Verifying OCL specifications of UML models : tool support and compositionality

Kyas, M.

Bibliography

[1] Erika Ábrahám. *An Assertional Proof System for Multithreaded Java: Theory and Tool Support*. PhD thesis, Universiteit Leiden, 2005.

[2] Erika Ábrahám, Marcello M. Bonsangue, Frank S. de Boer, and Martin Steffen. Object connectivity and full abstraction for a concurrent calculus of classes. In *Proceedings of the First International Colloquium on Theoretical Aspects of Computing ICTAC 2004*, number 3704 in Lecture Notes in Computer Science, pages 38–52. Springer-Verlag, 2004.

[3] Erika Ábrahám, Frank S. de Boer, Willem-Paul de Roever, and Martin Steffen. A compositional operational semantics for Java_{MT}. In Dershowitz [48], pages 290–303.

[4] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers. Principles, Techniques, and Tools*. Addison Wesley Publishing Company, 1986.

[5] Pierre America and Frank S. de Boer. Reasoning about dynamically evolving process structures. *Formal Aspects of Computing*, 6(3):269–316, 1994.

[6] Demissie Bediye Aredo. *Formal Development of Open Distributed Systems: Integration of UML and PVS*. PhD thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, 2005.

[7] Thomas Baar. *Über die Semantikbeschreibung OCL-artiger Sprachen*. PhD thesis, Fakultät für Informatik, Universität Karlsruhe, 2003. Logos Verlag, Berlin.

[8] John Warner Backus. The syntax and semantics of the proposed international algebraic language of the Zuerich acm-gramm conference. In *ICIP Paris*, June 1959.

[9] Hubert Baumeister, Rolf Hennicker, Alexander Knapp, and Martin Wirsing. OCL component invariants. In N. Chaki, editor, *Proceedings of the 8th Monterey Workshop "Engineering Automation for Software Intensive System Integration"*, pages 208–215, Monteray, California, 2001. U.S. Naval Postgraduate School.

[10] Marcello M. Bonsangue and Joost N. Kok. Infinite intersection types. *Information and Computation*, 186(2):285–318, 2003.

[11] Grady Booch. *Object-Oriented Analysis and Design with Applications*. Benjamin Cummings, 2nd edition edition, 1993.

[12] Grady Booch. Growing the UML. *Software and Systems Modeling*, 1(2):157–160, December 2002.

[13] Grady Booch, James Rumbaugh, and Ivar Jacobson. *Unified Modelling Language User Guide*. Addison Wesley Longman, 1998.

[14] Egon Börger, Alessandra Cavarra, and Elvinia Riccobene. Modeling the dynamics of UML state machines. In Yuri Gurevich, Philipp W. Kutter, Martin Odersky, and Lothar Thiele, editors, *Abstract State Machines, Theory and Applications*, volume 1912 of *Lecture Notes in Computer Science*, pages 223–241. Springer-Verlag, 2000.

[15] Nicolas Bourbaki. *Élements de Mathématique*, volume 1. Hermann, Paris, 1954.

[16] Marius Bozga, Jean-Claude Fernandez, Lucian Ghirvu, Susanne Graf, Jean-Pierre Krimm, and Laurent Mounier. IF: A validation environment for timed asynchronous systems. In E. Allen Emerson and A. Prasad Sistla, editors, *Computer Aided Verification '00*, volume 1855 of *Lecture Notes in Computer Science*, pages 543–547. Springer-Verlag, 2000.

[17] Julian C. Bradfield, Juliana Küster Filipe, and Perdita Stevens. Enriching OCL using observational mu-calculus. In Ralf-Detlef Kutsche and Herbert Weber, editors, *5th International Conference on Fundamental Approaches to Software Engineering (FASE 2002), April 2002, Grenoble, France*, volume 2306 of *Lecture Notes in Computer Science*, pages 203–217. Springer-Verlag, 2002.

[18] Ruth Breu, Ursula Hinkel, Christoph Hofmann, Cornel Klein, Barbara Paech, Bernhard Rumpe, and Veronika Thurner. Towards a formalization of the unified modeling language. In Mehmet Aksit and Satoshi Matsuoka, editors, *Proceedings of ECOOP'97 — Object-Oriented Programming, 11th European Conference*, volume 1241 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[19] Achim D. Brucker and Burkhart Wolff. HOL-OCL: Experiences, consequences and design choices. In Jean-Marc Jézéquel, Heinrich Hussman, and Stephen Cook, editors, *UML 2002 - The Unified Modeling Language*, volume 2460 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 2002.

[20] Achim D. Brucker and Burkhart Wolff. A proposal for a formal OCL semantics in Isabelle/HOL. In Victor Carreño, César Muñoz, and Sofiène Tashar, editors, *15th International Conference of Theorem Proving in Higher Order Logics*, volume 2410 of *Lecture Notes in Computer Science*, pages 99–114. Springer-Verlag, 2002.

[21] Luca Cardelli and Peter Wegener. On understanding types, data abstraction, and polymorphism. *ACM Computing Surveys*, 17(4):471–522, 1985.

[22] María Victoria Cengarle and Alexander Knapp. Towards OCL/RT. In Lars-Henrik Eriksson and Peter A. Lindsay, editors, *FME 2002: Formal Methods - Getting IT Right, International Symposium of Formal Methods Europe, Copenhagen, Denmark, July 22-24, 2002, Proceedings*, volume 2391 of *Lecture Notes in Computer Science*, pages 390–409. Springer-Verlag, 2002.

[23] María Victoria Cengarle and Alexander Knapp. OCL 1.4/5 vs. 2.0 expressions: Formal semantics and expressiveness. *Software and Systems Modeling*, 3(1):9–30, 2004.

[24] Peter Pin-Shan Chen. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976.

[25] Anthony Neil Clark. Typechecking UML static models. In Robert B. France and Bernhard Rumpe, editors, *Proceedings of UML'99: The Unified Modeling Language — Beyond the Standard, Second International Conference*, volume 1723 of *Lecture Notes in Computer Science*, pages 503–517. Springer-Verlag, 1999.

[26] Anthony Neil Clark and Jos B. Warmer, editors. *Object Modelling with the OCL*, volume 2263 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[27] Lauren Clark. The Apollo 35th anniversary reception. *IEEE Control Systems Magazine*, 24(6):100–101, December 2004.

[28] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, Yorktown Heights, New York, May 1981. Springer-Verlag. Published in 1982.

[29] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[30] Adriana Beatriz Compagnoni. Higher-order subtyping and its decidability. *Information and Computation*, 191(1):41–113, 2004.

[31] Adriana Beatriz Compagnoni and Benjamin C. Pierce. Intersection types and multiple inheritance. *Mathematical Structures in Computer Science*, 6(5):469–501, October 1996.

[32] Computer Science Research Laboratory, Babes-Bolyai University of Cluj-Napoca, Romania. OCLE 1.0, 2003. `http://lci.cs.ubbcluj.ro/ocle/`.

[33] Steve Cook and John Daniels. *Designing Object Systems: Object-Oriented Modelling with Syntropy*. Prentice Hall, 1994.

[34] Steve Cook, Anneke Kleppe, Richard Mitchell, Bernhard Rumpe, Jos B. Warmer, and Alan Wills. The Amsterdam manifesto on OCL. In Clark and Warmer [26], pages 115–149.

[35] Thierre Coquand and Gérard Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, February/March 1988.

[36] Patrick Cousot and Rhadia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, January 1977.

[37] Elspeth Cusack. Refinement, conformance and inheritance. *Formal Aspects of Computing*, 3(2):129–141, June 1991.

[38] Ole-Johan Dahl. Can program proving be made practical? In Michaneh Amirchahy and Danièle Néel, editors, *Les Fondements de la Programmation*, pages 57–114. INRIA, 1977.

[39] Werner Damm, Bernhard Josko, Amir Pnueli, and Angelika Votintseva. Understanding UML: A formal semantics of concurrency and communication in real-time uml. In Frank S. de Boer, Marcello Bonsangue, Susanne Graf, and Willem-Paul de Roever, editors, *Formal Methods for Components and Objects*, volume 2852 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[40] Dennis Dams, Rob Gerth, and Orna Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19(2):253–291, 1997.

[41] Luca de Alfaro and Thomas A. Henzinger. Interface automata. In *Proceedings of the Ninth Annual Symposium on Foundations of Software Engineering (FSE)*, pages 109–120. ACM Press, 2001.

[42] Jaco W. de Bakker, Willem-Paul de Roever, and Grzegorz Rozenberg, editors. *Current Trends in Concurrency*, volume 224 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[43] Frank S. de Boer and Cees Pierik. Computer-aided specification and verification of annotated object-oriented programs. In Jacobs and Rensink [70], pages 163–177.

[44] Willem-Paul de Roever. The quest for compositionality — a survey of assertion-based proof systems for concurrent programs, Part 1: Concurrency based on shared variables. In *Proceedings of the IFIP Working Conference 1985: The Role of Abstract Models in Computer Science*, pages 181–207. North-Holland, 1985.

[45] Willem-Paul de Roever, Frank Siepke de Boer, Ulrich Hannemann, Jozef Hooman, Yassine Lakhnech, Mannes Poel, and Job Zwiers. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Number 54 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.

[46] Willem-Paul de Roever and Kai Engelhardt. *Data Refinement: Model-Oriented Proof Methods and their Comparison*. Number 47 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1998.

[47] María del Mar Gallardo, Pedro Merino, and Ernesto Pimentel. Debugging UML designs with model checking. *Journal of Object Technology*, 1(2):101–117, July 2002. `http://www.jot.fm/issues/issue_2002_07/article1`.

[48] Nachum Dershowitz, editor. *Proceedings of the International Symposium on Verification – Theory and Practice – Honoring Zohar Manna's 64th Birthday (Taormina, Italy, June 2003)*, volume 2772 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[49] Desmond Francis D'Souza and Alan Cameron Wills. *Objects, Components, and Frameworks with UML: The Catalysis*^{SM} *Approach*. The Addison Wesley object technology series. Addison Wesley Longman, Inc., 1998.

[50] James Clark (ed.). *XSL Transformations (XSLT) Version 1.0*. W3C, November 1999. Available for download at `http://www.w3.org/TR/xslt`.

[51] Andy Evans, Robert France, Kevin Lano, and Bernhard Rumpe. Developing the UML as a formal modelling notation. In Jean Bézevin and Pierre-Alain Muller, editors, *The Unified Modelling Language UML'98 — Beyond the Notation*, volume 1618 of *Lecture Notes in Computer Science*, pages 297–307, Berlin, Heidelberg, New-York, June 1998. Springer-Verlag.

[52] Harald Fecher, Marcel Kyas, Frank S. de Boer, and Willem-Paul de Roever. Compositional operational semantics of an UML-kernel-model language. In Peter D. Mosses and Irek Ulidowski, editors, *Proceedings of the Second Workshop on Structural Operational Semantics (SOS 2005)*, Electronic Notes in Theoretical Computer Science. Elsevier, 2005. Accepted for publication.

[53] Harald Fecher, Jens Schönborn, Marcel Kyas, and Willem-Paul de Roever. 29 new unclarities in the semantics of UML 2.0 state machines. In Kung-Kiu Lau and Richard Banach, editors, *Formal Methods and Software Engineering (ICFEM 2005)*, volume 3785 of *Lecture Notes in Computer Science*, pages 52–65. Springer-Verlag, 2005.

[54] Stephan Flake and Wolfgang Mueller. Formal semantics of OCL messages. In Peter Schmitt, editor, *Proceedings of the Workshop OCL 2.0 – Industry standard or scientific playground?*, volume 102 of *Electronic Notes in Theoretical Computer Science*, pages 77–97. Elsevier, November 2004.

[55] Martin Fowler, Martin L. Griss, Luke Hohmann, Ian Hopper, Rebecca Joos, and William F. Opdyke. *Refactoring: Improving The Design of Existing Code*. Addison-Wesley, 1999.

[56] Adele Goldberg and David Robson. *Smalltalk-80: The Language*. Addison-Wesley, 1989.

[57] James Gosling, Bill Joy, and Guy L. Steele. *The Java Language Specification*. Addison-Wesley, 3rd edition, 2005.

[58] Martin Große-Rhode. Integrating semantics for object-oriented system models. In F. Orejas, P. G. Spirakis, and J. van Leeuwen, editors, *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP 2001)*, number 2076 in Lecture Notes in Computer Science, pages 40–60. Springer Verlag, 2001.

[59] Yuri Gurevich. Evolving algebra 1993: Lipari guide. In Egon Börger, editor, *Specification and Validation Methods*, pages 9–36. Oxford University Press, 1995.

[60] David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, July 1987.

[61] David Harel and Eran Gery. Executable object modeling with statecharts. *Computer*, 30(7):31–42, July 1997.

[62] Rolf Hennicker, Heinrich Hußmann, and Michel Bidoit. On the precise meaning of OCL constraints. In Clark and Warmer [26], pages 69–84.

[63] Carl Hewitt. Viewing control structures as patterns of passing messages. Technical Report 410, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, December 1976.

[64] Jozef Hooman. *Specification and Compositional Verification of Real-Time Systems*, volume 558 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991.

[65] Jozef Hooman. Compositional verification of real-time applications. In Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli, editors, *Compositionality: The Significant Difference, Proceedings of the International Symposium COMPOS '97, Malente, Germany, September 7–12, 1997*, volume 1536 of *Lecture Notes in Computer Science*, pages 276–300. Springer-Verlag, 1998.

[66] Jozef Hooman and Willem-Paul de Roever. The quest goes on: A survey of proof systems for partial correctness of CSP. In de Bakker et al. [42], pages 343–395.

[67] Jozef Hooman and Mark van der Zwaag. A semantics of communicating reactive objects with timing. In Susanne Graf, Øystein Haugen, Ileana Ober, and Bran Selic, editors, *1st Workshop on Specification and Validation of UML Models for Real Time and Embedded Systems, SVERTS 2003*, Verimag technical report 2003/10/22. Verimag, 2003. Available online at `http://www-verimag.imag.fr/EVENTS/2003/SVERTS/`.

[68] Heinrich Hußmann, Birgit Demuth, and Frank Finger. Modular architecture for a toolset supporting ocl. *Science of Computer Programming*, 44(1):51–69, 2002.

[69] Daniel Jackson. Alloy: A lightweight object modelling notation. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(2):256–290, April 2002.

[70] Bart Jacobs and Arend Rensink, editors. *Formal Methods for Open Object-Based Distributed Systems V*. Kluwer Academic Publishers, 2002.

[71] Ivar Jacobson, Magnus Christerson, and Patrick Jonsson. *Object-Oriented Software Engineering — A Use Case Driven Approach*. Addison-Wesley, 1992.

[72] Einar Broch Johnsen and Olaf Owe. A compositional formalism for object viewpoints. In Jacobs and Rensink [70], pages 45–60.

[73] Chris W. Johnson. The natural history of bugs: Using formal methods to analyse software related failures in space missions. In J.S. Fitzgerald, I.J. Hayes, and A. Tarlecki, editors, *Proc. Formal Methods 2005*, volume 3582 of *Lecture Notes in Computer Science*, pages 9–25. Springer-Verlag, 2005.

[74] Stephen Johnson. Lint, a C program checker. Technical Report Computer Science Technical Report 65, Bell Laboratories, December 1977.

[75] Cliff B. Jones. *Systematic Software Development using VDM*. Prentice Hall, 1990.

[76] Bengt Jonsson. A model and proof system for asynchronous networks. In *Proceeding of the Fourth Annual ACM Symposium on Principles of Distributed Computing*, pages 49–58, Minaki, Ontario, Canada, 1985. ACM Press.

[77] Stephen Cole Kleene. *Introduction to Metamathematics*. North Holland, 1952.

[78] Anneke Kleppe and Jos B. Warmer. The semantics of the OCL action clause. In Clark and Warmer [26], pages 213–227.

[79] Alexander Knapp. *A Formal Approach to Object-Oriented Software Engineering*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.

[80] Cris Kobryn. UML 3.0 and the future of modeling. *Software and Systems Modeling*, 3(1):4–8, March 2004.

[81] Marcel Kyas. A compositional proof of the sieve of Eratosthenes in PVS. Technical report, Institut für Informatik, Christian-Albrechts-Universität, Kiel, Germany, 2004. Available at `http://www.informatik.uni-kiel.de/~mky/`.

[82] Marcel Kyas. An extended type system for OCL supporting templates and transformations. In Martin Steffen and Gianluigi Zavattaro, editors, *Formal Methods for Open Object-Based Distributed Systems (FMOODS 2006)*, volume 3535 of *Lecture Notes in Computer Science*, pages 83–98. Springer-Verlag, 2005.

[83] Marcel Kyas and Frank S. de Boer. Compositional specification and verification of UML models. In Paul Pettersson and Wang Yi, editors, *Proceedings of the 16th Nordic Workshop on Programming Theory*, pages 34–35, Box 377, SE-751 05 Uppsala, Sweden, October 2004. Department of Information Technology, Uppsala University. Technical Report 2004-041.

[84] Marcel Kyas and Frank S. de Boer. On message specification in OCL. In Frank S. de Boer and Marcello Bonsangue, editors, *Proceedings of the Workshop on the Compositional Verification of UML Models (CVUML)*, volume 101 of *Electronic Notes in Theoretical Computer Science*, pages 73–93. Elsevier, November 2004.

[85] Marcel Kyas, Frank S. de Boer, and Willem-Paul de Roever. A compositional trace logic for behavioural interface specifications. *Nordic Journal of Computing*, 12(2):116–132, 2005.

[86] Marcel Kyas, Harald Fecher, Frank S. de Boer, Mark van der Zwaag, Jozef Hooman, Tamarah Arons, and Hillel Kugler. Formalizing UML models and OCL constraints in PVS. In Gerald Lüttgen, Natividad Martínez Madrid, and Michael Mendler, editors, *Proceedings of Semantic Foundations of Engineering Design Languages (SFEDL 2004)*, volume 115 of *Electronic Notes in Theoretical Computer Science*, pages 39–47. Elsevier, 2005.

[87] Marcel Kyas and Jozef Hooman. Compositional verification of the MARS case study using PVS. Technical report, Institut für Informatik, Christian-Albrechts-Universität, Kiel, Germany, 2005. Available at `http://www.informatik.uni-kiel.de/~mky/pvs/mars.html`.

[88] Marcel Kyas and Jozef Hooman. Compositional verification of timed components using PVS. In Bettina Biel, Matthias Book, and Volker Gruhn, editors, *Software Engineering 2006*, volume P-79 of *Lecture Notes in Informatics*, pages 143–154. Gesellschaft für Informatik e.V., Kollen Verlag, Bonn, 2006.

[89] Leslie Lamport. *Specifying Systems*. Addison-Wesley, 2002.

[90] Leslie Lamport and Lawrence C. Paulson. Should your specification language be typed? *ACM Transactions on Programming Languages and Systems*, 21(3):502–526, May 1999.

[91] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. Model-checking for real-time systems. In Horst Reichel, editor, *Proceedings of Fundamentals of Computation Theory*, volume 965 of *Lecture Notes in Computer Science*, pages 62–88. Springer-Verlag, 1995.

[92] Diego Latella, Istvan Majzik, and Mieke Massink. Automatic verification of a behavioural subset of UML statechart diagrams using the SPIN model-checker. *Formal Aspects of Computing*, 11(6):637–664, 1999.

[93] Daniel Leivant. Higher order logic. In Dov M. Gabbay, Christopher John Hogger, J. A. Robinson, and Jörg H. Siekmann, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 2 – Deduction Methodologies, pages 229–321. Oxford University Press, 1994.

[94] Barbara H. Liskov and Jeannette M. Wing. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 16(6):1811–1841, November 1994.

[95] David B. MacQueen. Should ML be object-oriented? *Formal Aspects of Computing*, 13(3–5):214–232, 2002.

[96] Satoshi Matsuoka and Akinori Yonezawa. Analysis of inheritance anomaly in object-oriented concurrent programming languages. In Gul Agha, Peter Wegner, and Akinori Yonezawa, editors, *Research Directions in Concurrent Object-Oriented Programming*, pages 107–150. MIT Press, 1993.

[97] Bertrand Meyer. *Eiffel: The Language*. Prentice Hall, 1992.

[98] Bertrand Meyer. *Object-Oriented Software Construction*. Prentice Hall, 2nd edition, 1997.

[99] Leonid Mikhajlov and Emil Sekerinski. A study of the fragile base class problem. In Eric Jul, editor, *ECOOP'98 - Object-Oriented Programming, 12th European Conference, Brussels, Belgium, July 20-24, 1998, Proceedings*, volume 1445 of *Lecture Notes in Computer Science*, pages 355–382. Springer-Verlag, 1998.

[100] Michael Möller, Ernst-Rüdiger Olderog, Holger Rasch, and Heike Wehrheim. Linking CSP-OZ with UML and Java: A case study. In Eerke A. Boiten, John Derrick, and Graeme Smith, editors, *Integrated Formal Methods (IFM 2004)*, volume 2999 of *Lecture Notes in Computer Science*, pages 267–286. Springer-Verlag, 2004.

[101] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[102] Iulian Ober, Susanne Graf, and Ileana Ober. Validation of UML models via a mapping to communicating extended timed automata. In Susanne Graf and Laurent Mounier, editors, *Model Checking Software: 11th International SPIN Workshop*, volume 2989 of *Lecture Notes in Computer Science*, pages 127–145. Springer-Verlag, 2004.

[103] Object Management Group. *OMG XMI Metadata Interchange (XMI) Specification*, June 2000. Version 1.0. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/00-06-01`.

[104] Object Management Group. *OMG XMI Metadata Interchange (XMI) Specification*, November 2000. Version 1.1. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/00-11-02`.

[105] Object Management Group. *OMG Unified Modeling Language Specification*, September 2001. Version 1.4. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/2001-09-67`.

[106] Object Management Group. *OMG XMI Metadata Interchange (XMI) Specification*, January 2002. Version 1.2. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/02-01-01`.

[107] Object Management Group. *UML™ Profile for Schedulability, Performance, and Time Specification*, March 2002. Available for download at `http://cgi.omg.org/cgi-bin/doc?ptc/2002-03-02`.

[108] Object Management Group. *OMG XMI Metadata Interchange (XMI) Specification*, May 2003. Version 2.0. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/03-05-02`.

[109] Object Management Group. *OMG XMI Metadata Interchange (XMI) Specification*, May 2003. Version 1.3. Available for download at `http://cgi.omg.org/cgi-bin/doc?formal/03-05-01`.

[110] Object Management Group. *UML 2.0 Infrastructure Specification*, November 2004. `http://www.omg.org/cgi-bin/doc?ptc/2004-10-14`.

[111] Object Management Group. *UML 2.0 Superstructure Specification*, October 2004. `http://www.omg.org/cgi-bin/doc?ptc/2004-10-02`.

[112] Object Management Group. *FTF Report of the OCL 2.0 Finalization Task Force*, June 2005. Available for download at `http://www.omg.org/cgi-bin/doc?ptc/2005-06-05`.

[113] Object Management Group. *OCL 2.0 Specification*, June 2005. Available for download at `http://www.omg.org/cgi-bin/doc?ptc/2005-06-06`.

[114] Ernst-Rüdiger Olderog. Process theory: Semantics, specifications and verification. In de Bakker et al. [42], pages 442–509.

[115] Ernst-Rüdiger Olderog. *Nets, Terms and Formulas: Three Views on Concurrent Processes and their Relationship*. Number 23 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1991.

[116] Omega Consortium. Omega: Correct development of real-time embedded systems, November 2003. Web-page at `http://www-omega.imag.fr`.

[117] Omega Consortium. Medium altitude reconnaissance system. Webpage at `http://www-omega.imag.fr/cs/MARS/MARS.php`, 2005.

[118] William F. Opdyke. *Refactoring Object-Oriented Frameworks*. PhD thesis, University of Illinois at Urbana-Campaign, 1992.

[119] William F. Opdyke and Ralph E. Johnson. Refactoring: An aid in designing frameworks and evolving object-oriented systems. In *Proceedings of SOOPPA '90: Symposium on Object-Oriented Programming Emphasizing Practical Applications*, September 1990.

[120] Olaf Owe and Isabelle Ryl. Reasoning control in presence of dynamic classes. In *Proceedings of the 12th Workshop in Programming Theory, October 11–13, 2000, Bergen, Norway*, 2000.

[121] Sam Owre, John M. Rushby, and Natarajan Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *Automated Deduction – CADE-11*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer-Verlag, 1992.

[122] Sam Owre, John M. Rushby, Natarajan Shankar, and Friedrich von Henke. Formal verification for fault-tolerant architectures: Prolegomena to the design of PVS. *IEEE Transactions on Software*, 21(2):107–125, 1995.

[123] Sam Owre and Natarajan Shankar. The formal semantics of PVS. Technical Report CSL-97-2R, SRI International Computer Science Laboratory, Menlo Park CA 94025 USA, 1999. August 1997, Revised March 1999.

[124] Sam Owre, Natarajan Shankar, John M. Rushby, and David W.J. Stringer-Calvert. *PVS Language Reference version 2.4*. SRI International, Computer Science Laboratory, Menlo Park, CA, dec 2001.

[125] Benjamin C. Pierce. *Programming with Intersection Types and Bounded Polymorphism*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1991.

[126] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[127] Cees Pierik and Frank S. de Boer. A syntax-directed hoare logic for object-oriented programming concepts. In Elie Najm, Uwe Nestmann, and Perdita Steven, editors, *Formal Methods for Open Object-Based Distributed Systems (FMOODS 2003)*, volume 2884 of *Lecture Notes in Computer Science*, pages 67–78. Springer-Verlag, 2003.

[128] Cees Pierik and Frank S. de Boer. A proof outline logic for object-oriented programming. *Theoretical Computer Science*, 343(3):413–442, October 2005.

[129] Amir Pnueli and Tamarah Arons. TLPVS: A PVS-based LTL verification system. In Dershowitz [48], pages 598–625.

[130] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *Proceedings of the 5th International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351, Paris, April 1981. Springer-Verlag.

[131] Rational Software Corporation. *UML Summary*, March 1997.

[132] Gianna Reggio, Maura Cerioli, and Egidio Astesiano. An algebraic semantics of UML supporting its multiview approach. In D. Heylen, A. Nijholt, and G. Scollo, editors, *Proc. AMiLP 2000*, 2000. Twente Workshop on Language Technology n. 16, Enschede, University of Twente.

[133] Mark Richters. *A Precise Approach to Validating UML Models and OCL Constraints*. PhD thesis, Universtät Bremen, 2002. Logos Verlag, Berlin, BISS Monographs, No. 14.

[134] Mark Richters and Martin Gogolla. A semantics for OCL pre- and postconditions. In Anthony Neil Clark and Jos B. Warmer, editors, *UML 2.0 — The Future of the UML Object Constraint Language (OCL)*, October 2000. Published at `http://www.comp.brad.ac.uk/research/OCL2000/index.html` (October 17, 2005).

[135] Mark Richters and Martin Gogolla. OCL: Syntax, semantics, and tools. In Clark and Warmer [26], pages 42–68.

[136] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, 1990.

[137] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.

[138] John M. Rushby and David W.J. Stringer-Calvert. A less elementary tutorial for the PVS specification and verification system. Technical Report CSL-95-10, SRI International Computer Science Laboratory, 1996.

[139] Michael Schenke and Ernst-Rüdiger Olderog. Transformational design of real-time systems — part 1: From requiremens to program specification. *Acta Informatika*, 36:1–65, 1999.

[140] Jens Schönborn. Formal semantics of UML 2.0 behavioral state machines. Diploma Thesis, Christian-Albrechts-Universität zu Kiel, April 2005.

[141] Andy Schürr. A new type checking approach for OCL 2.0? In Clark and Warmer [26], pages 21–40.

[142] Bran Selic, Garth Gullekson, and Paul T. Ward. *Real-Time Object-Oriented Modeling*. John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, 1994.

[143] Raymond Merrill Smullyan. *First Order Logic*. Springer-Verlag, 1968.

[144] Neelam Soundararajan. Axiomatic semantics of communicating sequential processes. *ACM TOPLAS*, 6:647–662, 1984.

[145] Martin Steffen. *Polarized Higher-Order Subtyping*. PhD thesis, Technische Fakutät, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1997.

[146] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, special edition, 2000.

[147] Wolfgang Thomas. Languages, automata, and logic. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 385–455. Springer-Verlag, 1997.

[148] Issa Traoré. An outline of PVS semantics for UML statecharts. *Journal of Universal Computer Science*, 6(11):1088–1108, November 2000. `http://www.jucs.org/jucs_6_11/an_outline_of_pvs`.

[149] Mark van der Zwaag and Jozef Hooman. A semantics of communicating reactive objects with timing. *Journal on Software Tools for Technology Transfer*, 2005. Accepted for Publication in STTT.

[150] Dániel Varró. A formal semantics of UML statecharts by model transition systems. In Andrea Corradini, Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Graph Transformation: First International Conference, ICGT 2002, Barcelona, Spain, October 7-12, 2002. Proceedings*, volume 2505 of *Lecture Notes in Computer Science*, pages 378–392. Springer-Verlag, 2002.

[151] Michael von der Beeck. A structured operational semantics for UML-statecharts. *Software and Systems Modeling*, 1(2):130–141, December 2002.

[152] Philip L. Wadler. Theorems for free! In *Fourth International Conference on Functional Programming Languages and Computer Architecture*, pages 347–359. ACM Press, 1989.

[153] Jos B. Warmer. OCL 1.4 syntax checker, 2001. `http://www.klasse.nl/ocl`.

[154] Jos B. Warmer and Anneke G. Kleppe. *The Object Constraint Language: Precise Modeling With UML*. Addison-Wesley, 1998.

[155] Jos B. Warmer and Anneke G. Kleppe. *The Object Constraint Language: Getting your models ready for MDA*. Addison-Wesley, 2nd edition edition, 2003.

[156] Pierre Wolper. The meaning of "formal": From weak to strong formal methods. *International Journal on Software Tools for Technology Transfer*, 1(1–2):6–8, December 1997.

[157] François Yergeau, Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, and Eve Maler. *Extensible Markup Language (XML) 1.0*. W3C (World Wide Web Consortium), 3rd edition edition, February 2004. Available at `http://www.w3.org/TR/2004/REC-xml-20040204/`.

[158] Job Zwiers. *Compositionality, Concurrency and Partial Correctness – Proof Theories for Networks of Processes, and Their Relationship*, volume 321 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.

Summary

Embedded real-time systems are small computer systems which are used to control an increasing number of devices in every-day life. They are embedded in, for example, DVD players, microwave ovens, antilock braking systems, and autopilots. It is important that these devices always perform their function correctly in case the life of people depends upon the software used in them. Moreover, high costs are usually involved in recalling defective devices, for example, in cars. Therefore, it is desirable that these systems are formally validated, that is, a *proof* of the correct functioning of the system is constructed. Such a proof is especially important for real-time systems, because they not only need to function correctly, but also deliver their reactions *on time*. For example, an air-bag should not only inflate when a car crashes, but it should inflate milliseconds after the impact, and not seconds.

Ever since the first embedded systems were developed, their complexity has been steadily increasing. In order to control and understand this complexity different methods are used to describe the structure, the behaviour, and the requirements of software systems. Such methods are provided by the *Unified Modelling Language* (UML) and its *Object Constraint Language* (OCL) as notations (as diagrams) for describing complex *object-oriented* software systems, where the parts of these systems during execution are called *objects*. Objects react to *messages* they exchange among each other and with their *environment*, that is, with their external world. This exchange of messages is considered to be (part of) their *behaviour*.

UML provides the schema language of *class diagrams* for describing the structure, that is, the parts of the system and which parts may communicate with each other, and the notation of *state machines* for describing the behaviour of a system or its parts. OCL is used to describes the *requirements* on the system. Requirements are the properties a system has to satisfy and describe its *correct* functioning from the point of view of these given requirements.

In order to enable the development and the formal validation of these systems we have to define a *formal semantics* for the notations of UML and OCL. This means, we assign a precise meaning to the constructs of UML and OCL. This is necessary, because at present UML notations have no precise meaning. To this end, we define an unambiguous subset of UML class diagrams and define a precise mathematical semantics for this subset in Chapter 2.

UML and OCL are *typed languages*. This means that there are so-called *typing rules* on diagrams and expressions which describe when they are well-formed and therefore

have a meaning that makes sense. In Chapter 3 we show that these rules are too inflexible for writing requirements while the system is still under development. Namely, development causes changes in the system which, according to the typing rules, unexpectedly render requirements ill-formed. As a consequence, these requirements are considered nonsensical in UML. However, in our semantics they have a well-defined meaning, which has not been changed by the development step. To overcome this problem we propose extensions of the typing rules (based on so-called intersection types, union types, and bounded operator abstraction) which also improve the integration of the OCL into the UML, and which considers more requirements as well-formed.

We use *logic* to formalise the meaning of UML diagrams and OCL expressions in order to enable their formal validation. Logic makes the use of *interactive theorem provers* possible. Theorem provers assist in constructing proofs of the correct functioning of systems. This means that a system and its requirements have to be *translated* into logic. The result of this translation should be of a form that allows one to exploit all automated reasoning facilities offered by the theorem prover in finding a proof, because otherwise the construction of proofs quickly becomes complex, burdensome, and (economically) infeasible. In Chapter 4 we describe such a translation, performed by a computer program, into the input language of the theorem prover PVS and show why the translator preserves the meaning of the system and its requirements.

In order to support the specification of systems during early stages of design, we have analysed the semantics of OCL Message Expressions in Chapter 5. Message expressions specify whether messages have been sent by objects. These have been found to be inadequate. Therefore, we propose introducing *history variables* to OCL. History variables allow not only to specify and reason about the messages sent during the invocation of an operation, but also about the history of *all* messages sent and received by an object. We also show that everything which can be expressed by message expressions can also be expressed with history variables.

We strictly separate local specifications, which are requirements on the internal state of objects (and play the role of so-called *data invariants*), from local behavioural specifications, which describe the messages sent and received by an object. At a third level, we introduce global specifications which specify how objects in a system may interact.

This formalisation leads to a compositional history-based specification formalism, for which we give a compositional proof rule in Chapter 6. A specification is called *compositional* if the function of a system can be derived from the functions of its parts and the way they are put together. The main problem to solve here is the treatment of the evolution of object structures. Object structures change because objects learn about other objects during their lifetime, which enables them to communicate with new acquaintances; especially, when objects create new objects.

Finally, in Chapter 7 we extend this history-based formalism to real-time specifications. We specify a part of a *medium altitude reconnaissance system*, which is deployed by the Royal Dutch Air-Force, and prove its correctness. This example shows that the methods described in this thesis can be applied in principle to real-world case studies.

Samenvatting

Ingebedde real-time systemen zijn (kleine) computer systemen die ertoe dienen de apparaten waarin ze ingebed zijn te helpen (be)sturen. Voorbeelden van zulke apparaten zijn DVD spelers, automatische remmen, autopiloten, mobiele telefoons en Magnetic-Resonance scanners. Zulke ingebedde systemen komen meer en meer voor en worden in hoog tempo snel complexer. Ook komt het steeds vaker voor dat mensenlevens van het correct functioneren van de door hen gestuurde apparaten afhangen. Deze ontwikkeling is niet meer te stuiten. Daarom is het belangrijk dat zulke apparaten correct functioneren. En dat hangt weer af van het correcte functioneren van de hen sturende real-time systemen.

Aangezien deze systemen alom tegenwoordig zijn, zijn er industriële standaards ontwikkeld on hun functionaliteit te beschrijven. Een veel gebruikte standaard hiervoor is de UML (voor Unified Modeling Language–de naam zegt het al) en in het bijzonder zijn deeltaal OCL (voor Object Constraint Language), die ertoe dient de bedoelde betekenis van constructies in UML nader vast te leggen.

Jammer genoeg is noch de betekenis van UML, noch die van OCL eenduidig vastgelegd. (Sommige bronnen beweren dat dit met opzet gebeurd is om tegenstrijdige industriële belangen te dienen). Het is duidelijk dat als je niet precies weet wat een bepaalde taalkonstruktie betekent, je hem ook niet met 100 % zekerheid kunt gebruiken om een apparaat te sturen waar mensenlevens van afhangen.

Om in deze situatie verandering te brengen is dit proefschrift geschreven.

Het beschrijft een formele, dat wil zeggen, in wiskundige zin exacte, semantiek voor de taalconstructies van UML en OCL, en voorziet deze talen van een zinvol typesysteem dat ertoe dient om aan te geven in welke context een UML of OCL taalkonstruktie zinvol te gebruiken is. Dit type systeem is, als onderdeel vam dit proefschrift, geïmplementeerd, zodat het voldoen aan de betreffende typerings regels elektronisch kan worden gecheckt.

Om te bewijzen dat deze semantiek eenduidig is, is hij omgezet in de specificatietaal van PVS, een elektronisch systeem dat bewijzen van wiskundige stellingen op hun correctheid checkt en dat veel gebruikt wordt om er correctheidsbewijzen van programma's elektronisch mee te controleren.

Vervolgens worden in dit proefschrift een paar karakteristieke toepassingen van UML korrekt bewezen, waarbij de gebruikte semantiek die is welke in dit proefschrift vastgelegd wordt, de architectuur van deze toepassingen in UML gegeven wordt en hun functionaliteit in OCL wordt gespecificeerd.

De eerste toepassing betreft een programma voor de Zeef van Eratosthenes, dat ertoe dient de priemgetallen te genereren. Dit ontleent zijn belang aan het feit dat de desbetreffende "zeef" zich in principe een onbegrensd aantal malen (recursief) oproepen kan. Er wordt aangegeven hoe dit probleem in PVS gecodeerd kan worden, waarna de correctheid van dit programma met behulp van PVS bewezen wordt.

De tweede toepassing is ontleend aan een programma dat gebruikt wordt door de Koninklijke Luchtmacht in hun verkenningsvliegtuigen om daar zeer nauwkeurige fotos mee te maken. Wanneer namelijk vanuit straaljagers gefotografeerd wordt, moet voor nauwkeurige fotos een compensatie-mechanisme ingebouwd worden in verband met de tijdens een opname afgelegde afstand; die moet door bewegende spiegels gecompenseerd worden. Van het centrale deel van het elektronische ingebedde real-time systeem dat de beweging van deze spiegels regelt wordt een nauwkeurige specificatie in OCL gegeven en met behulp van PVS bewezen dat de UML beschrijving van de architectuur van het desbetreffende besturingssysteem aan deze specificatie voldoet.

Daarmee wordt aangetoond dat deze semantieken en hun omzetting in PVS zich er in principe toe lenen om er industriële toepassingen, waarvan architectuur en functionaliteit in UML en OCL beschreven zijn, mee korrekt te bewijzen.

Curriculum Vitæ

January 30, 1975 Born in Pinneberg, Germany.

August 1981–July 1985 Hans-Clausen-Schule, Pinneberg.

August 1985–June 1994 Diploma qualifying for university admission (Abitur) from *Johannes-Brahms-Schule*, Pinneberg, (major fields of study: mathematics and chemistry).

July 1994–September 1995 Alternative civilian service at the nursing home Haus am Rosengarten, Pinneberg.

October 1995–November 2000 Diploma from Christian-Albrechts-Universität zu Kiel (CAU) in computer science under supervision of Yassine Lakhnech and Willem-Paul de Roever. Title of diploma thesis: "Verifikation parameterisierter Netzwerke durch Abstraktion (Verification of parameterised networks by abstraction)". Minor subject: electrical engineering.

January 1997–March 1999 Student assistant (Wissenschaftliche Hilfskraft) at CAU, Institute of Economics (Operations Research), implementing a distributed version of a resource constraint scheduling problem.

April 1999–December 2000 Student assistant at CAU, Institute of Computer Science and Applied Mathematics (Software Technology), implementing static analysers for sequential function charts.

October 2000–December 2001 Assistant professor (nebenamtlicher Dozent) lecturing on *Algorithms and data structures* at FH Nordakademie.

January 2001–today Researcher at Christian-Albrechts-Universität zu Kiel, working for the IST-project *Omega* (IST-2001-33522), DFG/NWO-project *Mobi-J* (RO-1122/9-1 and RO1122/9-2), and DFG-project *SFC-Check* (LA-1021/6-1).

Current address: Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
24098 Kiel
Germany

Titles in the IPA Dissertation Series

Titles in the IPA Dissertation Series are *not* available from Lehmanns Media. Please contact the IPA Secretariat (`http://www.win.tue.nl/ipa/`) for help on obtaining a dissertation from this list.

J.O. Blanco. *The State Operator in Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1996-01

A.M. Geerling. *Transformational Development of Data-Parallel Algorithms*. Faculty of Mathematics and Computer Science, KUN. 1996-02

P.M. Achten. *Interactive Functional Programs: Models, Methods, and Implementation*. Faculty of Mathematics and Computer Science, KUN. 1996-03

M.G.A. Verhoeven. *Parallel Local Search*. Faculty of Mathematics and Computing Science, TUE. 1996-04

M.H.G.K. Kesseler. *The Implementation of Functional Languages on Parallel Machines with Distrib. Memory*. Faculty of Mathematics and Computer Science, KUN. 1996-05

D. Alstein. *Distributed Algorithms for Hard Real-Time Systems*. Faculty of Mathematics and Computing Science, TUE. 1996-06

J.H. Hoepman. *Communication, Synchronization, and Fault-Tolerance*. Faculty of Mathematics and Computer Science, UvA. 1996-07

H. Doornbos. *Reductivity Arguments and Program Construction*. Faculty of Mathematics and Computing Science, TUE. 1996-08

D. Turi. *Functorial Operational Semantics and its Denotational Dual*. Faculty of Mathematics and Computer Science, VUA. 1996-09

A.M.G. Peeters. *Single-Rail Handshake Circuits*. Faculty of Mathematics and Computing Science, TUE. 1996-10

N.W.A. Arends. *A Systems Engineering Specification Formalism*. Faculty of Mechanical Engineering, TUE. 1996-11

P. Severi de Santiago. *Normalisation in Lambda Calculus and its Relation to Type Inference*. Faculty of Mathematics and Computing Science, TUE. 1996-12

D.R. Dams. *Abstract Interpretation and Partition Refinement for Model Checking*. Faculty of Mathematics and Computing Science, TUE. 1996-13

M.M. Bonsangue. *Topological Dualities in Semantics*. Faculty of Mathematics and Computer Science, VUA. 1996-14

B.L.E. de Fluiter. *Algorithms for Graphs of Small Treewidth*. Faculty of Mathematics and Computer Science, UU. 1997-01

W.T.M. Kars. *Process-algebraic Transformations in Context*. Faculty of Computer Science, UT. 1997-02

P.F. Hoogendijk. *A Generic Theory of Data Types*. Faculty of Mathematics and Computing Science, TUE. 1997-03

T.D.L. Laan. *The Evolution of Type Theory in Logic and Mathematics*. Faculty of Mathematics and Computing Science, TUE. 1997-04

C.J. Bloo. *Preservation of Termination for Explicit Substitution*. Faculty of Mathematics and Computing Science, TUE. 1997-05

J.J. Vereijken. *Discrete-Time Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1997-06

F.A.M. van den Beuken. *A Functional Approach to Syntax and Typing*. Faculty of Mathematics and Informatics, KUN. 1997-07

A.W. Heerink. *Ins and Outs in Refusal Testing*. Faculty of Computer Science, UT. 1998-01

G. Naumoski and W. Alberts. *A Discrete-Event Simulator for Systems Engineering*. Faculty of Mechanical Engineering, TUE. 1998-02

J. Verriet. *Scheduling with Communication for Multiprocessor Computation*. Faculty of Mathematics and Computer Science, UU. 1998-03

J.S.H. van Gageldonk. *An Asynchronous Low-Power 80C51 Microcontroller*. Faculty of Mathematics and Computing Science, TUE. 1998-04

A.A. Basten. *In Terms of Nets: System Design with Petri Nets and Process Algebra*. Faculty of Mathematics and Computing Science, TUE. 1998-05

E. Voermans. *Inductive Datatypes with Laws and Subtyping – A Relational Model*. Faculty of Mathematics and Computing Science, TUE. 1999-01

H. ter Doest. *Towards Probabilistic Unification-based Parsing*. Faculty of Computer Science, UT. 1999-02

J.P.L. Segers. *Algorithms for the Simulation of Surface Processes*. Faculty of Mathematics and Computing Science, TUE. 1999-03

C.H.M. van Kemenade. *Recombinative Evolutionary Search*. Faculty of Mathematics and Natural Sciences, UL. 1999-04

E.I. Barakova. *Learning Reliability: a Study on Indecisiveness in Sample Selection*. Faculty of Mathematics and Natural Sciences, RUG. 1999-05

M.P. Bodlaender. *Scheduler Optimization in Real-Time Distributed Databases*. Faculty of Mathematics and Computing Science, TUE. 1999-06

M.A. Reniers. *Message Sequence Chart: Syntax and Semantics*. Faculty of Mathematics and Computing Science, TUE. 1999-07

J.P. Warners. *Nonlinear approaches to satisfiability problems*. Faculty of Mathematics and Computing Science, TUE. 1999-08

J.M.T. Romijn. *Analysing Industrial Protocols with Formal Methods*. Faculty of Computer Science, UT. 1999-09

P.R. D'Argenio. *Algebras and Automata for Timed and Stochastic Systems*. Faculty of Computer Science, UT. 1999-10

G. Fábián. *A Language and Simulator for Hybrid Systems*. Faculty of Mechanical Engineering, TUE. 1999-11

J. Zwanenburg. *Object-Oriented Concepts and Proof Rules*. Faculty of Mathematics and Computing Science, TUE. 1999-12

R.S. Venema. *Aspects of an Integrated Neural Prediction System*. Faculty of Mathematics and Natural Sciences, RUG. 1999-13

J. Saraiva. *A Purely Functional Implementation of Attribute Grammars*. Faculty of Mathematics and Computer Science, UU. 1999-14

R. Schiefer. *Viper, A Visualisation Tool for Parallel Program Construction*. Faculty of Mathematics and Computing Science, TUE. 1999-15

K.M.M. de Leeuw. *Cryptology and Statecraft in the Dutch Republic*. Faculty of Mathematics and Computer Science, UvA. 2000-01

T.E.J. Vos. *UNITY in Diversity. A stratified approach to the verification of distributed algorithms*. Faculty of Mathematics and Computer Science, UU. 2000-02

W. Mallon. *Theories and Tools for the Design of Delay-Insensitive Communicating Processes*. Faculty of Mathematics and Natural Sciences, RUG. 2000-03

W.O.D. Griffioen. *Studies in Computer Aided Verification of Protocols*. Faculty of Science, KUN. 2000-04

P.H.F.M. Verhoeven. *The Design of the MathSpad Editor*. Faculty of Mathematics and Computing Science, TUE. 2000-05

J. Fey. *Design of a Fruit Juice Blending and Packaging Plant*. Faculty of Mechanical Engineering, TUE. 2000-06

M. Franssen. *Cocktail: A Tool for Deriving Correct Programs*. Faculty of Mathematics and Computing Science, TUE. 2000-07

P.A. Olivier. *A Framework for Debugging Heterogeneous Applications*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2000-08

E. Saaman. *Another Formal Specification Language*. Faculty of Mathematics and Natural Sciences, RUG. 2000-10

M. Jelasity. *The Shape of Evolutionary Search Discovering and Representing Search Space Structure*. Faculty of Mathematics and Natural Sciences, UL. 2001-01

R. Ahn. *Agents, Objects and Events a computational approach to knowledge, observation and communication*. Faculty of Mathematics and Computing Science, TU/e. 2001-02

M. Huisman. *Reasoning about Java programs in higher order logic using PVS and Isabelle*. Faculty of Science, KUN. 2001-03

I.M.M.J. Reymen. *Improving Design Processes through Structured Reflection*. Faculty of Mathematics and Computing Science, TU/e. 2001-04

S.C.C. Blom. *Term Graph Rewriting: syntax and semantics*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2001-05

R. van Liere. *Studies in Interactive Visualization*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2001-06

A.G. Engels. *Languages for Analysis and Testing of Event Sequences*. Faculty of Mathematics and Computing Science, TU/e. 2001-07

J. Hage. *Structural Aspects of Switching Classes*. Faculty of Mathematics and Natural Sciences, UL. 2001-08

M.H. Lamers. *Neural Networks for Analysis of Data in Environmental Epidemiology: A Case-study into Acute Effects of Air Pollution Episodes*. Faculty of Mathematics and Natural Sciences, UL. 2001-09

T.C. Ruys. *Towards Effective Model Checking*. Faculty of Computer Science, UT. 2001-10

D. Chkliaev. *Mechanical verification of concurrency control and recovery protocols*. Faculty of Mathematics and Computing Science, TU/e. 2001-11

M.D. Oostdijk. *Generation and presentation of formal mathematical documents*. Faculty of Mathematics and Computing Science, TU/e. 2001-12

A.T. Hofkamp. *Reactive machine control: A simulation approach using χ*. Faculty of Mechanical Engineering, TU/e. 2001-13

D. Bošnački. *Enhancing state space reduction techniques for model checking*. Faculty of Mathematics and Computing Science, TU/e. 2001-14

M.C. van Wezel. *Neural Networks for Intelligent Data Analysis: theoretical and experimental aspects*. Faculty of Mathematics and Natural Sciences, UL. 2002-01

V. Bos and J.J.T. Kleijn. *Formal Specification and Analysis of Industrial Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2002-02

T. Kuipers. *Techniques for Understanding Legacy Software Systems*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2002-03

S.P. Luttik. *Choice Quantification in Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-04

R.J. Willemen. *School Timetable Construction: Algorithms and Complexity*. Faculty of Mathematics and Computer Science, TU/e. 2002-05

M.I.A. Stoelinga. *Alea Jacta Est: Verification of Probabilistic, Real-time and Parametric Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-06

N. van Vugt. *Models of Molecular Computing*. Faculty of Mathematics and Natural Sciences, UL. 2002-07

A. Fehnker. *Citius, Vilius, Melius: Guiding and Cost-Optimality in Model Checking of Timed and Hybrid Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-08

R. van Stee. *On-line Scheduling and Bin Packing*. Faculty of Mathematics and Natural Sciences, UL. 2002-09

D. Tauritz. *Adaptive Information Filtering: Concepts and Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2002-10

M.B. van der Zwaag. *Models and Logics for Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-11

J.I. den Hartog. *Probabilistic Extensions of Semantical Models*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2002-12

L. Moonen. *Exploring Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-13

J.I. van Hemert. *Applying Evolutionary Computation to Constraint Satisfaction and Data Mining*. Faculty of Mathematics and Natural Sciences, UL. 2002-14

S. Andova. *Probabilistic Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2002-15

Y.S. Usenko. *Linearization in μCRL*. Faculty of Mathematics and Computer Science, TU/e. 2002-16

J.J.D. Aerts. *Random Redundant Storage for Video on Demand*. Faculty of Mathematics and Computer Science, TU/e. 2003-01

M. de Jonge. *To Reuse or To Be Reused: Techniques for component composition and construction*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-02

J.M.W. Visser. *Generic Traversal over Typed Source Code Representations*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-03

S.M. Bohte. *Spiking Neural Networks*. Faculty of Mathematics and Natural Sciences, UL. 2003-04

T.A.C. Willemse. *Semantics and Verification in Process Algebras with Data and Timing*. Faculty of Mathematics and Computer Science, TU/e. 2003-05

S.V. Nedea. *Analysis and Simulations of Catalytic Reactions*. Faculty of Mathematics and Computer Science, TU/e. 2003-06

M.E.M. Lijding. *Real-time Scheduling of Tertiary Storage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-07

H.P. Benz. *Casual Multimedia Process Annotation – CoMPAs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-08

D. Distefano. *On Modelchecking the Dynamics of Object-based Software: a Foundational Approach*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-09

M.H. ter Beek. *Team Automata – A Formal Approach to the Modeling of Collaboration Between System Components*. Faculty of Mathematics and Natural Sciences, UL. 2003-10

D.J.P. Leijen. *The λ Abroad – A Functional Approach to Software Components*. Faculty of Mathematics and Computer Science, UU. 2003-11

W.P.A.J. Michiels. *Performance Ratios for the Differencing Method*. Faculty of Mathematics and Computer Science, TU/e. 2004-01

G.I. Jojgov. *Incomplete Proofs and Terms and Their Use in Interactive Theorem Proving*. Faculty of Mathematics and Computer Science, TU/e. 2004-02

P. Frisco. *Theory of Molecular Computing – Splicing and Membrane systems*. Faculty of Mathematics and Natural Sciences, UL. 2004-03

S. Maneth. *Models of Tree Translation*. Faculty of Mathematics and Natural Sciences, UL. 2004-04

Y. Qian. *Data Synchronization and Browsing for Home Environments*. Faculty of Mathematics and Computer Science and Faculty of Industrial Design, TU/e. 2004-05

F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-06

L. Cruz-Filipe. *Constructive Real Analysis: a Type-Theoretical Formalization and Applications*. Faculty of Science, Mathematics and Computer Science, KUN. 2004-07

E.H. Gerding. *Autonomous Agents in Bargaining Games: An Evolutionary Investigation of Fundamentals, Strategies, and Business Applications*. Faculty of Technology Management, TU/e. 2004-08

N. Goga. *Control and Selection Techniques for the Automated Testing of Reactive Systems*. Faculty of Mathematics and Computer Science, TU/e. 2004-09

M. Niqui. *Formalising Exact Arithmetic: Representations, Algorithms and Proofs*. Faculty of Science, Mathematics and Computer Science, RU. 2004-10

A. Löh. *Exploring Generic Haskell*. Faculty of Mathematics and Computer Science, UU. 2004-11

I.C.M. Flinsenberg. *Route Planning Algorithms for Car Navigation*. Faculty of Mathematics and Computer Science, TU/e. 2004-12

R.J. Bril. *Real-time Scheduling for Media Processing Using Conditionally Guaranteed Budgets*. Faculty of Mathematics and Computer Science, TU/e. 2004-13

J. Pang. *Formal Verification of Distributed Systems*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-14

F. Alkemade. *Evolutionary Agent-Based Economics*. Faculty of Technology Management, TU/e. 2004-15

E.O. Dijk. *Indoor Ultrasonic Position Estimation Using a Single Base Station*. Faculty of Mathematics and Computer Science, TU/e. 2004-16

S.M. Orzan. *On Distributed Verification and Verified Distribution*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-17

M.M. Schrage. *Proxima - A Presentation-oriented Editor for Structured Documents*. Faculty of Mathematics and Computer Science, UU. 2004-18

E. Eskenazi and A. Fyukov. *Quantitative Prediction of Quality Attributes for Component-Based Software Architectures*. Faculty of Mathematics and Computer Science, TU/e. 2004-19

P.J.L. Cuijpers. *Hybrid Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2004-20

N.J.M. van den Nieuwelaar. *Supervisory Machine Control by Predictive-Reactive Scheduling*. Faculty of Mechanical Engineering, TU/e. 2004-21

E. Ábrahám. *An Assertional Proof System for Multithreaded Java -Theory and Tool Support-* . Faculty of Mathematics and Natural Sciences, UL. 2005-01

R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02

C.N. Chong. *Experiments in Rights Control - Expression and Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-03

H. Gao. *Design and Verification of Lock-free Parallel Algorithms*. Faculty of Mathematics and Computing Sciences, RUG. 2005-04

H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05

M.T. Ionita. *Scenario-Based System Architecting - A Systematic Approach to Developing Future-Proof System Architectures*. Faculty of Mathematics and Computing Sciences, TU/e. 2005-06

G. Lenzini. *Integration of Analysis Techniques in Security and Fault-Tolerance*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-07

I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, UU. 2005-09

O. Tveretina. *Decision Procedures for Equality Logic with Uninterpreted Functions*. Faculty of Mathematics and Computer Science, TU/e. 2005-10

A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14

M.R. Mousavi. *Structuring Structural Operational Semantics*. Faculty of Mathematics and Computer Science, TU/e. 2005-15

A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. Faculty of Mathematics and Computer Science, TU/e. 2005-16

T. Gelsema. *Effective Models for the Structure of pi-Calculus Processes with Replication*. Faculty of Mathematics and Natural Sciences, UL. 2005-17

P. Zoeteweij. *Composing Constraint Solvers*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-18

J.J. Vinju. *Analysis and Transformation of Source Code by Parsing and Rewriting*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-19

M.Valero Espada. *Modal Abstraction and Replication of Processes with Data*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2005-20

A. Dijkstra. *Stepping through Haskell*. Faculty of Science, UU. 2005-21

Y.W. Law. *Key management and link-layer security of wireless sensor networks: energy-efficient attack and defense*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-22

E. Dolstra. *The Purely Functional Software Deployment Model*. Faculty of Science, UU. 2006-01

R.J. Corin. *Analysis Models for Security Protocols*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-02

P.R.A. Verbaan. *The Computational Complexity of Evolving Systems*. Faculty of Science, UU. 2006-03

K.L. Man and R.R.H. Schiffelers. *Formal Specification and Analysis of Hybrid Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2006-04

M. Kyas. *Verifying OCL Specifications of UML Models: Tool Support and Compositionality*. Faculty of Mathematics and Natural Sciences, UL. 2006-05