

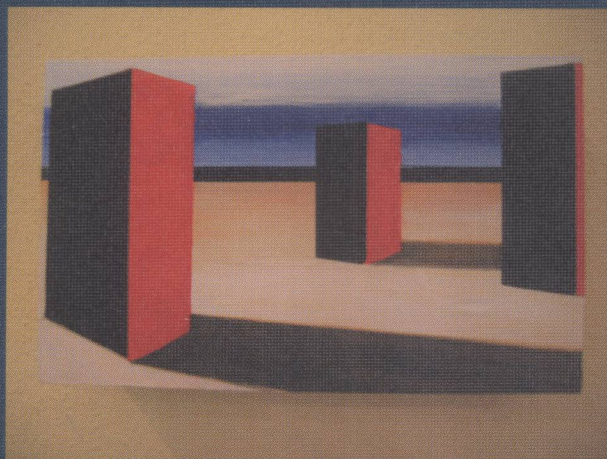
Festschrift

LNCS 4444

Thomas Reps  
Mooly Sagiv  
Jörg Bauer (Eds.)

# Program Analysis and Compilation, Theory and Practice

Essays Dedicated to Reinhard Wilhelm  
on the Occasion of His 60th Birthday



Springer

TP311-53  
P964.5  
2006

Thomas Reps Mooly Sagiv  
Jörg Bauer (Eds.)

# Program Analysis and Compilation, Theory and Practice

Essays Dedicated to Reinhard Wilhelm  
on the Occasion of His 60th Birthday



Springer



E2007003199

## Volume Editors

Thomas Reps  
University of Wisconsin-Madison  
Computer Sciences Department  
1210 West Dayton Street, Madison, WI 53706-1685, USA  
E-mail: reps@cs.wisc.edu

Mooly Sagiv  
Tel-Aviv University  
School of Computer Science, Schreiber 317  
Tel-Aviv, 69978, Israel  
E-mail: msagiv@post.tau.ac.il

Jörg Bauer  
Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 322, 2800 Kongens Lyngby, Denmark  
E-mail: joba@imm.dtu.dk

The illustration appearing on the cover of the book is the work of Sabrina Sperl, Saarbrücken.

The piece of art is in the possession of the International Conference and Research Centre for Informatics in Schloss Dagstuhl.

Library of Congress Control Number: 2007922986

CR Subject Classification (1998): D.3, F.3.1-2, D.2.8, F.4.2, D.1

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743  
ISBN-10 3-540-71315-8 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-71315-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media  
springer.com

© Springer-Verlag Berlin Heidelberg 2007  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 12032570 06/3142 5 4 3 2 1 0

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## 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

On June 9–10, 2006, a celebratory symposium in honor of Reinhard Wilhelm’s 60th birthday was held at Schloß Dagstuhl. Many of his graduate students and research collaborators, as well as current and former colleagues, gathered for this celebration, and Schloß Dagstuhl was certainly the most appropriate location to honor its founder. During this day and a half of scientific discussions, there were 15 research presentations, as well as a group discussion on the topic “The Future of Static Analysis.” The event consisted of the following talks:

1. *Christian Ferdinand*:  
“New Developments in WCET Analysis”
2. *Wolfgang J. Paul*:  
“Verification of Distributed Systems with Realistic Time Bounds”
3. *Dieter Maurer*:  
“Lazy Execution of Boolean Queries”
4. *Raimund Seidel*:  
“Ideally Distributed Dynamic Random Search Trees Without Using Extra Space”
5. *Andreas Zeller*:  
“Static and Dynamic Inference of Temporal Properties”
6. *Helmut Seidl*:  
“Verification of Cryptographic Protocols by Means of Tractable Classes of Horn Clauses”
7. *Neil Jones*:  
“The Flow of Data and the Complexity of Algorithms”
8. *Jaan Penjam*:  
“Deductive and Inductive Methods for Program Synthesis”
9. *Arnd Poetsch-Heffter*:  
“Encapsulation Properties of OO-Programs Require New Heap Analyses”
10. *Bernhard Steffen*:  
“Service-Oriented Compiler Construction”
11. *Chris Hankin*:  
“Average Case Execution Times Matter Too”
12. *Robert Giegerich*:  
“Making Dynamic Programming Fun”
13. *Patrick Cousot*:  
“Grammar Abstract Interpretation”
14. *Yosi Ben-Asher*:  
“Source Level Merging of Programs for Embedded Systems”
15. *Mila Majster-Cederbaum*:  
“Ensuring Properties of Interaction Systems by Construction”

The articles of this volume serve roughly as a proceedings of this Dagstuhl symposium, although some of the contributors could not participate in the celebration, and

not all of the participants managed to contribute an article. However, we are pleased to have been able to assemble this stimulating collection of research articles, which honor both Reinhard's 60th birthday and more broadly his contributions to computer science (some of which are summarized in the lead-off article, "An Appreciation of the Work of Reinhard Wilhelm").

The Dagstuhl symposium could not have been the success that it turned out to be without the help of Angelika Müller, Rosi Bardohl, Annette Beyer, and all of the Dagstuhl staff.

Following lunch on the second day, the participants traveled to Saarbrücken to attend a less scientific (and more alcoholic) 60th-birthday celebration, as is traditional for a German academic, given by the Computer Science Department at Saarland University. There, Reinhard endured another series of talks offered by his graduate students, his Dagstuhl colleagues, his colleagues from the department, his brother, and several close friends. Reinhard's closing speech in response provided insight into some of the events and influences—both personal and intellectual—that have shaped his career.

We would like to thank the local organizers of the afternoon/evening celebration, and in particular, Erich Reindel from the Computer Science Department at Saarland University and Volker Geiß from the Max Planck Institute for Computer Science.

We offer this volume to Reinhard as a token of the honor that he so richly deserves. It is fitting that the ideas in the articles found herein span such a broad range of Reinhard's research interests, from ones from very nearly the beginning of his career to current and—we hope!—future work to come.

December 2006

Thomas Reps  
Mooly Sagiv  
Jörg Bauer

# Lecture Notes in Computer Science

For information about Vols. 1–4322

please contact your bookseller or Springer

- Vol. 4444: T. Reps, M. Sagiv, J. Bauer (Eds.), *Program Analysis and Compilation, Theory and Practice*. X, 361 pages. 2007.
- Vol. 4429: R. Lu, J.H. Siekmann, C. Ullrich (Eds.), *Cognitive Systems*. X, 161 pages. 2007. (Sublibrary LNAI).
- Vol. 4424: O. Grumberg, M. Huth (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems*. XX, 738 pages. 2007.
- Vol. 4423: H. Seidl (Ed.), *Foundations of Software Science and Computational Structures*. XVI, 379 pages. 2007.
- Vol. 4422: M.B. Dwyer, A. Lopes (Eds.), *Fundamental Approaches to Software Engineering*. XV, 440 pages. 2007.
- Vol. 4421: R. De Nicola (Ed.), *Programming Languages and Systems*. XVII, 538 pages. 2007.
- Vol. 4420: S. Krishnamurthi, M. Odersky (Eds.), *Compiler Construction*. XIV, 233 pages. 2007.
- Vol. 4419: P.C. Diniz, E. Marques, K. Bertels, M.M. Fernandes, J.M.P. Cardoso (Eds.), *Reconfigurable Computing: Architectures, Tools and Applications*. XIV, 391 pages. 2007.
- Vol. 4415: P. Lukowicz, L. Thiele, G. Tröster (Eds.), *Architecture of Computing Systems - ARCS 2007*. X, 297 pages. 2007.
- Vol. 4414: S. Hochreiter, R. Wagner (Eds.), *Bioinformatics Research and Development*. XVI, 482 pages. 2007. (Sublibrary LNBI).
- Vol. 4410: A. Branco (Ed.), *Anaphora: Analysis, Algorithms and Applications*. X, 191 pages. 2007. (Sublibrary LNAI).
- Vol. 4407: G. Puebla (Ed.), *Logic-Based Program Synthesis and Transformation*. VIII, 237 pages. 2007.
- Vol. 4405: L. Padgham, F. Zambonelli (Eds.), *Agent-Oriented Software Engineering*. VII. XII, 225 pages. 2007.
- Vol. 4403: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), *Evolutionary Multi-Criterion Optimization*. XIX, 954 pages. 2007.
- Vol. 4399: X. Llorà, T. Kovacs, K. Takadama, P.L. Lanzi, S.W. Wilson, W. Stolzmann (Eds.), *Learning Classifier Systems*. XII, 345 pages. 2007. (Sublibrary LNAI).
- Vol. 4397: C. Stephanidis, M. Pieper (Eds.), *Universal Access in Ambient Intelligence Environments*. XV, 467 pages. 2007.
- Vol. 4396: J. García-Vidal, L. Cerdà-Alabern (Eds.), *Wireless Systems and Mobility in Next Generation Internet*. IX, 271 pages. 2007.
- Vol. 4395: M. Daydé, J.M.L.M. Palma, A.L.G.A. Coutinho, E. Pacitti, J.C. Lopes (Eds.), *High Performance Computing for Computational Science - VEC- PAR 2006*. XXIV, 721 pages. 2007.
- Vol. 4394: A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing*. XVI, 648 pages. 2007.
- Vol. 4393: W. Thomas, P. Weil (Eds.), *STACS 2007*. XVIII, 708 pages. 2007.
- Vol. 4392: S.P. Vadhan (Ed.), *Theory of Cryptography*. XI, 595 pages. 2007.
- Vol. 4390: S.O. Kuznetsov, S. Schmidt (Eds.), *Formal Concept Analysis*. X, 329 pages. 2007. (Sublibrary LNAI).
- Vol. 4389: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), *Environments for Multi-Agent Systems III*. X, 273 pages. 2007. (Sublibrary LNAI).
- Vol. 4385: K. Coninx, K. Luyten, K.A. Schneider (Eds.), *Task Models and Diagrams for Users Interface Design*. XI, 355 pages. 2007.
- Vol. 4384: T. Washio, K. Satoh, H. Takeda, A. Inokuchi (Eds.), *New Frontiers in Artificial Intelligence*. IX, 401 pages. 2007. (Sublibrary LNAI).
- Vol. 4383: E. Bin, A. Ziv, S. Ur (Eds.), *Hardware and Software, Verification and Testing*. XII, 235 pages. 2007.
- Vol. 4381: J. Akiyama, W.Y.C. Chen, M. Kano, X. Li, Q. Yu (Eds.), *Discrete Geometry, Combinatorics and Graph Theory*. XI, 289 pages. 2007.
- Vol. 4380: S. Spaccapietra, P. Atzeni, F. Fages, M.-S. Hacid, M. Kifer, J. Mylopoulos, B. Pernici, P. Shvaiko, J. Trujillo, I. Zaihrayev (Eds.), *Journal on Data Semantics*. VIII. XV, 219 pages. 2007.
- Vol. 4378: I. Virbitskaite, A. Voronkov (Eds.), *Perspectives of Systems Informatics*. XIV, 496 pages. 2007.
- Vol. 4377: M. Abe (Ed.), *Topics in Cryptology – CT-RSA 2007*. XI, 403 pages. 2006.
- Vol. 4376: E. Frachtenberg, U. Schwiegelshohn (Eds.), *Job Scheduling Strategies for Parallel Processing*. VII, 257 pages. 2007.
- Vol. 4374: J.F. Peters, A. Skowron, I. Düntsch, J. Grzymała-Busse, E. Orłowska, L. Polkowski (Eds.), *Transactions on Rough Sets VI, Part I*. XII, 499 pages. 2007.
- Vol. 4373: K. Langendoen, T. Voigt (Eds.), *Wireless Sensor Networks*. XIII, 358 pages. 2007.
- Vol. 4372: M. Kaufmann, D. Wagner (Eds.), *Graph Drawing*. XIV, 454 pages. 2007.
- Vol. 4371: K. Inoue, K. Satoh, F. Toni (Eds.), *Computational Logic in Multi-Agent Systems*. X, 315 pages. 2007. (Sublibrary LNAI).

- Vol. 4370: P.P. Lévy, B. Le Grand, F. Poulet, M. Soto, L. Darago, L. Toubiana, J.-F. Vibert (Eds.), Pixelization Paradigm. XV, 279 pages. 2007.
- Vol. 4369: M. Umeda, A. Wolf, O. Bartenstein, U. Geske, D. Seipel, O. Takata (Eds.), Declarative Programming for Knowledge Management. X, 229 pages. 2006. (Sublibrary LNAI).
- Vol. 4368: T. Erlebach, C. Kaklamanis (Eds.), Approximation and Online Algorithms. X, 345 pages. 2007.
- Vol. 4367: K. De Bosschere, D. Kaeli, P. Stenström, D. Whalley, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XI, 307 pages. 2007.
- Vol. 4366: K. Tuyls, R. Westra, Y. Saecys, A. Nowé (Eds.), Knowledge Discovery and Emergent Complexity in Bioinformatics. IX, 183 pages. 2007. (Sublibrary LNBI).
- Vol. 4364: T. Kühne (Ed.), Models in Software Engineering. XI, 332 pages. 2007.
- Vol. 4362: J. van Leeuwen, G.F. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Pláčil (Eds.), SOFSEM 2007: Theory and Practice of Computer Science. XXI, 937 pages. 2007.
- Vol. 4361: H.J. Hoogeboom, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 555 pages. 2006.
- Vol. 4360: W. Dubitzky, A. Schuster, P.M.A. Sloat, M. Schroeder, M. Romberg (Eds.), Distributed, High-Performance and Grid Computing in Computational Biology. X, 192 pages. 2007. (Sublibrary LNBI).
- Vol. 4358: R. Vidal, A. Heyden, Y. Ma (Eds.), Dynamical Vision. IX, 329 pages. 2007.
- Vol. 4357: L. Buttyán, V. Gligor, D. Westhoff (Eds.), Security and Privacy in Ad-Hoc and Sensor Networks. X, 193 pages. 2006.
- Vol. 4355: J. Julliand, O. Kouchnarenko (Eds.), B 2007: Formal Specification and Development in B. XIII, 293 pages. 2006.
- Vol. 4354: M. Hanus (Ed.), Practical Aspects of Declarative Languages. X, 335 pages. 2006.
- Vol. 4353: T. Schwentick, D. Suciu (Eds.), Database Theory – ICDT 2007. XI, 419 pages. 2006.
- Vol. 4352: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part II. XVIII, 743 pages. 2006.
- Vol. 4351: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part I. XIX, 797 pages. 2006.
- Vol. 4349: B. Cook, A. Podelski (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 395 pages. 2007.
- Vol. 4348: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder, P. Leroy (Eds.), Ada 2005 Reference Manual. XXII, 765 pages. 2006.
- Vol. 4347: J. Lopez (Ed.), Critical Information Infrastructures Security. X, 286 pages. 2006.
- Vol. 4346: L. Brim, B. Haverkort, M. Leucker, J. van de Pol (Eds.), Formal Methods: Applications and Technology. X, 363 pages. 2007.
- Vol. 4345: N. Maglaveras, I. Chouvarda, V. Koutkias, R. Brause (Eds.), Biological and Medical Data Analysis. XIII, 496 pages. 2006. (Sublibrary LNBI).
- Vol. 4344: V. Gruhn, F. Oquendo (Eds.), Software Architecture. X, 245 pages. 2006.
- Vol. 4342: H. de Swart, E. Orłowska, G. Schmidt, M. Roubens (Eds.), Theory and Applications of Relational Structures as Knowledge Instruments II. X, 373 pages. 2006. (Sublibrary LNAI).
- Vol. 4341: P.Q. Nguyen (Ed.), Progress in Cryptology - VIETCRYPT 2006. XI, 385 pages. 2006.
- Vol. 4340: R. Prodan, T. Fahringer, Grid Computing. XXIII, 317 pages. 2007.
- Vol. 4339: E. Ayguadé, G. Baumgartner, J. Ramanujam, P. Sadayappan (Eds.), Languages and Compilers for Parallel Computing. XI, 476 pages. 2006.
- Vol. 4338: P. Kalra, S. Peleg (Eds.), Computer Vision, Graphics and Image Processing. XV, 965 pages. 2006.
- Vol. 4337: S. Arun-Kumar, N. Garg (Eds.), FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science. XIII, 430 pages. 2006.
- Vol. 4336: V.R. Basili, D. Rombach, K. Schneider, B. Kitchenham, D. Pfahl, R.W. Selby, Empirical Software Engineering Issues. XVII, 193 pages. 2007.
- Vol. 4335: S.A. Brueckner, S. Hassas, M. Jelasity, D. Yamins (Eds.), Engineering Self-Organising Systems. XII, 212 pages. 2007. (Sublibrary LNAI).
- Vol. 4334: B. Beckert, R. Hähnle, P.H. Schmitt (Eds.), Verification of Object-Oriented Software. XXIX, 658 pages. 2007. (Sublibrary LNAI).
- Vol. 4333: U. Reimer, D. Karagiannis (Eds.), Practical Aspects of Knowledge Management. XII, 338 pages. 2006. (Sublibrary LNAI).
- Vol. 4332: A. Bagchi, V. Atluri (Eds.), Information Systems Security. XV, 382 pages. 2006.
- Vol. 4331: G. Min, B. Di Martino, L.T. Yang, M. Guo, G. Ruenger (Eds.), Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops. XXXVII, 1141 pages. 2006.
- Vol. 4330: M. Guo, L.T. Yang, B. Di Martino, H.P. Zima, J. Dongarra, F. Tang (Eds.), Parallel and Distributed Processing and Applications. XVIII, 953 pages. 2006.
- Vol. 4329: R. Barua, T. Lange (Eds.), Progress in Cryptology - INDOCRYPT 2006. X, 454 pages. 2006.
- Vol. 4328: D. Penkler, M. Reitenspiess, F. Tam (Eds.), Service Availability. X, 289 pages. 2006.
- Vol. 4327: M. Baldoni, U. Endriss (Eds.), Declarative Agent Languages and Technologies IV. VIII, 257 pages. 2006. (Sublibrary LNAI).
- Vol. 4326: S. Göbel, R. Malkewitz, I. Iurgel (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. X, 384 pages. 2006.
- Vol. 4325: J. Cao, I. Stojmenovic, X. Jia, S.K. Das (Eds.), Mobile Ad-hoc and Sensor Networks. XIX, 887 pages. 2006.
- Vol. 4323: G. Doherty, A. Blandford (Eds.), Interactive Systems. XI, 269 pages. 2007.

¥562.00元



# Table of Contents

## An Appreciation

An Appreciation of the Work of Reinhard Wilhelm . . . . .	1
<i>Thomas Reps, Mooly Sagiv, and Jörg Bauer</i>	

## Technical Papers

New Developments in WCET Analysis . . . . .	12
<i>Christian Ferdinand, Florian Martin, Christoph Cullmann, Marc Schlickling, Ingmar Stein, Stephan Thesing, and Reinhold Heckmann</i>	
Realistic Worst-Case Execution Time Analysis in the Context of Pervasive System Verification . . . . .	53
<i>Steffen Knapp and Wolfgang Paul</i>	
Lazy Execution of Boolean Queries . . . . .	82
<i>Dieter Maurer</i>	
Cryptographic Protocol Verification Using Tractable Classes of Horn Clauses . . . . .	97
<i>Helmut Seidl and Kumar Neeraj Verma</i>	
Inferring Ownership Types for Encapsulated Object-Oriented Program Components . . . . .	120
<i>Arnd Poetzsch-Heffter, Kathrin Geilmann, and Jan Schäfer</i>	
ViDoC- Visual Design of Optimizing Compilers . . . . .	145
<i>Tiziana Margaria, Oliver Rüthing, and Bernhard Steffen</i>	
Abstract Interpretation for Worst and Average Case Analysis . . . . .	160
<i>Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky</i>	
Grammar Analysis and Parsing by Abstract Interpretation . . . . .	175
<i>Patrick Cousot and Radhia Cousot</i>	
Ensuring Properties of Interaction Systems . . . . .	201
<i>Gregor Gössler, Susanne Graf, Mila Majster-Cederbaum, Moritz Martens, and Joseph Sifakis</i>	
On the Expressive Power of Live Sequence Charts . . . . .	225
<i>Werner Damm, Tobe Toben, and Bernd Westphal</i>	

Refinement-Based Verification for Possibly-Cyclic Lists .....	247
<i>Alexey Loginov, Thomas Reps, and Mooly Sagiv</i>	
Abstract Counterexample-Based Refinement for Powerset Domains .....	273
<i>Roman Manevich, John Field, Thomas A. Henzinger, G. Ramalingam, and Mooly Sagiv</i>	
Types from Control Flow Analysis .....	293
<i>Flemming Nielson and Hanne Riis Nielson</i>	
Data Flow Analysis for CCS .....	311
<i>Hanne Riis Nielson and Flemming Nielson</i>	
Towards a Source Level Compiler: Source Level Modulo Scheduling .....	328
<i>Yosi Ben-Asher and Danny Meisler</i>	
<b>Author Index</b> .....	<b>361</b>

# An Appreciation of the Work of Reinhard Wilhelm

Thomas Reps<sup>1</sup>, Mooly Sagiv<sup>2</sup>, and Jörg Bauer<sup>3</sup>

<sup>1</sup> Comp. Sci. Dept., University of Wisconsin  
reps@cs.wisc.edu

<sup>2</sup> School of Comp. Sci., Tel-Aviv University  
msagiv@post.tau.ac.il

<sup>3</sup> Fachrichtung Informatik, Univ. des Saarlandes  
joba@cs.uni-sb.de

Reinhard Wilhelm's career in Computer Science spans more than a third of a century. During this time, he has made numerous research contributions in the areas of programming languages, compilers and compiler generators, static program analysis, program transformation, algorithm animation, and real-time systems; co-founded a company to transfer some of these ideas to industry; held the Chair for Programming Languages and Compiler Construction at Saarland University; and served since its inception as the Scientific Director of the International Conference and Research Center for Computer Science at Schloß Dagstuhl.

## 1 Research Activities

### 1.1 Foundations of Programming and Programming Languages

Reinhard's work in the area of programming languages is unusual in that he has had an interest in, and made contributions in, *all* styles of programming languages (imperative, functional, logic, parallel, object-oriented, and constraint-oriented) [14, 15, 16, 17, 20, 19, 21, 22, 24, 25, 26, 31, 77].

### 1.2 Compilers, Compiler Generators, and Compilation Algorithms

*Compilers* convert a program or specification written in some language into a form that allows it to be executed on a computer or network of computers. *Compiler generators* are tools for creating compilers themselves (or components of compilers) from specifications. From the number of his publications on these subjects, and their distribution in time, it is easy to see that Reinhard's longest-held interest and deepest attachment in Computer Science has been to the area of compilers [32], including compiler generators [27, 28, 29, 30] and the algorithms needed to accomplish the tasks required in various phases of compilation (see below).

**Attribute Grammars.** Attribute grammars are a language-description formalism that was introduced by Donald Knuth in 1968.<sup>1</sup> In an attribute grammar, a language and its properties are specified by giving

<sup>1</sup> D.E. Knuth: Semantics of context-free languages. Math. Syst. Theory 2(2): 127-145 (1968).

- a context-free grammar for the language,
- sets of “attributes” (annotations) to be attached to nodes of the grammar’s derivation trees, and
- rules that describe how information from one node’s attributes affects the attributes of other nodes.

Reinhard’s research in this area concerned analysis and applications of attribute grammars [33, 34, 35, 36, 37, 38, 39, 40, 41, 60, 61]. Reinhard also spearheaded the implementation of three of the most influential compiler-generation systems that were based on attribute grammars: MUG1 [28], MUG2 [29, 30], and OPTRAN [38, 77, 40, 41, 72].

**Code Generation via Transformational Grammars/Tree Automata.** The code-generation phase of a compiler turns an intermediate representation of a program into a sequence of machine-code instructions. One code-generation subtask is code selection—the actual selection of the instructions to be emitted. In a sequence of papers [44, 46, 48], Reinhard together with Helmut Seidl developed the connections between the code-selection problem and the theories of regular-tree grammars and finite tree automata. This work provided a sound theoretical basis for code selection, generalized and improved existing methods, and furthered the understanding of how to generate code-selection components of compilers from machine descriptions.

**Other Compilation Algorithms.** Other compilation issues with which Reinhard has been concerned include table compression [47], graph reduction [42], code optimization [43, 49], and virtual machines [50, 51].

### 1.3 Static Program Analysis

One of the areas in which Reinhard has worked for multiple decades, and made many important contributions, is *static program analysis* (also known as “dataflow analysis” or “abstract interpretation”). Static analysis is the basic technique used in optimizing compilers and other programming tools for obtaining information about the possible states that a program can reach during execution, but without actually running the program on specific inputs. Instead, static-analysis techniques explore a program’s behavior for *all* possible inputs to determine (an approximation of) *all* possible states that the program can reach. To make this feasible, the program is “run in the aggregate”—i.e., on descriptors that represent collections of many states. The fundamental theory that provides the foundation for such techniques is that of *abstract interpretation*, enunciated in 1977 by Patrick and Radhia Cousot.<sup>2</sup>

Reinhard’s work in the area of static analysis includes [52, 53, 54, 55, 56, 57, 58, 61, 59], as well as many other papers mentioned below.

**Grammar Flow Analysis.** In two papers with U. Möncke [60, 61], Reinhard proposed the technique of “grammar flow analysis”. This work generalized the concepts and

---

<sup>2</sup> P. Cousot, R. Cousot: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. POPL 1977: 238-252.



algorithms of (intraprocedural) dataflow analysis—which generally is applied to graph data structures—to a method for obtaining information about properties of derivation trees of the nonterminals in context-free grammars (but without actually building the grammar’s derivation trees). Not only does this generalization have important applications in interprocedural dataflow analysis, but Möncke and Wilhelm also showed how grammar flow analysis has applications in many other areas of compilers—ranging from static-semantic analysis to code generation.

**Shape Analysis.** In a series of papers with M. Sagiv, T. Reps, and others, (e.g., [62, 63, 64, 65, 66, 67, 68, 69, 70, 71]), Reinhard has addressed one of the major remaining challenges in static analysis of program behavior—sometimes called *shape analysis*—namely, how to check properties of programs that manipulate linked data structures (i.e., programs that use dynamic allocation and freeing of storage cells, and destructive updating of structure fields—as happens pervasively in languages such as C, C++, and Java). In such programs, data structures can grow and shrink dynamically, with no fixed upper bound on their size or number. The analysis problem is complicated even more by the fact that such languages also permit fields of dynamically allocated objects to be destructively updated. In the case of thread-based languages, such as Java, the number of threads can also grow and shrink dynamically—again with no fixed upper bound on size or number. These features create considerable difficulties for any method that tries to check program properties, and this subject is considered to be one of the most challenging areas of program analysis.

A key issue when analyzing programs that use such features is how to create finite-sized descriptors of memory configurations, such that the descriptors

- abstract away certain details, but
- retain enough key information so that the analysis can identify interesting properties that hold.

One of the crucial obstacles is that dynamically-allocated storage cells have no static names, and the number of such objects is, in general, unbounded.

The Sagiv-Reps-Wilhelm approach [67] provides a solution to the problem of devising finite-sized descriptors of the infinite number of storage configurations that can arise at a point in the program. Moreover, it also provides a way to tune the precision of the descriptors in use, which is important both for reducing the time required to run an analysis, and for reducing the number of false positives that an analysis may report. From a static-analysis perspective, the work is novel because of the way that it uses 3-valued logic to address these issues.

#### 1.4 Program Transformation/Rewriting

The heart of many compilation steps (such as optimization and code generation) is *tree transformation* or *rewriting*. Reinhard’s work in this area has concerned systems

for program transformation and optimization—originally formalized using attribute-grammar-based notations and implemented as rewriting of attributed abstract-syntax trees [72, 73, 74, 75, 77]. Later work addressed these problems using a specialized functional notation for specifying patterns and transformations [78].

### 1.5 Algorithm Animation and Visualization

*Algorithm animation* concerns how to create visual presentations of computations. This area has been another of Reinhard’s long-term interests [79, 80, 81, 82, 83]. He has worked on such varied problems as the depiction of compilation steps [83], finite-state automata [82], and abstract interpretation [79, 80].

### 1.6 Timing Analysis for Real-Time Systems

In a series of papers with various authors (beginning with [84]), Reinhard spearheaded a new approach to predicting the timing behavior of processors that are equipped with cache memory. Because of the substantial differences exhibited by modern processors between the latency for an access that goes to main memory versus an access that can be resolved in the cache, cache behavior has to be taken into account to predict a program’s execution time. Classical methods based on experimental measurement provide only soft guarantees: any measurement-based approach, via either hardware or software, only determines the execution time for the specific inputs of the test suite. Moreover, software-based monitoring alters the code, thereby influencing the cache behavior—and hence the performance—of the program whose performance one is trying to measure.

The behavior-prediction techniques that Reinhard has helped to devise make use of static program analysis, which (as noted above) provides a way to obtain information about the possible states that a program reaches during execution, but without actually running the program on specific inputs. Using this approach, Reinhard and his colleagues have shown that it is possible to determine the worst-case cache behavior (i.e., to identify what could be in the cache at each point in the program) [84, 85, 86, 87, 88, 89], as well as to bound the worst-case running time of real-time programs and to check the program for the possibility of cache conflicts that could affect time-critical parts [90, 91, 92, 93, 94, 95, 96, 97, 98]. The advantage of their approach is that the predictions obtained using their techniques are valid for all inputs. From a static-analysis perspective, the work often combines “may” and “must” information in an unusual way.

An additional area of concern has been with processor models—e.g., with such issues as the semantics of processors and how to specify them—as well as with gaining an understanding of which real-time systems are time-predictable [99, 100, 101, 102, 103].

## 2 Technology Transfer

Reinhard has been successful in making the work on timing analysis for real-time systems accessible to the embedded-systems and real-time communities, thereby

demonstrating how static program analysis can be applied to the problems that arise in these domains. This work also forms the basis for the products of a company that Reinhard co-founded (AbsInt Angewandte Informatik GmbH, Saarbruecken, Germany [104]). Their tool has been successfully used to certify time-critical subsystems of the Airbus A380 plane.

### 3 Pedagogical Activities

Reinhard Wilhelm holds the Chair for Programming Languages and Compiler Construction at Saarland University, where he has supervised twenty-five Ph.D. theses and nearly one hundred fifty Masters/Diploma theses.

Reinhard's textbook ("Compiler Design"), co-authored with D. Maurer and published in German [10], French [11], and English [12], is successfully used in many graduate and undergraduate compiler courses. The book is noteworthy for the way that it presents the problems of compiling imperative languages, functional languages, object-oriented languages, and logic-programming languages in a way that draws out their commonalities. The book covers in depth many difficult aspects of compilation, such as bottom-up parsing with error recovery, attribute grammars, and abstract interpretation. Compared with other textbooks, Reinhard's book provides the most theoretically well-grounded treatment of the problems that arise in writing compilers.

Recently, Reinhard also participated in an effort to design a graduate curriculum on embedded software and systems [13].

### 4 Schloß Dagstuhl

Reinhard has also performed a notable service to the international Computer Science community by having served since its inception as the Scientific Director of the International Conference and Research Center for Computer Science at Schloß Dagstuhl. Dagstuhl was set up in 1990 along the lines of the famous conference center for Mathematics at Oberwolfach. Dagstuhl hosts several kinds of activities, but predominantly "Dagstuhl Seminars", which are week-long intensive seminars involving lectures, group discussions, software demonstrations, etc. By now, Dagstuhl Seminars have been held in a large number of different subject areas of Computer Science. There are about forty of these every year, each with somewhere between twenty and sixty participants from all over the world.

Reinhard has been involved with Dagstuhl from the start, and as the founding Scientific Director, his imprint is to be found on all aspects of its operation, including not just the choice of seminar topics and attention to maintaining the highest scientific standards, but also the original renovation of Schloß Dagstuhl, the design and construction of a major second building, the solicitation of donations from industry and charitable foundations, the arrangement of a special funding program for junior researchers (young faculty and graduate students), and the list could go on and on. Many weeks, he even leads the traditional Wednesday mid-week hike.

Reinhard has made Dagstuhl a very special place for Computer Scientists by his quiet, but skillful, way of using his “bully pulpit” to nudge the Computer Science community into fruitful and interesting interactions:

- He has made seminars that bring together groups that have common interests, but that for one reason or another have had relatively limited contact, a Dagstuhl specialty.
- He attends at least one day of each Dagstuhl seminar to monitor progress and offer advice to that week’s organizers about possible mid-course corrections.
- He actively solicits Seminars on new and promising topics.
- During the planning stages of Seminars, he keeps an eye out for potential attendees who might have been overlooked, but who would especially profit from and/or contribute to the activities of a Seminar.

Dagstuhl and its attendees have also benefited from Reinhard’s careful—and tasteful—attention to detail, which goes far beyond the scientific aspects of the establishment: the choice of music scores and musical instruments in Dagstuhl’s music room; the organizing of displays of modern art in the Dagstuhl buildings and on the Dagstuhl grounds; even the choice of wine in the famous Dagstuhl Wine Cellar, where participants are encouraged to repair for the evening for both technical and non-technical conversation.

## 5 A Partial List of Reinhard Wilhelm’s Collaborators

L. Almeida, M. Alt, H.-J. Bach, M. Baston, J. Bauer, G. Becker, Y. Ben-Asher, A. Benveniste, C. Berg, J. Börstler, P. G. Bouillon, B. Bouyssounouse, B. Braune, F. Warren Burton, G. C. Buttazzo, P. Caspi, J. Ciesinger, V. Claus, I. Crnkovic, W. Damm, S. Diehl, J. Engblom, A. A. Evstiougov-Babaev, C. Fecht, C. Ferdinand, G. Fohler, F. Fontaine, N. Francez, N. Fritz, H. Ganzinger, M. García-Valls, R. Giegerich, I. Glasner, H. Hagen, R. Heckmann, E. Hoffmann, D. Johannes, D. Kästner, A. Kerren, H. Kopetz, B. Kuhn, W. Lahner, Y. Lakhnech, M. Langenbach, F. Laroussinie, L. Lavagno, T. Lev-Ami, G. Lipari, P. Lipps, J. Loeckx, P. Lucas, A. Lucks-Baus, F. Maranchi, F. Martin, D. Maurer, K. Mehlhorn, J. Messerschmidt, U. Möncke, T. Müldner, F. Müller, R. Nollmann, H.-G. Oberhauser, M. Olk, O. Parshin, P. Peti, J. Antonio de la Puente, M. Raber, A. Rakib, F. Randimbivololona, T. Rauber, T. Rimmel, T. Reps, N. Rinetzky, K. Ripken, B. Robinet, M. Rodeh, G. Rünger, M. Sagiv, G. Sander, A. L. Sangiovanni-Vincentelli, N. Scaife, M. Schmidt, J. Schneider, A. Schuster, R. Seidel, H. Seidl, M. Sicks, J. Sifakis, R. de Simone, J. Souyris, O. Spaniol, H. Theiling, S. Thesing, L. Thiele, W. Thome, M. Törngren, P. Veríssimo, B. Weisgerber, A.J. Wellings, D.B. Whalley, S. Wilhelm, T.A.C. Willemse, E. Yahav, W. Yi, G. Yorsh.

## Acknowledgments

We would like to thank Angelika Müller, Rosi Bardohl, Annette Beyer, and all of the Dagstuhl staff for the splendid job they did organizing and hosting the Reinhard Wilhelm 60th-Birthday Celebratory Symposium. We would also like to thank Patrick



Cousot for furnishing us with the list of references to Reinhard's publications that he assembled for the presentation of Reinhard's work at the Saarland University celebration of Reinhard's birthday.

## References

1. R. Wilhelm. *Informatics: 10 Years Back. 10 Years Ahead.* Lecture Notes in Computer Science 2000. Springer, Berlin, Germany, 2001.
2. R. Wilhelm: Compiler Construction, 10th International Conference, CC 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2-6, 2001, Proceedings Springer 2001
3. R. Wilhelm: Informatik: Grundlagen - Anwendungen - Perspektiven [Forum "Perspektiven der Informatik", Dagstuhl, November 1993] Verlag C. H. Beck 1996
4. R. Wilhelm: Generische und generative Methoden. Perspektiven der Informatik 1993: 84-85
5. R. Wilhelm, H. Hagen: Programmiersprachen. Perspektiven der Informatik 1993: 86-90
6. V. Claus, R. Wilhelm: Einleitung. Perspektiven der Informatik 1993: 9-12
7. R. Wilhelm, O. Spaniol: Parallele und verteilte Systeme. Perspektiven der Informatik 1993: 90-94
8. B. Robinet, R. Wilhelm: ESOP 86, European Symposium on Programming, Saarbrücken, Federal Republic of Germany, March 17-19, 1986, Proceedings, LNCS 213, Springer 1986
9. R. Wilhelm: GI - 10. Jahrestagung, Saarbrücken, 30. September - 2. Oktober 1980, Proceedings Springer 1980
10. R. Wilhelm and D. Maurer. *Übersetzerbau - Theorie, Konstruktion, Generierung.* Springer, Berlin, Germany, 1992, 2. Auflage Springer 1997
11. R. Wilhelm and D. Maurer. *Les Compilateurs, théorie, construction, génération.* Masson, Paris, France, 1994.
12. R. Wilhelm and D. Maurer. *Compiler Design: Theory, Construction, Generation.* Addison-Wesley, Reading, MA, 1996.
13. P. Caspi, A. L. Sangiovanni-Vincentelli, Luís Almeida, A. Benveniste, B. Bouyssounouse, G. C. Buttazzo, I. Crnkovic, W. Damm, J. Engblom, G. Fohler, M. García-Valls, H. Kopetz, Y. Lakhnech, François Laroussinie, L. Lavagno, G. Lipari, F. Maranchi, P. Peti, J. Antonio de la Puente, N. Scaife, J. Sifakis, R. de Simone, M. Törngren, P. Verissimo, A.J. Wellings, R. Wilhelm, T.A.C. Willemse, W. Yi: Guidelines for a Graduate Curriculum on Embedded Software and Systems. ACM Trans. Embedded Comput. Syst. 4(3): 587-611 (2005)
14. R. Wilhelm: Imperative, prädikative und funktionale Programmierung (Kurzfassung). GI Jahrestagung 1982: 188-193
15. J. Messerschmidt, R. Wilhelm: Constructors for Composed Objects. Comput. Lang. 7(2): 53-59 (1982)
16. R. Wilhelm: Symbolische Programmausführung - Das aktuelle Schlagwort. Informatik Spektrum 6(3): 170 (1983)
17. J. Loeckx, K. Mehlhorn, R. Wilhelm: Grundlagen der Programmiersprachen. Teubner, 1986
18. G. Becker, B. Kuhn, D. Maurer, R. Wilhelm: SIATEX - eine interaktive Arbeitsumgebung für TEX. Innovative Informations-Infrastrukturen 1988: 162-169
19. J. Loeckx, K. Mehlhorn, R. Wilhelm: Foundations of Programming Languages. John Wiley, 1989
20. M. Baston, H.-J. Bach, A. Lucks-Baus, F. Müller, R. Wilhelm: Implementierung der funktionalen Programmiersprache HOPE mit Hilfe von Kombinatoren. Innovative Informations-Infrastrukturen 1988: 114-131