a corresponding open-call. The latter provide a tutorial perspective on recent developments. In contrast to many existing conferences, about half of the program consisted of invited keynote lectures by top researchers sharing their interest in the application or development of formal methods for large-scale software systems (object or component oriented). FMCO does not focus on specific aspects of the use of formal methods, but rather it aims at a systematic and comprehensive account of the expanding body of knowledge on modern software systems.

This volume contains the contributions submitted after the symposium by both the invited and selected lecturers. The proceedings of FMCO 2002, FMCO 2003, FMCO 2004 and FMCO 2005 have already been published as volumes 2852, 3188, 3657, and 4111 of Springer’s Lecture Notes in Computer Science. We believe that these proceedings provide a unique combination of ideas on software engineering and formal methods which reflect the expanding body of knowledge on modern software systems.

Finally, we thank all authors for the high quality of their contributions, and the reviewers for their help in improving the papers for this volume.

June 2007

Frank de Boer
Marcello Bonsangue
Susanne Graf
Willem-Paul de Roever
Organization

The FMCO symposia are organized in the context of the project Mobi-J, a project founded by a bilateral research program of The Dutch Organization for Scientific Research (NWO) and the Central Public Funding Organization for Academic Research in Germany (DFG). The partners of the Mobi-J projects are: the Centrum voor Wiskunde en Informatica, the Leiden Institute of Advanced Computer Science, and the Christian-Albrechts-Universität Kiel.

This project aims at the development of a programming environment which supports component-based design and verification of Java programs annotated with assertions. The overall approach is based on an extension of the Java language with a notion of component that provides for the encapsulation of its internal processing of data and composition in a network by means of mobile asynchronous channels.

Sponsoring Institutions

The Dutch Organization for Scientific Research (NWO)
The Royal Netherlands Academy of Arts and Sciences (KNAW)
The Dutch Institute for Programming research and Algorithmics (IPA)
The Centrum voor Wiskunde en Informatica (CWI), The Netherlands
The Leiden Institute of Advanced Computer Science (LIACS), The Netherlands
# Table of Contents

## Testing

Model-Based Testing of Environmental Conformance of Components . . . 1  
* Lars Frantzen and Jan Tretmans

Exhaustive Testing of Exception Handlers with Enforcer 26  
* Cyrille Artho, Armin Biere, and Shinichi Honiden

Model-Based Test Selection for Infinite-State Reactive Systems . . . . 47  
* Bertrand Jeannet, Thierry Jéron, and Vlad Rusu

## Program Verification

Verifying Object-Oriented Programs with KeY: A Tutorial . . . . . . 70  
* Wolfgang Ahrendt, Bernhard Beckert, Reiner Hähnle,  
  Philipp Rümmer, and Peter H. Schmitt

Rebeca: Theory, Applications, and Tools ................................ 102  
* Marjan Sirjani

Learning Meets Verification ............................................... 127  
* Martin Leucker

## Trust and Security

JACK—A Tool for Validation of Security and Behaviour of Java Applications ................................................................. 152  
* Gilles Barthe, Lilian Burdy, Julien Charles, Benjamin Grégoire,  
  Marieke Huisman, Jean-Louis Lanet, Mariela Pavlova, and  
  Antoine Requet

Towards a Formal Framework for Computational Trust ............... 175  
* Vladimiro Sassone, Karl Krukow, and Mogens Nielsen

## Models of Computation

On Recursion, Replication and Scope Mechanisms in Process Calculi . . . 185  
* Jesús Aranda, Cinzia Di Giusto, Catuscia Palamidessi, and  
  Frank D. Valencia

Bounded Session Types for Object Oriented Languages ............... 207  
* Mariangiola Dezani-Ciancaglini, Elena Giachino,  
  Sophia Drossopoulou, and Nobuko Yoshida
Distributed Programming

Reflecting on Aspect-Oriented Programming, Metaprogramming, and Adaptive Distributed Monitoring ........................................... 246

*Bill Donkervoet and Gul Agha*

Links: Web Programming Without Tiers ........................................... 266

*Ezra Cooper, Sam Lindley, Philip Wadler, and Jeremy Yallop*

Author Index .................................................................................. 297