PVS Bibliography

John Rushby
Computer Science Laboratory
SRI International
Menlo Park CA 94025 USA

Email: `pvs-request@csl.sri.com`
WWW: `http://www.csl.sri.com/pvs.html`

December 13, 1999

This bibliography lists most of the publications that I am aware of concerning the development and application of the PVS verification system and its predecessors. The files used to create this report, including the BibTeX bibliography, are available at `http://www.csl.sri.com/pvs-bib.html`. PVS users are encouraged to use the BibTeX entries from these files, which are as accurate, complete, and up to date as I can make them.

I have limited this list to publications that should be available in a university or research library. For this reason, I have omitted citations to conference proceedings that were not issued by a technical society or commercial publisher, unless they are on the Web; similarly, technical reports are cited only when they are widely distributed (e.g., NASA), or where there is no other source and they are are on the Web (or otherwise freely available, if they predate the Web). Many of the journal and conference papers listed are supported by technical reports that provide much more detail; I am willing to add these if there is demand. Because papers are often available on the web in advance of the journal or proceedings in which they will appear, I now include such papers and mark them "to appear."

Most of the publications later than about 1990, and some earlier ones, are available on the Web from `http://www.csl.sri.com/fm-papers.html` (for SRI papers) or `http://www.csl.sri.com/pvs-users.html` (for others); hard copies of all these papers are available at SRI. Many papers are archived at the NECI Scientific Literature Digital Library `http://citeseer.nj.nec.com/cs` and at The Hypatia Digital Library at `http://hypatia.dcs.qmw.ac.uk/`.

Please keep me informed of new papers (or forgotten historical ones) that should be added to this collection, and of any inaccuracies.

History

PVS is the most recent in a line of specification languages, theorem provers, and verification systems developed at SRI that goes back over 20 years. Early systems included the Jovial Verification System [1] (Jovial was a language based on Algol 58, a precursor to the more famous Algol 60, that was used by the US Air Force), and the Hierarchical Development Methodology (HDM) [2–4]. HDM had a "security analyzer" [5] based on information flow [6] that was used in the verification of the Honeywell SCOMP [7,8] (the first computer to gain the NSA's A1 [9] rating) and several other secure systems [10,11].

The HDM security flow analyzer used the Boyer-Moore theorem prover, much of whose early development was performed at SRI [12]. A parallel development was Shostak's Theorem Prover STP [13], which was based on his decision procedures [14–18] and used in the analysis of some aspects of the SIFT flight control computer [19, 20][1] and early experiments in hardware verification [23].

HDM and STP were the main design influences on "Enhanced HDM" (EHDM), whose development began in 1983. EHDM introduced the higher-order specification language with theory-level (they are called "modules" in EHDM) parameterization that provides much of the foundation for the PVS language. EHDM was operational by 1985 [24–26] and was further developed through the following decade [27, 28].[2] Verification of fault-tolerant algorithms and architectures provided the biggest applications of EHDM [29–35], and some small experiments in hardware verification [36, 37] were also performed.

PVS was started in late 1990 as an SRI-funded research program to prototype ideas for an interactive prover for EHDM and the design of a "next generation" verification system (hence its original name—"*Prototype* Verification System"). SRI made the first version of PVS freely available in 1993 [38] and it proved so popular that we have continued to develop it ever since (first with SRI IR&D funds, and later with additional support from NASA, NRL, AFOSR, ARPA, and NSF).

Documentation and Availability

PVS is freely available under license from SRI. Information, examples, papers, documentation, and the system itself are available from the PVS home page at `http://www.csl.sri.com/pvs.html` and the CSL ftp server at `ftp://ftp.csl.sri.com/pub/pvs`, and from mirror sites in England, France, Germany, and Japan.[3]

[1]Although the SIFT verification was a significant accomplishment at the time, the paper describing it [21] leaves the impression that the full operating system was verified, as opposed to some properties of its design. This led to severe criticism by a NASA peer review [22].

[2]EHDM has some capabilities currently lacking in PVS: for example, hierarchical development based on theory interpretations, imperative program verification using a small subset of a Pascal-like language, a superior prover interface (to an interactive prover derived from that of PVS) and more capable (though slower) arithmetic decision procedures. However, it lacks some of the linguistic capabilities of PVS: for example, dependent typing, abstract data types, inductive relations, and several type constructors. EHDM continues to be available, but its distribution is controlled by the US Government.

[3]These sites are `ftp://ftp.cs.york.ac.uk/pub/pvs` at the University of York, England; `ftp://ftp.ibp.fr/pub/pvs` at the University of Paris VI, France; `ftp://ftp.informatik.uni-ulm.de/pub/KI/pvs` at the University of Ulm, Germany; and `ftp://nicosia.is.s.u-tokyo.ac.jp/pub/misc/pvs` at Tokyo

The PVS system is described briefly in three system descriptions [39–41] and fully documented in separate manuals for the language [42], prover [43], system [44], and semantics [45]. Some of the motivation and design decisions for PVS are described in [46], and the utility of its type system is argued in [47]. Tutorials provide general introductions to PVS [48–50], plus more specialized treatments for hardware [51], abstract data types [52], tabular and requirements specifications [53], and distributed systems [54]. General introductions to formal methods, with PVS examples, are provided by the NASA Guidebook series [55, 56].

Methodology and Technology

Some of the philosophical and methodological ideas behind PVS and its applications are described in [57–69], while issues in the application of formal methods to safety-critical systems are described in [55, 70–72] and hardware verification is discussed in [73].

The theorem-proving techniques employed in PVS are described in [74–78], and their integration with model checking and with tabular specifications are described in [79] and [80], respectively. Experimental support for efficient evaluation and its use to support reflection in PVS are described in [81, 82]. Techniques and tools to support automated generation of abstractions and invariants in PVS are described in [83–93].

Comparisons between PVS and ACL2 treatments of an interactive consistency algorithm [94], and between PVS and HOL specifications of the IEEE 854 floating-point standard [95] are available, while Mike Gordon provides a discussion of PVS from a HOL perspective [96] and David Griffioen and Marieke Huisman compare PVS and Isabelle/HOL [97]. Jan Friso Groote, Fran cois Monin and Jaco van de Pol provide a tutorial on protocol and distributed systems verification that considers Isabelle, Coq, and PVS, and makes some comparison between the systems [98], while Kern and Greenstreet survey several methods for hardware verification, including PVS [99]. Heinrich Rust presents a PVS treatment for an "invoicing" example used in the comparison of many systems [100]. Matt Wilding describes an approach, derived from those used with Nqthm and ACL2, for developing robust proofs of machine code programs [101]. Georg Droschl compares some aspects of PVS and VDM on an industrially-motivated example [102, 103]. Merriam and Harrison provide an examination of interface issues, and compare PVS with two other systems [104], while Archer and Heitmeyer describe using PVS to produce "human-style" proofs [105, 106]. Issues in introducing formal methods, including PVS, to students and engineers are discussed by Knight et al [107].

Applications

PVS has been applied to classical topics in mathematical analysis [108], to the modeling of process calculi [109, 110] and transition systems [111, 112], to the specification of abstract datatypes [113–115], and classes in object-oriented languages [116–118], to critical algorithms for fault tolerance in automobile and aircraft control systems [119–125], to partitioning in these systems [126–128] to software requirements and software design

University, Japan.

for aircraft [129–131] and for manned [132–139] and interplanetary [140–142] spacecraft, to protection systems for nuclear power plants [143], to command and control [144], to issues in computer security [145–147], active networks [148], and cryptographic protocols [149,150], to operating systems [151], distributed algorithms [152–162], real-time [163–179], reactive [180], and hybrid systems [181–183], to embedded systems [184,185], distributed systems [186–194] and communications protocols [195–197], to software architecture [198], program development [199–201], software development steps [202–205] and refinement [206–208], to compilers [209–215], to hardware design [216–242] and synthesis [243–248], to memory models and cache coherence protocols [249–252], to multimedia collaborations [253], to testing program visualization tools [254], to validating fault-tolerant real-time systems [255–257], and to self-stabilization [258,259]. PVS has also been used to support other specification notations including the B Method [260,261], BON [262], Lustre [263], Tabular methods [264–269], VDM [270–272], and VSPEC [273], computer algebra systems [274,275], and in combination with other systems [276–278].

References

[1] B. Elspas, M. Green, M. Moriconi, and R. Shostak. A JOVIAL verifier. Technical report, Computer Science Laboratory, SRI International, January 1979.

[2] Lawrence Robinson and Karl N. Levitt. Proof techniques for hierarchically structured programs. *Communications of the ACM*, 20(4):271–283, April 1976.

[3] L. Robinson, K. N. Levitt, and B. A. Silverberg. *The HDM Handbook*. Computer Science Laboratory, SRI International, Menlo Park, CA, June 1979. Three Volumes.

[4] Jay M. Spitzen, Karl N. Levitt, and Lawrence Robinson. An example of hierarchical design and proof. *Communications of the ACM*, 21(12):1064–1075, December 1978.

[5] R. J. Feiertag. A technique for proving specifications are multilevel secure. Technical Report CSL-109, Computer Science Laboratory, SRI International, Menlo Park, CA, January 1980.

[6] R. J. Feiertag, K. N. Levitt, and L. Robinson. Proving multilevel security of a system design. In *Sixth ACM Symposium on Operating System Principles*, pages 57–65, November 1977.

[7] Terry Vickers Benzel. Analysis of a kernel verification. In *Proceedings of the Symposium on Security and Privacy*, pages 125–131, Oakland, CA, April 1984. IEEE Computer Society.

[8] J. M. Silverman. Reflections on the verification of the security of an operating system kernel. In *Ninth ACM Symposium on Operating System Principles*, pages

143–154, Bretton Woods, NH, October 1983. (ACM Operating Systems Review, Vol. 17, No. 5).

[9] Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985. DOD 5200.28-STD (supersedes CSC-STD-001-83).

[10] Kenneth S. Lindsay. A taxonomy of the causes of proof failures in applications using the HDM methodology. In *Fourth Aerospace Computer Security Applications Conference*, pages 419–423, Orlando, FL, December 1988. IEEE Computer Society. Reprinted in [279, pp. 79–83].

[11] Bill Smith, Cynthia Reese, Kenneth S. Lindsay, and Brian Crane. A description of a formal verification and validation (FVV) process. In *Fourth Aerospace Computer Security Applications Conference*, pages 401–408, Orlando, FL, December 1988. IEEE Computer Society. Reprinted in [279, pp. 71–78].

[12] R. S. Boyer and J S. Moore. *A Computational Logic.* Academic Press, New York, NY, 1979.

[13] R. E. Shostak, R. Schwartz, and P. M. Melliar-Smith. STP: A mechanized logic for specification and verification. In D. Loveland, editor, *6th International Conference on Automated Deduction (CADE)*, volume 138 of *Lecture Notes in Computer Science*, New York, NY, 1982. Springer-Verlag.

[14] Robert E. Shostak. On the SUP-INF method for proving Presburger formulas. *Journal of the ACM*, 24(4):529–543, October 1977.

[15] Robert E. Shostak. An algorithm for reasoning about equality. *Communications of the ACM*, 21(7):583–585, July 1978.

[16] Robert E. Shostak. A practical decision procedure for arithmetic with function symbols. *Journal of the ACM*, 26(2):351–360, April 1979.

[17] Robert E. Shostak. Deciding linear inequalities by computing loop residues. *Journal of the ACM*, 28(4):769–779, October 1981.

[18] Robert E. Shostak. Deciding combinations of theories. *Journal of the ACM*, 31(1):1–12, January 1984.

[19] Charles B. Weinstock. SIFT: System design and implementation. In *Fault Tolerant Computing Symposium 10*, pages 75–77, Kyoto, Japan, October 1980. IEEE Computer Society. Reprinted in [280, pp. 29–30].

[20] John H. Wensley, Leslie Lamport, Jack Goldberg, Milton W. Green, Karl N. Levitt, P. M. Melliar-Smith, Robert E. Shostak, and Charles B. Weinstock. SIFT: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, 66(10):1240–1255, October 1978.

[21] P. M. Melliar-Smith and R. L. Schwartz. Formal specification and verification of SIFT: A fault-tolerant flight control system. *IEEE Transactions on Computers*, C-31(7):616–630, July 1982.

[22] NASA Conference Publication 2377. *Peer Review of a Formal Verification/Design Proof Methodology*, July 1983.

[23] Robert E. Shostak. Formal verification of circuit designs. In T. Uehara and M. Barbacci, editors, *Computer Hardware Description Languages*, pages 13–29. North-Holland, 1983.

[24] Richard A. Kemmerer. Verification assessment study final report. Technical Report C3-CR01-86, National Computer Security Center, Ft. Meade, MD, 1986. 5 Volumes (Overview, Gypsy, Affirm, FDM, and EHDM). US distribution only.

[25] P. Michael Melliar-Smith and John Rushby. The Enhanced HDM system for specification and verification. In *Proc. VerkShop III*, pages 41–43, Watsonville, CA, February 1985. Published as ACM Software Engineering Notes, Vol. 10, No. 4, Aug. 85.

[26] John Rushby. The security model of Enhanced HDM. In *Proceedings 7th DoD/NBS Computer Security Initiative Conference*, pages 120–136, Gaithersburg, MD, September 1984.

[27] F. W. von Henke, J. S. Crow, R. Lee, J. M. Rushby, and R. A. Whitehurst. The EHDM verification environment: An overview. In *Proceedings 11th National Computer Security Conference*, pages 147–155, Baltimore, MD, October 1988. NBS/NCSC.

[28] John Rushby, Friedrich von Henke, and Sam Owre. An introduction to formal specification and verification using EHDM. Technical Report SRI-CSL-91-2, Computer Science Laboratory, SRI International, Menlo Park, CA, February 1991.

[29] Ben L. Di Vito and Ricky W. Butler. Formal techniques for synchronized fault-tolerant systems. In C. E. Landwehr, B. Randell, and L. Simoncini, editors, *Dependable Computing for Critical Applications—3*, volume 8 of *Dependable Computing and Fault-Tolerant Systems*, pages 163–188. Springer-Verlag, Vienna, Austria, September 1992.

[30] Paul S. Miner. Verification of fault-tolerant clock synchronization systems. NASA Technical Paper 3349, NASA Langley Research Center, Hampton, VA, November 1993.

[31] John Rushby. Formal verification of an Oral Messages algorithm for interactive consistency. Technical Report SRI-CSL-92-1, Computer Science Laboratory, SRI International, Menlo Park, CA, July 1992. Also available as NASA Contractor Report 189704, October 1992.

[32] John Rushby. A fault-masking and transient-recovery model for digital flight-control systems. In Jan Vytopil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Kluwer International Series in Engineering and Computer Science, chapter 5, pages 109–136. Kluwer, Boston, Dordecht, London, 1993.

[33] John Rushby. A formally verified algorithm for clock synchronization under a hybrid fault model. In *Thirteenth ACM Symposium on Principles of Distributed Computing*, pages 304–313, Los Angeles, CA, August 1994. Association for Computing Machinery. Also available as NASA Contractor Report 198289.

[34] John Rushby and Friedrich von Henke. Formal verification of algorithms for critical systems. *IEEE Transactions on Software Engineering*, 19(1):13–23, January 1993.

[35] Natarajan Shankar. Mechanical verification of a generalized protocol for Byzantine fault-tolerant clock synchronization. In J. Vytopil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 571 of *Lecture Notes in Computer Science*, pages 217–236, Nijmegen, The Netherlands, January 1992. Springer-Verlag.

[36] Ricky W. Butler and Jon A. Sjogren. Formal design verification of digital circuitry. *Reliability Engineering and System Safety*, 32:67–93, 1991.

[37] J. Joyce, E. Liu, J. Rushby, N. Shankar, R. Suaya, and F. von Henke. From formal verification to silicon compilation. In *IEEE Compcon*, pages 450–455, San Francisco, CA, February 1991.

[38] N. Shankar, S. Owre, and J. M. Rushby. *PVS Tutorial.* Computer Science Laboratory, SRI International, Menlo Park, CA, February 1993. Also appears in Tutorial Notes, *Formal Methods Europe '93: Industrial-Strength Formal Methods*, pages 357–406, Odense, Denmark, April 1993.

[39] S. Owre, S. Rajan, J.M. Rushby, N. Shankar, and M.K. Srivas. PVS: Combining specification, proof checking, and model checking. In Alur and Henzinger [293], pages 411–414.

[40] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, June 1992. Springer-Verlag.

[41] N. Shankar. PVS: Combining specification, proof checking, and model checking. In Srivas and Camilleri [292], pages 257–264.

[42] S. Owre, N. Shankar, J. M. Rushby, and D. W. J. Stringer-Calvert. *PVS Language Reference.* Computer Science Laboratory, SRI International, Menlo Park, CA, September 1999.

[43] N. Shankar, S. Owre, J. M. Rushby, and D. W. J. Stringer-Calvert. *PVS Prover Guide*. Computer Science Laboratory, SRI International, Menlo Park, CA, September 1999.

[44] S. Owre, N. Shankar, J. M. Rushby, and D. W. J. Stringer-Calvert. *PVS System Guide*. Computer Science Laboratory, SRI International, Menlo Park, CA, September 1999.

[45] Sam Owre and Natarajan Shankar. The formal semantics of PVS. Technical Report SRI-CSL-97-2, Computer Science Laboratory, SRI International, Menlo Park, CA, August 1997.

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

[47] John Rushby, Sam Owre, and N. Shankar. Subtypes for specifications: Predicate subtyping in PVS. *IEEE Transactions on Software Engineering*, 24(9):709–720, September 1998.

[48] Ricky W. Butler. An elementary tutorial on formal specification and verification using PVS 2. NASA Technical Memorandum 108991, NASA Langley Research Center, Hampton, VA, June 1993. Revised June 1995. Available, with PVS specification files, at `http://atb-www.larc.nasa.gov/ftp/larc/PVS-tutorial`; use only files marked "revised.".

[49] Judy Crow, Sam Owre, John Rushby, Natarajan Shankar, and Mandayam Srivas. A tutorial introduction to PVS. Presented at WIFT '95: Workshop on Industrial-Strength Formal Specification Techniques, Boca Raton, Florida, April 1995. Available, with specification files, at `http://www.csl.sri.com/wift-tutorial.html`.

[50] John Rushby and David W. J. Stringer-Calvert. A less elementary tutorial for the PVS specification and verification system. Technical Report SRI-CSL-95-10, Computer Science Laboratory, SRI International, Menlo Park, CA, June 1995. Revised, July 1996. Available, with specification files, at `http://www.csl.sri.com/csl-95-10.html`.

[51] S. Owre, J. M. Rushby, N. Shankar, and M. K. Srivas. A tutorial on using PVS for hardware verification. In Kumar and Kropf [286], pages 258–279.

[52] Sam Owre and Natarajan Shankar. Abstract datatypes in PVS. Technical Report SRI-CSL-93-9R, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1993. Extensively revised June 1997; Also available as NASA Contractor Report CR-97-206264.

[53] Sam Owre, John Rushby, and Natarajan Shankar. Analyzing tabular and state-transition specifications in PVS. Technical Report SRI-CSL-95-12, Computer Science Laboratory, SRI International, Menlo Park, CA, July 1995. Available, with

specification files, at http://www.csl.sri.com/csl-95-12.html. Also published as NASA Contractor Report 201729.

[54] John Rushby. Specification, proof checking, and model checking for protocols and distributed systems with PVS. Tutorial presented at FORTE X/PSTV XVII '97: Formal Description Techniques and Protocol Specification, Testing and Verification, November 1997. Available, with specification files, at http://www.csl.sri.com/forte97.html.

[55] Judith Crow et al. *NASA Formal Methods Specification and Verification Guidebook for Software and Computer Systems, Volume I: Planning and Technology Insertion*. NASA Office of Safety and Mission Assurance, Washington, DC, 1995. Available at http://eis.jpl.nasa.gov/quality/Formal_Methods/.

[56] Judith Crow et al. *NASA Formal Methods Specification and Verification Guidebook for Software and Computer Systems, Volume II: A Practitioner's Companion*. NASA Office of Safety and Mission Assurance, Washington, DC, 1997. Available at http://eis.jpl.nasa.gov/quality/Formal_Methods/.

[57] David A. Cyrluk and Mandayam K. Srivas. Theorem proving: Not an esoteric diversion, but the unifying framework for industrial verification. In *International Conference on Computer Design: VLSI in Computers and Processors (ICCD '95)*, pages 538–544, Austin, TX, October 1995. IEEE Computer Society.

[58] David L. Dill and John Rushby. Acceptance of formal methods: Lessons from hardware design. *IEEE Computer*, 29(4):23–24, April 1996. Part of [281].

[59] John Rushby. Formal methods for dependable real-time systems. In *International Symposium on Real-Time Embedded Processing for Space Applications*, pages 355–366, Les Saintes-Maries-de-la-Mer, France, November 1992. CNES, the French Space Agency. Published by Cépaduès-Éditions, Toulouse, France.

[60] John Rushby. Mechanizing formal methods: Opportunities and challenges. In Jonathan P. Bowen and Michael G. Hinchey, editors, *ZUM '95: The Z Formal Specification Notation; 9th International Conference of Z Users*, volume 967 of *Lecture Notes in Computer Science*, pages 105–113, Limerick, Ireland, September 1995. Springer-Verlag.

[61] John Rushby. Automated deduction and formal methods. In Alur and Henzinger [293], pages 169–183.

[62] John Rushby. Mechanized formal methods: Progress and prospects. In V. Chandru and V. Vinay, editors, *16th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1180 of *Lecture Notes in Computer Science*, pages 43–51, Hyderabad, India, December 1996. Springer-Verlag.

[63] John Rushby. Calculating with requirements. In RE-3 [294], pages 144–146.

[64] John Rushby. Integrated formal verification: Using model checking with automated abstraction, invariant generation, and theorem proving. In D. Dams, R. Gerth, S. Leue, and M. Massink, editors, *Theoretical and Practical Aspects of SPIN Model Checking: 5th and 6th International SPIN Workshops*, volume 1680 of *Lecture Notes in Computer Science*, pages 1–11, Trento, Italy, and Toulouse, France, July and September 1999. Springer-Verlag.

[65] John Rushby. Mechanized formal methods: Where next? In Wing and Woodcock [284], pages 48–51. Pages 1–938 are in the first volume, 939–1872 in the second.

[66] N. Shankar. Towards mechanical metamathematics. *Journal of Automated Reasoning*, 1(4):407–434, 1985.

[67] N. Shankar. Observations on the use of computers in proof checking. *Notices of the American Mathematical Society*, 35(6):804–805, July/August 1988.

[68] N. Shankar. *Metamathematics, Machines, and Gödel's Proof.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, UK, 1994.

[69] Natarajan Shankar. Unifying verification paradigms. In Jonsson and Parrow [291], pages 22–39.

[70] John Rushby. Formal methods and the certification of critical systems. Technical Report SRI-CSL-93-7, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1993. Also issued under the title *Formal Methods and Digital Systems Validation for Airborne Systems* as NASA Contractor Report 4551, December 1993. A book based on this material will be published by Cambridge University Press in 1998/9.

[71] John Rushby. Critical system properties: Survey and taxonomy. *Reliability Engineering and System Safety*, 43(2):189–219, 1994.

[72] John Rushby. Formal methods and their role in the certification of critical systems. Technical Report SRI-CSL-95-1, Computer Science Laboratory, SRI International, Menlo Park, CA, March 1995. Also available as NASA Contractor Report 4673, August 1995, and issued as part of the *FAA Digital Systems Validation Handbook* (the guide for aircraft certification). Reprinted in [282, pp. 1–42].

[73] Mandayam Srivas, Harald Rueß, and David Cyrluk. Hardware verification using PVS. In Thomas Kropf, editor, *Formal Hardware Verification: Methods and Systems in Comparison*, volume 1287 of *Lecture Notes in Computer Science*, pages 156–205. Springer-Verlag, 1997.

[74] David Cyrluk, Patrick Lincoln, and N. Shankar. On Shostak's decision procedure for combinations of theories. In M. A. McRobbie and J. K. Slaney, editors, *Automated Deduction—CADE-13*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 463–477, New Brunswick, NJ, July/August 1996. Springer-Verlag.

[75] David Cyrluk, Harald Rueß, and Oliver Möller. An efficient decision procedure for the theory of fixed-sized bit-vectors. In Grumberg [295], pages 60–71.

[76] D. Cyrluk and P. Narendran. Ground temporal logic—a logic for hardware verification. In David Dill, editor, *Computer-Aided Verification, CAV '94*, volume 818 of *Lecture Notes in Computer Science*, pages 247–259, Stanford, CA, June 1994. Springer-Verlag.

[77] D. Cyrluk, S. Rajan, N. Shankar, and M. K. Srivas. Effective theorem proving for hardware verification. In Kumar and Kropf [286], pages 203–222.

[78] Oliver Möller and Harald Rueß. Solving bit-vector equations. In Gopalakrishnan and Windley [306], pages 36–48.

[79] S. Rajan, N. Shankar, and M.K. Srivas. An integration of model-checking with automated proof checking. In Pierre Wolper, editor, *Computer-Aided Verification, CAV '95*, volume 939 of *Lecture Notes in Computer Science*, pages 84–97, Liege, Belgium, June 1995. Springer-Verlag.

[80] Sam Owre, John Rushby, and N. Shankar. Integration in PVS: Tables, types, and model checking. In Ed Brinksma, editor, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '97)*, volume 1217 of *Lecture Notes in Computer Science*, pages 366–383, Enschede, The Netherlands, April 1997. Springer-Verlag.

[81] Friedrich von Henke, Stephan Pfab, Harald Rueß, and Sam Owre. Towards lightweight verification and heavy-weight testing. In Berghammer and Lakhnech [305], pages 189–200. Proceedings published in May 1999.

[82] Friedrich von Henke, Stephan Pfab, Holger Pfeifer, and Harald Rueß. Case studies in meta-level theorem proving. In Grundy and Newey [303], pages 461–478.

[83] Saddek Bensalem, Yassine Lakhnech, and Hassen Saïdi. Powerful techniques for the automatic generation of invariants. In Alur and Henzinger [293], pages 323–335.

[84] Saddek Bensalem, Yassine Lakhnech, and Sam Owre. Computing abstractions of infinite state systems compositionally and automatically. In Hu and Vardi [304], pages 319–331.

[85] Saddek Bensalem, Yassine Lakhnech, and Sam Owre. InVeSt: A tool for the verification of invariants. In Hu and Vardi [304], pages 505–510.

[86] Parosh Aziz Abdulla, Aurore Annichini, Saddek Bensalem, Ahmed Bouajjani, Peter Habermehl, and Yassine Lakhnech. Verification of infinite-state systems by combining abstraction and reachability analysis. In Halbwachs and Peled [310], pages 146–159.

[87] Vlad Rusu and Eli Singerman. On proving safety properties by integrating static analysis, theorem proving and abstraction. In W. Rance Cleaveland, editor, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '99)*, volume 1579 of *Lecture Notes in Computer Science*, pages 178–192, Amsterdam, The Netherlands, March 1999. Springer-Verlag.

[88] Susanne Graf and Hassen Saïdi. Verifying invariants using theorem proving. In Alur and Henzinger [293], pages 196–207.

[89] Hassen Saïdi. A tool for proving invariance properties of concurrent systems automatically. In *Tools and Algorithms for the Construction and Analysis of Systems TACAS '96*, volume 1055 of *Lecture Notes in Computer Science*, pages 412–416, Passau, Germany, March 1996. Springer-Verlag.

[90] Hassen Saïdi. The Invariant Checker: Automated deductive verification of reactive systems. In Grumberg [295], pages 436–439.

[91] Hassen Saïdi and Susanne Graf. Construction of abstract state graphs with PVS. In Grumberg [295], pages 72–83.

[92] Hassen Saïdi. Automated deductive verification of parallel systems. In Holloway and Hayhurst [299], pages 163–176. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[93] Hassen Saïdi and N. Shankar. Abstract and model check while you prove. In Halbwachs and Peled [310], pages 443–454.

[94] William D. Young. Comparing verification systems: Interactive Consistency in ACL2. *IEEE Transactions on Software Engineering*, 23(4):214–223, April 1997.

[95] Victor A. Carreño and Paul S. Miner. Specification of the IEEE-854 floating-point standard in HOL and PVS. In *HOL95: Eighth International Workshop on Higher-Order Logic Theorem Proving and Its Applications*, Aspen Grove, UT, September 1995. Category B proceedings, available at `http://lal.cs.byu.edu/lal/hol95/Bprocs/indexB.html`.

[96] Mike Gordon. Notes on PVS from a HOL perspective. Available at `http://www.cl.cam.ac.uk/users/mjcg/PVS.html`, August 1995.

[97] David Griffioen and Marieke Huisman. A comparison of PVS and Isabelle/HOL. In Grundy and Newey [303], pages 123–142.

[98] Jan Friso Groote, cois Monin Fran and Jaco van de Pol. Checking verifications of protocols and distributed systems by computer. In Davide Sangiorgi and Robert de Simone, editors, *CONCUR'98: Concurrency Theory*, volume 1466 of *Lecture Notes in Computer Science*, pages 629–655, Nice, France, September 1998. Springer-Verlag.

[99] Christoph Kern and Mark R. Greenstreet. Formal verification in hardware design: A survey. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 4(2):123–193, April 1999.

[100] Heinrich Rust. A PVS specification of an invoicing system. In M. Allemand, C. Attiogbé, and H. Habrias, editors, *Proceedings of an International Workshop on Specification Techniques and Formal Methods*, pages 51–65. University of Nantes, France, March 1998. Available at `http://www.informatik.tu-cottbus.de/~rust/publications/a_pvs_specificat%ion_of_an_invoicing_system.ps.gz`.

[101] Matthew Wilding. Robust computer system proofs in PVS. In Holloway and Hayhurst [299], pages 177–184. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[102] Georg Droschl. On the integration of formal methods: Events and scenarios in PVS and VDM. In *Proceedings, 3rd. Irish Workshop in Formal Methods*, Galway, Ireland, July 1999. Electronic Workshops in Computing (`http://ewic.org.uk/ewic/`).

[103] Georg Droschl. Analyzing the requirements of an access control using VDMTools and PVS. In Wing and Woodcock [284], page 1807. Pages 1–938 are in the first volume, 939–1872 in the second.

[104] Nicholas A. Merriam and Michael D. Harrison. Evaluating the interfaces of three theorem proving assistants. In F. Bodart and J. Vanderdonckt, editors, *Proceedings of the 3rd International Eurographics Workshop on Design, Specification, and Verification of Interactive Systems*, 1996.

[105] Myla Archer and Constance Heitmeyer. Human-style theorem proving using PVS. In Gunter and Felty [296], pages 33–48.

[106] Myla Archer, Constance Heitmeyer, and Steve Sims. TAME: A PVS interface to simplify proofs for automata models. In *User Interfaces for Theorem Provers*, Eindhoven, The Netherlands, July 1998. Informal proceedings available at `http://www.win.tue.nl/cs/ipa/uitp/proceedings.html`.

[107] John Knight, Colleen DeJong, Matthew Gibble, and Luís Nakano. Why are formal methods not used more widely? In Holloway and Hayhurst [299], pages 1–12. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[108] Bruno Dutertre. Elements of mathematical analysis in PVS. In von Wright et al. [290], pages 141–156.

[109] Twan Basten and Jozef Hooman. Process algebra in PVS. In W. Rance Cleaveland, editor, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '99)*, volume 1579 of *Lecture Notes in Computer Science*, pages 270–284, Amsterdam, The Netherlands, March 1999. Springer-Verlag.

[110] Dmitri Schamschurko. Modeling process calculi with PVS. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *First Workshop on Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 197–214, Lisbon, Portugal, March 1998. Elsevier. Available at `http://www.elsevier.nl/cas/tree/store/tcs/free/entcs/store/tcs11/tcs110%11.ps`.

[111] David Griffioen and Frits Vaandrager. Normed simulations. In Hu and Vardi [304], pages 332–344.

[112] Amir Pnueli, N. Shankar, and Eli Singerman. Fair synchronous transition systems and their liveness proofs. In FTRTFT [302], pages 198–209.

[113] D. Griffioen M. Devillers and and O. Mueller. Possibly infinite sequences in theorem provers: A comparative study. In Gunter and Felty [296], pages 89–104.

[114] Ulrich Hensel and Bart Jacobs. Proof principles for datatypes with iterated recursion. In Eugenio Moggi and Giuseppe Rosolini, editors, *Category Theory and Computer Science*, volume 1290 of *Lecture Notes in Computer Science*, pages 220–241, Santa Margherita Ligure, Italy, September 1997. Springer-Verlag.

[115] Ulrich Hensel and Bart Jacobs. Coalgebraic theories of sequences in PVS. *Journal of Logic and Computation*, 1998. To appear; available at `http://www.cs.kun.nl/~bart/PAPERS/JLC.ps.Z`.

[116] Ulrich Hensel, Marieke Huisman, Bart Jacobs, and Hendrik Tews. Reasoning about classes in object-oriented languages: Logical models and tools. In Chris Hankin, editor, *Programming Languages and Systems: 7th European Symposium On Programming (ESOP)*, volume 1381 of *Lecture Notes in Computer Science*, pages 105–121, Lisbon, Portugal, March 1998. Springer-Verlag.

[117] Joachim van den Berg Marieke Huisman, Bart Jacobs. A case study in class library verification: Java's vector class. In *Object-Oriented Technology: ECOOP'99 Workshop Reader*, volume 1743 of *Lecture Notes in Computer Science*, Lisbon, Portugal, June 1999. Springer-Verlag. To appear.

[118] Bart Jacobs, Joachim van den Berg, Marieke Huisman, Martijn van Berkum, Ulrich Hensel, and Hendrick Tews. Reasoning about Java classes. In *Proceedings, Object-Oriented Programming Systems, Languages and Applications (OOPSLA'98)*, pages 329–340, Vancouver, Canada, October 1998. Association for Computing Machinery. Proceedings issued as ACM SIGPLAN Notices Vol. 33, No. 10, October 1998.

[119] Patrick Lincoln and John Rushby. Formal verification of an algorithm for interactive consistency under a hybrid fault model. In Courcoubetis [285], pages 292–304.

[120] Patrick Lincoln and John Rushby. A formally verified algorithm for interactive consistency under a hybrid fault model. In *Fault Tolerant Computing Symposium 23*, pages 402–411, Toulouse, France, June 1993. IEEE Computer Society. Reprinted in [280, pp. 438–447].

[121] Patrick Lincoln and John Rushby. Formal verification of an interactive consistency algorithm for the Draper FTP architecture under a hybrid fault model. In *COMPASS '94 (Proceedings of the Ninth Annual Conference on Computer Assurance)*, pages 107–120, Gaithersburg, MD, June 1994. IEEE Washington Section.

[122] Holger Pfeifer, Detlef Schwier, and Friedrich W. von Henke. Formal verification for time-triggered clock synchronization. In Weinstock and Rushby [308], pages 207–226.

[123] John Rushby. Systematic formal verification for fault-tolerant time-triggered algorithms. In Mario Dal Cin, Catherine Meadows, and William H. Sanders, editors, *Dependable Computing for Critical Applications—6*, volume 11 of *Dependable Computing and Fault Tolerant Systems*, pages 203–222, Garmisch-Partenkirchen, Germany, March 1997. IEEE Computer Society. To appear in IEEE Transactions on Software Engineering.

[124] D. Schwier and F. von Henke. Mechanical verification of clock synchronization algorithms. In FTRTFT [302], pages 262–271.

[125] Chris J. Walter, Patrick Lincoln, and Neeraj Suri. Formally verified on-line diagnosis. *IEEE Transactions on Software Engineering*, 23(11):684–721, November 1997.

[126] Ben L. Di Vito. A formal model of partitioning for integrated modular avionics. NASA Contractor Report CR-1998-208703, NASA Langley Research Center, August 1998.

[127] Ben L. Di Vito. A model of cooperative noninterference for integrated modular avionics. In Weinstock and Rushby [308], pages 269–286.

[128] Matthew M. Wilding, David S. Hardin, and David A. Greve. Invariant performance: A statement of task isolation useful for embedded application integration. In Weinstock and Rushby [308], pages 287–300.

[129] Ricky W. Butler. An introduction to requirements capture using PVS: Specification of a simple autopilot. NASA Technical Memorandum 110255, NASA Langley Research Center, Hampton, VA, May 1996.

[130] Bruno Dutertre and Victoria Stavridou. Formal requirements analysis of an avionics control system. *IEEE Transactions on Software Engineering*, 23(5):267–278, May 1997.

[131] Bruno Dutertre and Victoria Stavridou. Requirements analysis of real-time control systems using PVS. In Holloway and Hayhurst [299], pages 65–74. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[132] Judith Crow and Ben L. Di Vito. Formalizing Space Shuttle software requirements: Four case studies. *ACM Transactions on Software Engineering and Methodology*, 7(3):296–332, July 1998.

[133] Steve Easterbrook, Robyn Lutz, Richard Covington, John Kelly, Yoko Ampo, and David Hamilton. Experiences using lightweight formal methods for requirements modeling. *IEEE Transactions on Software Engineering*, 24(1):4–14, January 1998.

[134] Larry W. Roberts and Mike Beims. Using formal methods to assist in the requirements analysis of the Space Shuttle HAC Change Request (CR 90960E). Technical Report JSC-27599, NASA Johnson Space Center, Houston, TX, September 1996.

[135] Ben L. Di Vito. Formalizing new navigation requirements for NASA's Space Shuttle. In FME'96 [289], pages 160–178.

[136] Ben L. Di Vito. High-automation proofs for properties of requirements models. *Software Tools for Technology Transfer*, 2, 1999. To appear.

[137] Ben L. Di Vito and Larry W. Roberts. Using formal methods to assist in the requirements analysis of the Space Shuttle GPS Change Request. NASA Contractor Report 4752, NASA Langley Research Center, August 1996.

[138] David Hamilton, Rick Covington, and John Kelly. Experiences in applying formal methods to the analysis of software and system requirements. In *WIFT '95: Workshop on Industrial-Strength Formal Specification Techniques*, pages 30–43, Boca Raton, FL, 1995. IEEE Computer Society.

[139] David Hamilton, Rick Covington, and Alice Lee. An experience report on requirements reliability engineering using formal methods. In *Sixth International Conference on Software Reliability Engineering, ISSRE '95*, pages 52–57, Toulouse, France, 1995. IEEE Computer Society.

[140] Yoko Ampo and Robyn Lutz. Evaluation of software safety analysis using formal methods. In *Workshop for Foundation of Software Engineering (FOSE)*, Hamana-Ko, Japan, December 1995. In Japanese.

[141] Robyn Lutz and Yoko Ampo. Experience report: Using formal methods for requirements analysis of critical spacecraft software. In *Proceedings of the 19th Annual Software Engineering Workshop*, pages 231–248, Greenbelt, MD, December 1994. NASA Goddard Space Flight Center.

[142] Robyn Lutz. Reuse of a formal model for requirements validation. In Holloway and Hayhurst [299], pages 75–86. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[143] S. Koo, H. Son, and P. Seong. Mathematical verification of a nuclear power plant protection system function with combined CPN and PVS. *Journal of the Korean Nuclear Society*, 31:157–171, April 1999.

[144] Jaco van de Pol, Jozef Hooman, and Edwin de Jong. Formal requirements specification for command and control systems. In *Engineering of Computer Based Systems (ECBS)*, pages 37–44, Jerusalem, Israel, March–April 1998. IEEE Computer Society. Available at `http://www.win.tue.nl/cs/tt/hooman/ECBS98.html`.

[145] Drew Dean. Static typing with dynamic linking. In *Fourth ACM Conference on Computer and Communications Security*, pages 18–27, Zurich, Switzerland, April 1997. Association for Computing Machinery.

[146] John Hoffman and Charlie Payne. A formal experience at Secure Computing Corporation. In Hu and Vardi [304], pages 49–56.

[147] Ramesh V. Peri, William A. Wulf, and Darrell M. Kienzle. A logic of composition for information flow predicates. In *9th Computer Security Foundations Workshop*, pages 82–94, Kenmare, Ireland, June 1996. IEEE Computer Society.

[148] Darryl Dieckman, Perry Alexander, and Philip A. Wilsey. ActiveSPEC: A framework for the specification and verification of active network services and security policies. In Nevin Heintze and Jeannette Wing, editors, *Workshop on Formal Methods and Security Protocols*, Indianapolis, IN, June 1998. Informal proceedings available at `http://www.cs.bell-labs.com/who/nch/fmsp/program.html`.

[149] Bruno Dutertre and Steve Schneider. Embedding CSP in PVS. an application to authentication protocols. In Gunter and Felty [296], pages 121–136.

[150] Jonathan K. Millen. A necessarily parallel attack. In Nevin Heintze and Edmund Clarke, editors, *Workshop on Formal Methods and Security Protocols, Part of the Federated Logic Conference*, Trento, Italy, July 1999.

[151] N.S. Pendharkar and K. Gopinath. Formal verification of an O.S. submodule. In V. Arvind and R. Ramanujin, editors, *18th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1530 of *Lecture Notes in Computer Science*, pages 197–208, Madras, India, December 1998. Springer-Verlag.

[152] Raphaël Couturier and Dominique Méry. An experiment in parallelizing an application using formal methods. In Hu and Vardi [304], pages 345–356.

[153] Dmitri Chkliaev, Jozef Hooman, and Peter van der Stok. Mechanical verification of a non-blocking atomic commitment protocol. In *submitted*, 2000.

[154] Dmitri Chkliaev, Jozef Hooman, and Peter van der Stok. Serializability preserving extensions of concurrency control protocols. In *Third Andrei Ershov International Conference Perspectives of System Informatics*, volume 1755 of *Lecture Notes in Computer Science*, Novosibirsk, Russia, July 1999. Springer-Verlag. To appear.

[155] Raphaël Couturier. Formal engineering of the bitonic sort using PVS. In *Proceedings, 2nd. Irish Workshop in Formal Methods*, Cork, Ireland, July 1998. Electronic Workshops in Computing (`http://ewic.org.uk/ewic/`).

[156] Marcelo Glusman and Shmuel Katz. Mechanizing proofs of computation equivalence. In Halbwachs and Peled [310], pages 354–367.

[157] Klaus Havelund. Mechanical verification of a garbage collector. In Rolim et al. [309], pages 1258–1283. Presented at the Workshop on Formal Methods for Parallel Programming: Theory and Applications (FMPPTA).

[158] Jozef Hooman. Formal verification of the binary exponential backoff protocol. *Proceedings of the Estonian Academy of Sciences (Special issue on the 9th Nordic Workshop on Programming Theory)*, 4(2):89–105, 1998. Also available at `http://www.win.tue.nl/cs/tt/hooman/BEB.html`.

[159] Paul Jackson. Verifying a garbage collection algorithm. In Grundy and Newey [303], pages 225–244.

[160] Abdel Mokkedem, Ravi Hosabettu, and Ganesh Gopalakrishnan. Formalization and proof of a solution to the PCI 2.1 bus transaction ordering problem. In Gopalakrishnan and Windley [306], pages 237–254.

[161] John Penix, Dale Martin, Peter Frey, Radharamanan Radhakrishnan, Perry Alexander, and Phillip A. Wilsey. Experiences in verifying parallel simulation algorithms. In Mark Ardis, editor, *Second Workshop on Formal Methods in Software Practice (FMSP '98)*, pages 16–23, Clearwater Beach, FL, March 1998. Association for Computing Machinery.

[162] Kothanda Umamageswaran, Krishnan Subramani, Philip A. Wilsey, and Perry Alexander. Formal verification and empirical analysis of rollback relaxation. *Journal of Systems Architecture (formerly published as Microprocessing and Microprogramming: the Euromicro Journal)*, 44(6–7):473–495, March 1998.

[163] Andren Alborghetti, Angelo Gargantini, and Angelo Morzenti. Providing automated support to deductive analysis of time critical systems. In Mehdi Jazayeri and Helmut Schauer, editors, *Software Engineering—ESEC/FSE '97: Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, volume 1301 of *Lecture Notes in Computer Science*, pages 211–226, Zurich, Switzerland, September 1997. Springer-Verlag.

[164] Myla Archer and Constance Heitmeyer. Mechanical verification of timed automata: A case study. In *IEEE Real-Time Technology and Applications Symposium (RTAS'96)*, pages 192–203, Brookline, MA, June 1996. IEEE Computer Society.

[165] M. Archer and C. Heitmeyer. TAME: A specialized specification and verification system for timed automata. In Azer Bestavros, editor, *Work In Progress (WIP) Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS'96)*, pages 3–6, Washington, DC, December 1996. The WIP Proceedings is available at `http://www.cs.bu.edu/pub/ieee-rts/rtss96/wip/proceedings`.

[166] Leon Bun. Checking properties of ASTRAL specifications with PVS. In *Proceedings of 2nd Annual Conference of the Advanced School for Computing and Imaging (ASCI'96)*, pages 102–107, Lommel, Belgium, June 1996. Available at `http://sepc.twi.tudelft.nl/~leonbun/papers/asci96.ps`.

[167] Leon Bun. Embedding Astral in PVS. In *Proceedings of 3rd Annual Conference of the Advanced School for Computing and Imaging (ASCI'97)*, pages 130–136, Lommel, Belgium, June 1997. Available at `http://sepc.twi.tudelft.nl/~leonbun/papers/asci97.ps`.

[168] F. S. de Boer, H. Tej, W.-P. de Roever, and M. van Hulst. Compositionality in real-time shared variable concurrency. In Jonsson and Parrow [291], pages 420–439.

[169] Philippe Du Bois, Eric Dubois, and Jean-Marc Zeippen. On the use of a formal RE language: The generalized railroad crossing problem. In RE-3 [294], pages 128–137.

[170] Simon Fowler and Andy Wellings. Formal analysis of a real-time kernel specification. In Jonsson and Parrow [291], pages 440–458.

[171] Simon Fowler and Andy Wellings. Formal development of a real-time kernel. In *Real Time Systems Symposium*, pages 220–229, San Francisco, CA, December 1997. IEEE Computer Society.

[172] Jozef Hooman. Correctness of real time systems by construction. In Langmaack et al. [287], pages 19–40.

[173] Jozef Hooman. Using PVS for an assertional verification of the RPC-Memory specification problem. In Manfred Broy, Stephan Merz, and Katharina Spies, editors, *Formal Systems Specification: The RPC-Memory Specification Case Study*, volume 1169 of *Lecture Notes in Computer Science*, pages 275–304. Springer-Verlag, 1996.

[174] Jozef Hooman. Verification of distributed real-time and fault-tolerant protocols. In Johnson [300], pages 261–275.

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

[176] Paul Z. Kolano. Proof assistance for real-time systems using an interactive theorem prover. In J.-P. Katoen, editor, *Formal Methods for Real-Time and Probabilistic Systems (Proceedings 5th International AMAST Workshop, ARTS'99)*, volume 1601 of *Lecture Notes in Computer Science*, pages 315–333, Bamberg, Germany, May 1999. Springer-Verlag.

[177] Natarajan Shankar. Verification of real-time systems using PVS. In Courcoubetis [285], pages 280–291.

[178] Jens Ulrik Skakkebæk. *A Verification Assistant for a Real-Time Logic*. PhD thesis, Department of Computer Science, Technical University of Denmark, Lyngby, Denmark, November 1994.

[179] Jens U. Skakkebæk and N. Shankar. Towards a Duration Calculus proof assistant in PVS. In Langmaack et al. [287], pages 660–679.

[180] Pertti Kellomäki. Verification of reactive systems using DisCo and PVS. In FME [297], pages 589–604.

[181] M. Archer and C. Heitmeyer. Verifying hybrid systems modeled as timed automata: A case study. In Oded Maler, editor, *Proceedings of the International Workshop on Hybrid and Real-Time Systems (HART'97)*, volume 1201 of *Lecture Notes in Computer Science*, pages 171–185, Grenoble, France, March 1997. Springer-Verlag.

[182] Thomas A. Henzinger and Vlad Rusu. Reachability verification for hybrid automata. In Thomas A. Henzinger and Shankar Sastry, editors, *Hybrid Systems: Computation and Control (First International Workshop, HSCC'98)*, volume 1386 of *Lecture Notes in Computer Science*, pages 190–204, Berkeley, CA, April 1998. Springer-Verlag.

[183] Jan Vitt and Jozef Hooman. Assertional specification and verification using PVS of the steam boiler control system. In Jean-Raymond Abrial, Egon Boerger, and Hans Langmaack, editors, *Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control*, volume 1165 of *Lecture Notes in Computer Science*, pages 453–472. Springer-Verlag, 1996.

[184] Roel Bloo, Jozef Hooman, and Edwin de Jong. Semantical aspects of an architecture for distributed embedded systems. In *Proceedings of the 2000 ACM Symposium on Applied Computing (SAC 2000)*, Como, Italy, March 2000. Association for Computing Machinery. To appear.

[185] Jaco van de Pol, Jozef Hooman, and Edwin de Jong. Modular formal specification of data and behaviour. In K. Araki, A. Galloway, and K. Taguchi, editors, *Proceedings 1st Conference on Integrated Formal Methods (IFM'99)*, pages 109–128, York, UK, June 1999. Springer.

[186] Frank S. de Boer and Marten van Hulst. Local nondeterminism in asynchronously communicating processes. In FME'96 [289], pages 367–384.

[187] Victoria Chernyakhovsky, Peter Frey, Radharamanan Radhakrihnan, Philip A. Wilsey, Perry Alexander, and Harold W. Carter. A formal framework for specifying and verifying time warp optimizations. In Rolim et al. [309], pages 1228–1242. Presented at the Workshop on Formal Methods for Parallel Programming: Theory and Applications (FMPPTA).

[188] Peter Frey, Radharamanan Radhakrishnan, Philip A. Wilsey, Perry Alexander, and Harold W. Carter. An extensible formal framework for the specification and verification of an optimistic simulation protocol. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, January 1999. Available at `http://computer.org/conferen/proceed/hicss/0001/00013/00013049abs.htm`.

[189] Jozef Hooman. Developing proof rules for distributed real-time systems with PVS. In Buth et al. [307], pages 120–139. Proceedings of a workshop held at Bremen in 1996.

[190] Pertti Kellomäki. Mechanizing invariant proofs of joint action systems. In *Fourth Symposium on Programming Languages and Software Tools*, pages 141–152, Visegrad, Hungary, June 1995. Available at `http://www.cs.tut.fi/~pk/papers/visegrad-95/visegrad.ps.gz`.

[191] Pertti Kellomäki. Using auxiliary knowledge in automating invariant proofs. In *International Conference on Theorem Proving in Higher Order Logics*, pages 57–68, Turku, Finland, August 1996. Supplementary proceedings, available at `http://www.tucs.abo.fi/publications/general/G1.html`.

[192] David Lesens and Hassen Saïdi. Automatic verification of parameterized networks of processes by abstraction. In Faron Moller, editor, *2nd International Workshop on Verification of Infinite State Systems: Infinity '97*, volume 9 of *Electronic Notes in Theoretical Computer Science*, Bologna, Italy, July 1997. Elsevier. Available at `http://www.elsevier.nl/cas/tree/store/tcs/free/entcs/store/tcs9/tcs9009%.ps`.

[193] N. Shankar. A lazy approach to compositional verification. Technical Report SRI-CSL-93-8, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1993.

[194] Marten van Hulst. *Compositional Verification of Parallel Programs using Epistemic Logic and Abstract Assertional Languages*. PhD thesis, Faculteit Wiskunde en Informatica, Universiteit Utrecht, The Netherlands, June 1995.

[195] Jozef Hooman. Verifying part of the ACCESS.bus protocol using PVS. In P. S. Thiagarajan, editor, *15th Conference on the Foundations of Software Technology*

and Theoretical Computer Science, volume 1026 of *Lecture Notes in Computer Science*, pages 96–110, Bangalore, India, December 1995. Springer-Verlag.

[196] Klaus Havelund and N. Shankar. Experiments in theorem proving and model checking for protocol verification. In FME'96 [289], pages 662–681.

[197] Lars Khüne, Jozef Hooman, and Willem-Paul de Roever. Towards mechanical verification of parts of the IEEE P1394 serial bus. In *2nd International Workshop on Applied Formal Methods in System Design*, Zagreb, Croatia, June 1997. Available at `http://www.win.tue.nl/cs/tt/hooman/P1394.html`.

[198] John Herbert, Bruno Dutertre, Robert Riemenschneider, and Victoria Stavridou. A formalization of software architecture. In Wing and Woodcock [284], pages 116–133. Pages 1–938 are in the first volume, 939–1872 in the second.

[199] Jozef Hooman. Program design in PVS. In K.Berghammer, J.Peleska, and B. Buth, editors, *Workshop on Tool Support for System Development and Verification*, Bremen, Germany, 1997. Available at `http://www.win.tue.nl/cs/tt/hooman/PDPVS.html`.

[200] Michel Lévy and Laurent Trilling. A PVS-based approach for teaching constructing correct iterations. In Wing and Woodcock [284], pages 1859–1860. Pages 1–938 are in the first volume, 939–1872 in the second.

[201] Richard Verhoeven and Roland Backhouse. Interfacing program construction and verification. In Wing and Woodcock [284], pages 1128–1146. Pages 1–938 are in the first volume, 939–1872 in the second.

[202] Ralf Behnke, Rudolf Berghammer, and Sönke Magnussen. Supporting algebraic program derivation by PVS. In Buth et al. [307], pages 22–40. Proceedings of a workshop held at Bremen in 1996.

[203] Axel Dold. Representing, verifying and applying software development steps using the PVS system. In V. S. Alagar and Maurice Nivat, editors, *Algebraic Methodology and Software Technology, AMAST'95*, volume 936 of *Lecture Notes in Computer Science*, pages 431–445, Montreal, Canada, July 1995. Springer-Verlag.

[204] N. Shankar. Computer-aided computing. In Bernhard Möller, editor, *Mathematics of Program Construction '95*, volume 947 of *Lecture Notes in Computer Science*, pages 50–66. Springer-Verlag, 1995.

[205] N. Shankar. Steps towards mechanizing program transformations using PVS. *Science of Computer Programming*, 26(1–3):33–57, 1996.

[206] Bart Jacobs. Behaviour-refinement of coalgebraic specifications with coinductive correctness proofs. In Michel Bidoit and Max Dauchet, editors, *TAPSOFT '97: Theory and Practice of Software Development*, volume 1214 of *Lecture Notes in Computer Science*, pages 787–802, Lille, France, April 1997. Springer-Verlag.

[207] Bart Jacobs. Invariants, bisimulations and the correctness of coalgebraic refinements. In Johnson [300], pages 276–291.

[208] Jens Knappmann. A PVS based tool for developing programs in the Refinement Calculus. Master's thesis, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel, Kiel, Germany, October 1996.

[209] Karl-Heinz Buth. Automated code generator verification based on algebraic laws. Procos II Report Kiel KHB 5/1, Christian-Albrechts Universität zu Kiel, Kiel, Germany, September 1995. Available at `ftp://ftp.informatik.uni-kiel.de/pub/kiel/procos/kiel-khb-5-1.ps.Z`.

[210] A. Dold, F. W. von Henke, H. Pfeifer, and H. Rueß. Formal verification of transformations for peephole optimiztion. In FME [297], pages 459–472.

[211] A. Dold, T. Gaul, V. Vialard, and W. Zimmermann. ASM-Based Mechanized Verification of Compiler Backends. In Uwe Glässer and Peter H. Schmitt, editors, *Proceedings of the 5th International Workshop on Abstract State Machines*, pages 50–67, Magdeburg, Germany, September 1998. Available at `http://i44www.info.uni-karlsruhe.de/~verifix/pres/paper/ASM-WS98-DGVZ.p%s.gz`.

[212] Axel Dold, Thilo Gaul, and Wolf Zimmermann. Mechanized verification of compiler backends. In Tiziana Margaria and Bernhard Steffen, editors, *Proceedings of the International Workshop on Software Tools for Technology Transfer STTT'98*, pages 13–22, Aalborg, Denmark, June 1998. BRICS NS-98-4. Proceedings available at `http://www.brics.dk/NS/98/4/`.

[213] David W. J. Stringer-Calvert, Susan Stepney, and Ian Wand. Using PVS to prove a Z refinement: A case study. In FME [297], pages 573–588.

[214] David W. J. Stringer-Calvert. *Mechanical Verification of Compiler Correctness*. PhD thesis, University of York, Department of Computer Science, York, England, March 1998. Available at `http://www.csl.sri.com/~dave_sc/papers/thesis.html`.

[215] M. Wahab. Verification and abstraction of flow-graph programs with pointers and computed jumps. Research Report CS-RR-354, Department of Computer Science, University of Warwick, Coventry, UK, November 1998. Available at `http://www.dcs.warwick.ac.uk/pub/reports/rr/354.html`.

[216] Tamara Arons and Amir Pnueli. Verifying Tomasulo's algorithm by refinement. In *The Twelfth International Conference on VLSI Design*, pages 306–309, Goa, India, January 1999. IEEE Computer Society.

[217] David Cyrluk. Inverting the abstraction mapping: A methodology for hardware verification. In Srivas and Camilleri [292], pages 172–186.

[218] David Cyrluk, John Rushby, and Mandayam Srivas. Systematic formal verification of interpreters. In Hinchey and Liu [298], pages 140–149.

[219] Werner Damm and Amir Pnueli. Verifying out-of-order executions. In li and Probst [301], pages 23–47.

[220] Kathi Fisler. Extending formal reasoning with support for hardware diagrams. In Kumar and Kropf [286], pages 298–303.

[221] Ganesh Gopalakrishnan, Rajnish Ghughal, Ravi Hosabettu, Abdelillah Mokkedem, and Ratan Nalumasu. Formal modeling and validation applied to a commercial coherent bus: A case study. In li and Probst [301], pages 48–62.

[222] David Greve. Symbolic simulation of the JEM1 microprocessor. In Gopalakrishnan and Windley [306], pages 321–333.

[223] David Hardin, Matthew Wilding, and David Greve. Transforming the theorem prover into a digital design tool: From concept car to off-road vehicle. In Hu and Vardi [304], pages 39–44.

[224] Ravi Hosabettu, Mandayam Srivas, and Ganesh Gopalakrishnan. Decomposing the proof of correctness of pipelined microprocessors. In Hu and Vardi [304], pages 122–134.

[225] Ravi Hosabettu, Mandayam Srivas, and Ganesh Gopalakrishnan. Proof of correctness of a processor with reorder buffer using the completion functions approach. In Halbwachs and Peled [310], pages 47–59.

[226] Ravi Hosabettu, Ganesh Gopalakrishnan, and Mandayam Srivas. Proof of correctness of a processor implementing Tomasulo's algorithm without a reorder buffer. In *Advances in Hardware Design and Verification: IFIP WG10.5 International Conference on Correct Hardware Design and Verification Methods (CHARME '99)*, volume 1703 of *Lecture Notes in Computer Science*, pages 8–22, Bad Herrenalb, Germany, September 1999. Springer-Verlag.

[227] Steven D. Johnson, Paul S. Miner, and Albert Camlleri. Studies of the single-pulser in various reasoning systems. In Kumar and Kropf [286], pages 126–145.

[228] Steven D. Johnson and Paul S. Miner. Integrated reasoning support in system design: Design derivation and theorem proving. In li and Probst [301], pages 255–272.

[229] James Leathrum, Jr. Fundamental hardware design in PVS. In Holloway and Hayhurst [299], pages 193–204. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[230] Xiaoshan Li, Antonio Cau, Ben Moszkowski, Nick Coleman, and Hussein Zedan. Proving the correctness of the interlock mechanism in processor design. In li and Probst [301], pages 5–22.

[231] Paul S. Miner and Steven D. Johnson. Verification of an optimized fault-tolerant clock synchronization circuit: A case study exploring the boundary between formal reasoning systems. In Mary Sheeran and Satnam Singh, editors, *Designing Correct Circuits*, Bastad, Sweden, September 1996. Electronic Workshops in Computing (`http://ewic.org.uk/ewic/`).

[232] Paul S. Miner, Shyamsundar Pullela, and Steven D. Johnson. Interaction of formal design systems in the development of a fault-tolerant clock synchronization circuit. In *13th Symposium on Reliable Distributed Systems*, pages 128–137, Dana Point, CA, October 1994. IEEE Computer Society.

[233] Paul S. Miner and James F. Leathrum, Jr. Verification of IEEE compliant subtractive division algorithms. In Srivas and Camilleri [292], pages 64–78.

[234] Steven P. Miller and Mandayam Srivas. Formal verification of the AAMP5 microprocessor: A case study in the industrial use of formal methods. In *WIFT '95: Workshop on Industrial-Strength Formal Specification Techniques*, pages 2–16, Boca Raton, FL, 1995. IEEE Computer Society.

[235] Amir Pnueli and Tamara Arons. Verification of data-insensitive circuits: An in-order-retirement case study. In Gopalakrishnan and Windley [306], pages 351–368.

[236] H. Rueß, N. Shankar, and M. K. Srivas. Modular verification of SRT division. In Alur and Henzinger [293], pages 123–134.

[237] H. Rueß, N. Shankar, and M. K. Srivas. Modular verification of SRT division. *Formal Methods in Systems Design*, 14(1):45–73, January 1999.

[238] Sreeranga P. Rajan and Masahiro Fujita. ATM switch design: Parametric high-level modeling and formal verification. In Johnson [300], pages 437–450.

[239] S. P. Rajan, M. Fujita, K. Yuan, and M. T-C. Lee. ATM switch design by high level modeling, formal verification, and high level synthesis. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 3(4):554–562, October 1998.

[240] Mandayam K. Srivas and Steven P. Miller. Applying formal verification to a commercial microprocessor. In Johnson [288], pages 493–502.

[241] Mandayam K. Srivas and Steven P. Miller. Formal verification of the AAMP5 microprocessor. In Michael G. Hinchey and Jonathan P. Bowen, editors, *Applications of Formal Methods*, Prentice Hall International Series in Computer Science, chapter 7, pages 125–180. Prentice Hall, Hemel Hempstead, UK, 1995.

[242] Mandayam K. Srivas and Steven P. Miller. Applying formal verification to the AAMP5 microprocessor: A case study in the industrial use of formal methods. *Formal Methods in Systems Design*, 8(2):153–188, March 1996.

[243] Nazanin Mansouri and Ranga Vemuri. A methodology for automatic verification of synthesized RTL designs and its integration with a high-level synthesis tool. In Gopalakrishnan and Windley [306], pages 204–221.

[244] Naren Narasimhan, Elena Teica, Rajesh Radhakrishnan, Sriram Govindarajan, and Ranga Vemuri. Theorem proving guided development of formal assertions in a resource-constrained scheduler for high-level synthesis. In Andreas Kuehlmann, editor, *International Conference on Computer Design (ICCD'98)*, Austin, TX, October 1998. IEEE Computer Society.

[245] Naren Narasimhan and Ranga Vemuri. On the effectiveness of theorem proving guided discovery of formal assertions for a register allocator in a high-level synthesis system. In Grundy and Newey [303], pages 367–386.

[246] P. Sreeranga Rajan. Transformations in high level synthesis: Specification and verification. Technical Report NL-TN 118/94, Philips Research Laboratories, Eindhoven, The Netherlands, April 1994. Revised version available as SRI Technical Report SRI-CSL-94-10, October 1994.

[247] Sreeranga P. Rajan. Correctness of transformations in high level synthesis. In Johnson [288], pages 597–603.

[248] Sreeranga P. Rajan. *Transformations on Dependency Graphs: Formal Specification and Efficient Mechanical Verification*. PhD thesis, Department of Computer Science, University of British Columbia, Vancouver, Canada, October 1995.

[249] Ratan Nalumasu and Ganesh Gopalakrishnan. Deriving efficient cache coherence protocols through refinement. In *Formal Methods for Parallel Programming: Theory and Applications (FMPPTA)*, Orlando, FL, March 98. Available at `http://www.loria.fr/~mery/lncs_fmppta98/paper7.ps`.

[250] Seungjoon Park. *Computer Assisted Analysis of Multiprocessor Memory Systems*. PhD thesis, Department of Electrical Engineering, Stanford University, June 1996.

[251] Seungjoon Park and David L. Dill. An executable specification, analyzer and verifier for relaxed memory order: With formal proofs. Submitted for publication; an earlier version (without proofs) is available [283], 1996.

[252] Seungjoon Park and David L. Dill. Verification of cache coherence protocols by aggregation of distributed transactions. *Theory of Computing Systems*, 31(4):355–376, 1998.

[253] Sreeranga Rajan, P. Venkat Rangan, and Harrick M. Vin. A formal basis for structured multimedia collaborations. In *Proceedings of the 2nd IEEE International Conference on Multimedia Computing and Systems*, pages 194–201, Washington, DC, May 1995. IEEE Computer Society.

[254] T. C. Nicholas Graham. A method for the formal testing of program visualization tools. In *Proceedings of the Fourth Workshop on Program Comprehension*, pages 45–54, Berlin, Germany, March 1996. IEEE Computer Society.

[255] Neeraj Suri and Purnendu Sinha. On the use of formal methods for validation. In *Fault Tolerant Computing Symposium 28*, pages 390–399, Munich, Germany, June 1998. IEEE Computer Society.

[256] Purnendu Sinha and Neeraj Suri. Identification of test cases using a formal fault injection approach. In *Fault Tolerant Computing Symposium 29*, pages 314–321, Madison, WI, June 1999. IEEE Computer Society.

[257] Purnendu Sinha and Neeraj Suri. Formal techniques for dependable real time protocols. In *Real Time Systems Symposium*, Phoenix, AZ, December 1999. IEEE Computer Society. To appear.

[258] Shaz Qadeer and Natarajan Shankar. Verifying a self-stabilizing mutual exclusion algorithm. In David Gries and Willem-Paul de Roever, editors, *IFIP International Conference on Programming Concepts and Methods (PROCOMET '98)*, pages 424–443, Shelter Island, NY, June 1998. Chapman & Hall.

[259] Sandeep Kulkarni, John Rushby, and N. Shankar. A case study in component-based mechanical verification of fault-tolerant programs. In *ICDCS Workshop on Self-Stabilizing Systems*, pages 33–40, Austin, TX, June 1999. IEEE Computer Society.

[260] Jean Paul Bodeveix, Mamoun Filali, and César A. Muñoz. A formalization of the B-Method in Coq and PVS. In *FM99: The World Congress in Formal Methods, User Group Meeting 2. The B-Method: Applying B in an Industrial Context: Tools, Lessons and Techniques*, September 1999. Available on the CD-ROM for [284].

[261] César Muñoz and John Rushby. Structural embeddings: Mechanization with method. In Wing and Woodcock [284], pages 452–471. Pages 1–938 are in the first volume, 939–1872 in the second.

[262] Richard F. Paige and Jonathan S. Ostroff. Developing BON as an industrial-strength formal method. In Wing and Woodcock [284], pages 834–853. Pages 1–938 are in the first volume, 939–1872 in the second.

[263] S. Bensalem, P. Caspi, C. Parent-Vigouroux, and C. Dumas. A methodology for proving control systems with Lustre and PVS. In Weinstock and Rushby [308], pages 89–107.

[264] John Rushby and Mandayam Srivas. Using PVS to prove some theorems of David Parnas. In Jeffrey J. Joyce and Carl-Johan H. Seger, editors, *Higher Order Logic Theorem Proving and its Applications (6th International Workshop, HUG '93)*,

volume 780 of *Lecture Notes in Computer Science*, pages 163–173, Vancouver, Canada, August 1993. Springer-Verlag.

[265] Mats P. E. Heimdahl and Barbara J. Czerny. Using PVS to analyze hierarchical state-based requirements for completeness and consistency. In *IEEE High-Assurance Systems Engineering Workshop (HASE '96)*, pages 252–262, Niagara on the Lake, Canada, October 1996.

[266] Mats P.E. Heimdahl. Verifying communication related safety constraints in RSML specifications. In Holloway and Hayhurst [299], pages 115–128. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[267] Mats P. E. Heimdahl and Jeffrey M. Thompson. Specification and analysis of system level inter-component communication. In Hinchey and Liu [298], pages 192–201.

[268] Barbara J. Czerny and Mats P. E. Heimdahl. Automated integrative analysis of state-based requirements. In D. Redmiles and B. Nuseibeh, editors, *Thirteenth IEEE Conference on Automated Software Engineering (ASE '98)*, Nonolulu, Hawaii, October 1998. IEEE Computer Society.

[269] Mats P. E. Heimdahl, Jeffrey M. Thompson, and Barbara J. Czerny. Specification and analysis of intercomponent communication. *IEEE Computer*, 31(4):47–54, April 1998.

[270] Sten Agerholm. Translating specifications in VDM-SL to PVS. In von Wright et al. [290], pages 1–16.

[271] Sten Agerholm, Juan Bicarregui, and Savi Maharaj. On the verification of VDM specification and refinement with PVS. In Juan Bicarregui, editor, *Proof in VDM: Case Studies*, FACIT (Formal Approaches to Computing and Information Technology), chapter 6, pages 157–190. Springer-Verlag, London, UK, 1997.

[272] Savi Maharaj and Juan Bicarregui. On the verification of VDM specification and refinement with PVS. In M. Lowry and Y. Ledru, editors, *12th IEEE International Conference on Automated Software Engineering (ASE '97)*, pages 280–289, Incline Village, NV, November 1997. IEEE Computer Society.

[273] Perry Alexander, Murali Rangarajan, and Phillip Baraona. A brief summary of VSPEC. In Wing and Woodcock [284], pages 1068–1086. Pages 1–938 are in the first volume, 939–1872 in the second.

[274] A. A. Adams, H. Gottliebsen, S. A. Linton, and U. Martin. VSDITLU: A verifiable symbolic definite integral table look-up. In *Automated Deduction—CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 112–126, Trento, Italy, July 1999. Springer-Verlag.

[275] A. A. Adams, H. Gottliebsen, S. A. Linton, and U. Martin. Automated theorem proving in support of computer algebra: Symbolic definite integration as a case study. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation, (ISSAC '99)*, pages 253–260, Vancouver, B.C., Canada, July 1999. Association for Computing Machinery.

[276] Bettina Buth, Rachel Cardell-Oliver, and Jan Peleska. Combining tools for the verification of fault-tolerant systems. In Buth et al. [307], pages 41–69. Proceedings of a workshop held at Bremen in 1996.

[277] Bettina Buth. PAMELA + PVS. In Berghammer and Lakhnech [305], pages 62–76. Proceedings published in May 1999.

[278] Bettina Buth. PAMELA + PVS. In Johnson [300], pages 560–562.

[279] IEEE Washington Section. *COMPASS '89 (Proceedings of the Fourth Annual Conference on Computer Assurance)*, Gaithersburg, MD, June 1989.

[280] IEEE Computer Society. *Fault Tolerant Computing Symposium 25: Highlights from 25 Years*, Pasadena, CA, June 1995.

[281] Hossein Saiedian (Ed.). An invitation to formal methods. *IEEE Computer*, 29(4):16–30, April 1996. A "roundtable" of short articles by several authors.

[282] Roger Shaw, editor. *Safety and Reliability of Software Based Systems (Twelfth Annual CSR Workshop)*, Bruges, Belgium, September 1995. Springer.

[283] Seungjoon Park and David L. Dill. An executable specification, analyzer and verifier for RMO (Relaxed Memory Order). In *7th ACM Symposium on Parallel Algorithms and Architectures*, pages 34–51, July 1995.

[284] Jeannette Wing and Jim Woodcock, editors. *FM99: The World Congress in Formal Methods*, volume 1708 and 1709 of *Lecture Notes in Computer Science*, Toulouse, France, September 1999. Springer-Verlag. Pages 1–938 are in the first volume, 939–1872 in the second.

[285] Costas Courcoubetis, editor. *Computer-Aided Verification, CAV '93*, volume 697 of *Lecture Notes in Computer Science*, Elounda, Greece, June/July 1993. Springer-Verlag.

[286] Ramayya Kumar and Thomas Kropf, editors. *Theorem Provers in Circuit Design (TPCD '94)*, volume 910 of *Lecture Notes in Computer Science*, Bad Herrenalb, Germany, September 1994. Springer-Verlag.

[287] H. Langmaack, W.-P. de Roever, and J. Vytopil, editors. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, Lübeck, Germany, September 1994. Springer-Verlag.

[288] Steven D. Johnson, editor. *CHDL '95: 12th Conference on Computer Hardware Description Languages and their Applications*, Chiba, Japan, August 1995. Proceedings published in a single volume jointly with ASP-DAC '95, CHDL '95, and VLSI '95, IEEE Catalog no. 95TH8102.

[289] *Formal Methods Europe FME '96*, volume 1051 of *Lecture Notes in Computer Science*, Oxford, UK, March 1996. Springer-Verlag.

[290] Joakim von Wright, Jim Grundy, and John Harrison, editors. *Theorem Proving in Higher Order Logics: 9th International Conference, TPHOLs '96*, volume 1125 of *Lecture Notes in Computer Science*, Turku, Finland, August 1996. Springer-Verlag.

[291] Bengt Jonsson and Joachim Parrow, editors. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1135 of *Lecture Notes in Computer Science*, Uppsala, Sweden, September 1996. Springer-Verlag.

[292] Mandayam Srivas and Albert Camilleri, editors. *Formal Methods in Computer-Aided Design (FMCAD '96)*, volume 1166 of *Lecture Notes in Computer Science*, Palo Alto, CA, November 1996. Springer-Verlag.

[293] Rajeev Alur and Thomas A. Henzinger, editors. *Computer-Aided Verification, CAV '96*, volume 1102 of *Lecture Notes in Computer Science*, New Brunswick, NJ, July/August 1996. Springer-Verlag.

[294] IEEE Computer Society. *3rd IEEE International Symposium on Requirements Engineering*, Annapolis, MD, January 1997.

[295] Orna Grumberg, editor. *Computer-Aided Verification, CAV '97*, volume 1254 of *Lecture Notes in Computer Science*, Haifa, Israel, June 1997. Springer-Verlag.

[296] Elsa Gunter and Amy Felty, editors. *Theorem Proving in Higher Order Logics: 10th International Conference, TPHOLs '97*, volume 1275 of *Lecture Notes in Computer Science*, Murray Hill, NJ, August 1997. Springer-Verlag.

[297] *Formal Methods Europe FME '97*, volume 1313 of *Lecture Notes in Computer Science*, Graz, Austria, September 1997. Springer-Verlag.

[298] Michael G. Hinchey and Shaoying Liu, editors. *First International Conference on Formal Engineering Methods (ICFEM '97)*, Hiroshima, Japan, November 1997. IEEE Computer Society.

[299] C. Michael Holloway and Kelly J. Hayhurst, editors. *LFM' 97: Fourth NASA Langley Formal Methods Workshop*, NASA Conference Publication 3356, Hampton, VA, September 1997. NASA Langley Research Center. Available at `http://atb-www.larc.nasa.gov/Lfm97/proceedings/`.

[300] Michael Johnson, editor. *Algebraic Methodology and Software Technology, AMAST'97*, volume 1349 of *Lecture Notes in Computer Science*, Sydney, Australia, December 1997. Springer-Verlag.

[301] Hon F. li and David K. Probst, editors. *Advances in Hardware Design and Verification: IFIP WG10.5 International Conference on Correct Hardware Design and Verification Methods (CHARME)*, Montreal, Canada, October 1997. Chapman & Hall.

[302] *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1486 of *Lecture Notes in Computer Science*, Lyngby, Denmark, September 1998. Springer-Verlag.

[303] Jim Grundy and Malcolm Newey, editors. *Theorem Proving in Higher Order Logics: 11th International Conference, TPHOLs '98*, volume 1479 of *Lecture Notes in Computer Science*, Canberra, Australia, September 1998. Springer-Verlag.

[304] Alan J. Hu and Moshe Y. Vardi, editors. *Computer-Aided Verification, CAV '98*, volume 1427 of *Lecture Notes in Computer Science*, Vancouver, Canada, June 1998. Springer-Verlag.

[305] Rudolph Berghammer and Yassine Lakhnech, editors. *Tool Support for System Specification, Development, and Verification*, Advances in Computing Science, Malente, Germany, June 1998. Springer-Verlag. Proceedings published in May 1999.

[306] Ganesh Gopalakrishnan and Phillip Windley, editors. *Formal Methods in Computer-Aided Design (FMCAD '98)*, volume 1522 of *Lecture Notes in Computer Science*, Palo Alto, CA, November 1998. Springer-Verlag.

[307] Bettina Buth, Rudolf Berghammer, and Jan Peleska, editors. *Tools for System Development and Verification*, BISS Monographs, Aachen, Germany, 1998. Shaker Verlag. Proceedings of a workshop held at Bremen in 1996.

[308] Charles B. Weinstock and John Rushby, editors. *Dependable Computing for Critical Applications—7*, volume 12 of *Dependable Computing and Fault Tolerant Systems*, San Jose, CA, January 1999. IEEE Computer Society.

[309] José Rolim et al., editors. *Parallel and Distributed Processing (Combined Proceedings of 11 Workshops)*, volume 1586 of *Lecture Notes in Computer Science*, San Juan, Puerto Rico, April 1999. Springer-Verlag.

[310] Nicolas Halbwachs and Doron Peled, editors. *Computer-Aided Verification, CAV '99*, volume 1633 of *Lecture Notes in Computer Science*, Trento, Italy, July 1999. Springer-Verlag.