Three decades of success stories in formal methods

Hubert Garavel

INRIA / LIG

with contributions of Susanne Graf

CNRS / Verimag

Context of this work

- A study on formal methods launched in 2011 by the German BSI (federal office on security in information technology)

- Currently, formal methods are not so widely used – mostly in niche applications

- Can we prove to (industrial or governmental) managers that formal methods are useful?

A frequent answer: failure stories

- List of well-known software failures:
  - Therac 25 radiation therapy engine
  - Denver airport
  - Patriot missile interceptor
  - Pentium 5 division algorithm
  - Ariane 5.01 maiden flight
  - Mars orbiter
  - etc.
- But:
  - these are disaster stories
  - they threaten the audience, but do not prove that formal methods would have enabled to avoid such failures

Be positive!

- Try to convince rather than threaten
- Find success stories, not disaster stories

Related studies (1)

There already exist formal methods surveys:

- Craigen, Gerhart & Ralston 1993, 1994, 1995
  - 12 case studies
  - old and unbalanced: SCR (formulas and tables), B, CAS logic, Gypsy (1st order predicate calculus), Z, Z, Cleanroom, Z, RAISE, Z+Occam, RAISE, VDM
  - nothing about protocol or hardware verification
- Clarke & Wing 1996
  - 22 pages
  - rich and dense
  - quite exhaustive and fair, but 16 years-old now

Related studies (2)

- Woodcock, Larsen, Bicarregui & Fitzgerald 2009
  - 40 pages
  - rich and deep
  - but a bit biased:
    - towards "US+UK" and "paper and pencil" methods
    - against model checking (admitted)
    - (not a word about the PRISM model checker!)
  - 8 featured projects (Section 4): Z+CSP, B, Z, RAISE+PVS+ACL2, SCADE, Z+SPIN, SPARK, VDM++
- Haxthausen 2010
  - 32 pages, a bit superficial, not balanced
  - 6 case studies: B, B, B, RAISE, SPIN, Z

Related studies (3)

- Dagstuhl 2010 Manifesto
  - comprehensive panorama
  - no case studies
- www.fm4industry.org
  - famous formal methods success stories
  - 8 case studies listed (1980-2009)
- www.fmsurvey.org
  - survey on industrial use of formal methods (2011)
  - 62 projects considered
  - 34% agree and 61% strongly agree that using formal methods was successful

A different methodology

- How to ensure:
  - a larger coverage over years?
  - a greater diversity of approaches and methods?
  - a better balance between countries?

- Idea:
  - Review 3 decades (from 1982 to 2011 included)
  - Select one "success story" each year

  => 30 featured case studies

Selection criteria

- Focus on concrete **applications** of formal methods, rather than theoretical discoveries or software tool releases
- Many applications took several years: the year of first publication is chosen, rather the dates of start or end of the project (often unknown or unclear)
- Avoid selecting the same approach twice

Difficulties (1)

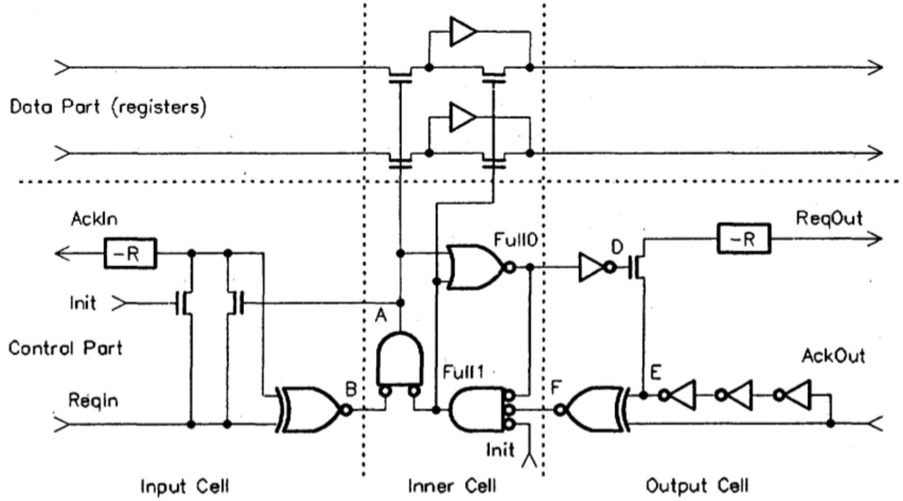
- Exhaustivity is impossible:
  - Only 30 slots for a large formal methods community
  - Google Scholar: 220,000 answers for "formal methods"
  - Nearly 4000 scientists in verification [Woodcock et al]
    - USA : 1000 scientists in verification [Shankar 2009]
    - Europe : 1000
    - Nordic countries : 500
    - China : 250
    - Japan : 250
    - Australia / New Zealand / Brazil / Canada / Singapore / South Africa : 1000

Difficulties (2)

- The problem (finding one single "success story" per year) is perhaps over-constrained
- In recent years, there are several convincing applications of formal methods. No one is really the "best"
- Priority to case studies that were influential and/or reproduced later independently
- Some "tweaking" needed on publication dates (conference vs journal, etc.)

1982

[Bochmann]
[Fujita-Tanaka-Moto Oka]
[Clarke-Mishra-Browne-Dill]

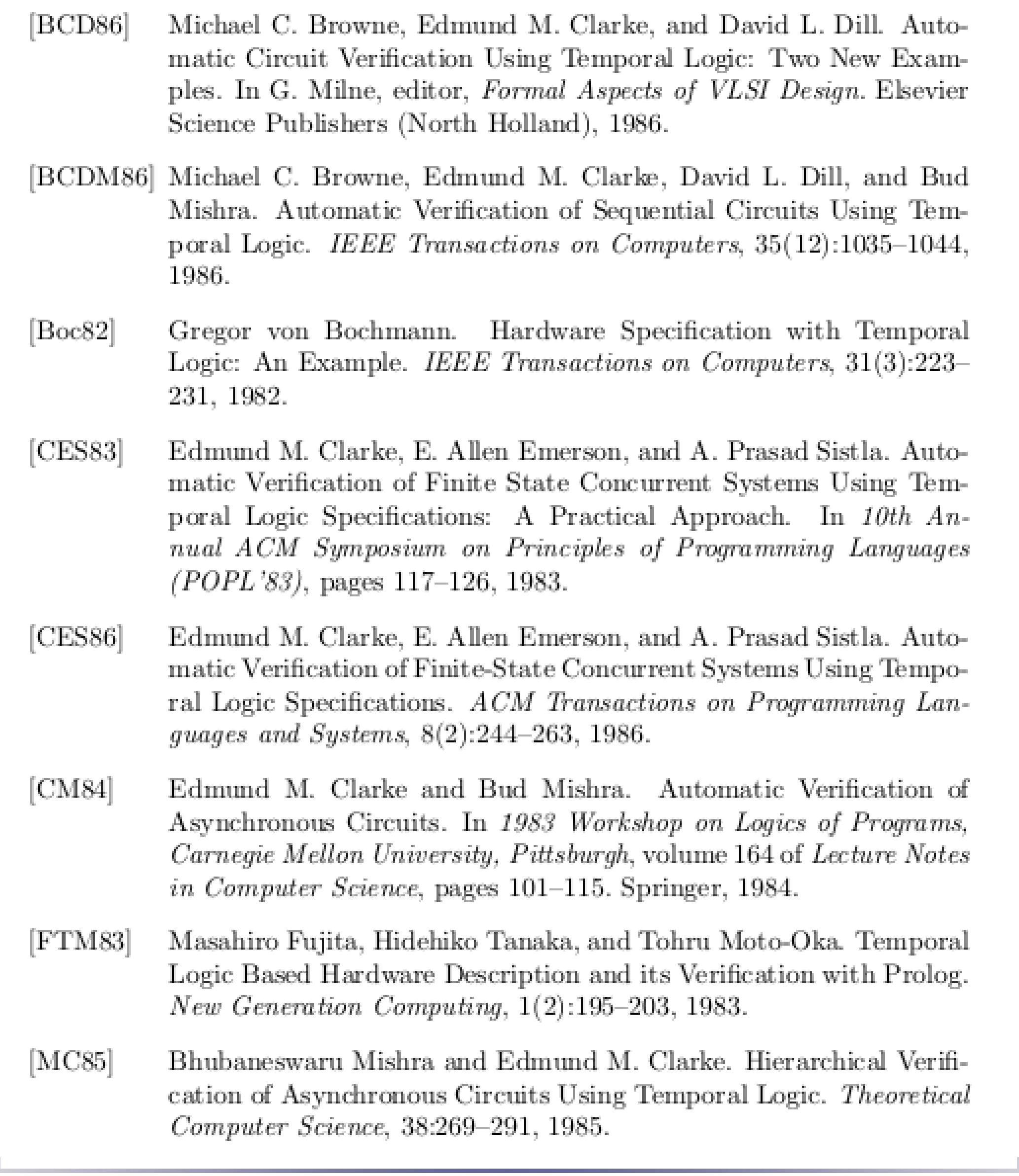


Formal specification, using temporal logic, of asynchronous circuits and sequential circuits

Verification of these circuits using state-space exploration and/or model checking

The EMC model checker revealed an error in a FIFO queue circuit element published in a popular textbook on VLSI design

[1982]



[BCD86] Michael C. Browne, Edmund M. Clarke, and David L. Dill. Automatic Circuit Verification Using Temporal Logic: Two New Examples. In G. Milne, editor, *Formal Aspects of VLSI Design*. Elsevier Science Publishers (North Holland), 1986.

[BCDM86] Michael C. Browne, Edmund M. Clarke, David L. Dill, and Bud Mishra. Automatic Verification of Sequential Circuits Using Temporal Logic. *IEEE Transactions on Computers*, 35(12):1035–1044, 1986.

[Boc82] Gregor von Bochmann. Hardware Specification with Temporal Logic: An Example. *IEEE Transactions on Computers*, 31(3):223–231, 1982.

[CES83] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications: A Practical Approach. In *10th Annual ACM Symposium on Principles of Programming Languages (POPL'83)*, pages 117–126, 1983.

[CES86] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

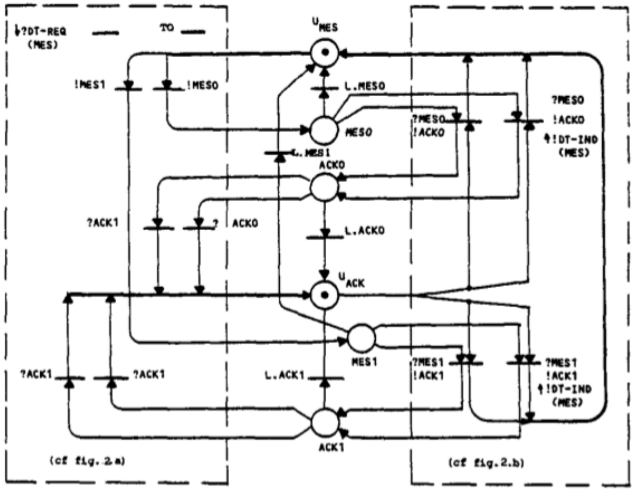
[CM84] Edmund M. Clarke and Bud Mishra. Automatic Verification of Asynchronous Circuits. In *1983 Workshop on Logics of Programs, Carnegie Mellon University, Pittsburgh*, volume 164 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 1984.

[FTM83] Masahiro Fujita, Hidehiko Tanaka, and Tohru Moto-Oka. Temporal Logic Based Hardware Description and its Verification with Prolog. *New Generation Computing*, 1(2):195–203, 1983.

[MC85] Bhubaneswaru Mishra and Edmund M. Clarke. Hierarchical Verification of Asynchronous Circuits Using Temporal Logic. *Theoretical Computer Science*, 38:269–291, 1985.

1983

[Billington-Bearman-Wilbur Ham]
[Courtiat-Ayache-Algayres]
[Jürgensen-Vuong]



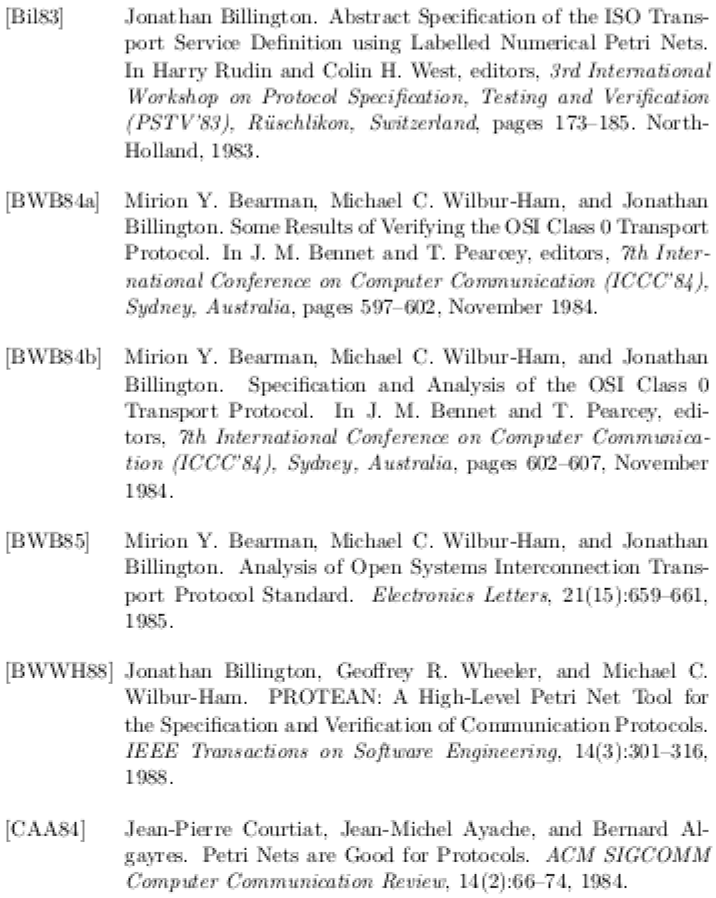
a) Global model interconnection
FIGURE 4. Alternating bit protocol and service

Formal specifications with (extended Petri nets) of the OSI transport layer protocol

Three independent teams, one using the PROTEAN tool (Austria) and two using the OGIVE/OVIDE tool (France)

Various properties checked (general, specific, structural). No harmful error found

[1983]



[Bil83] Jonathan Billington. Abstract Specification of the ISO Transport Service Definition using Labelled Numerical Petri Nets. In Harry Rudin and Colin H. West, editors, *3rd International Workshop on Protocol Specification, Testing and Verification (PSTV'83), Rüschlikon, Switzerland*, pages 173–185. North-Holland, 1983.

[BWB84a] Mirion Y. Bearman, Michael C. Wilbur-Ham, and Jonathan Billington. Some Results of Verifying the OSI Class 0 Transport Protocol. In J. M. Bennet and T. Pearcey, editors, *7th International Conference on Computer Communication (ICCC'84), Sydney, Australia*, pages 597–602, November 1984.

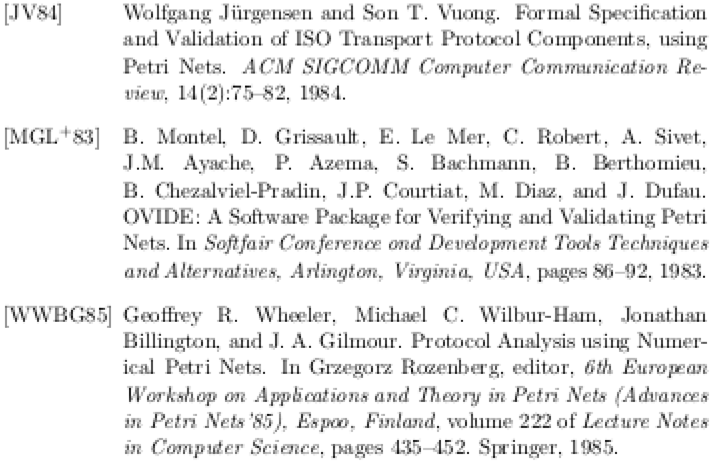
[BWB84b] Mirion Y. Bearman, Michael C. Wilbur-Ham, and Jonathan Billington. Specification and Analysis of the OSI Class 0 Transport Protocol. In J. M. Bennet and T. Pearcey, editors, *7th International Conference on Computer Communication (ICCC'84), Sydney, Australia*, pages 602–607, November 1984.

[BWB85] Mirion Y. Bearman, Michael C. Wilbur-Ham, and Jonathan Billington. Analysis of Open Systems Interconnection Transport Protocol Standard. *Electronics Letters*, 21(15):659–661, 1985.

[BWWH88] Jonathan Billington, Geoffrey R. Wheeler, and Michael C. Wilbur-Ham. PROTEAN: A High-Level Petri Net Tool for the Specification and Verification of Communication Protocols. *IEEE Transactions on Software Engineering*, 14(3):301–316, 1988.

[CAA84] Jean-Pierre Courtiat, Jean-Michel Ayache, and Bernard Algayres. Petri Nets are Good for Protocols. *ACM SIGCOMM Computer Communication Review*, 14(2):66–74, 1984.

[1983]



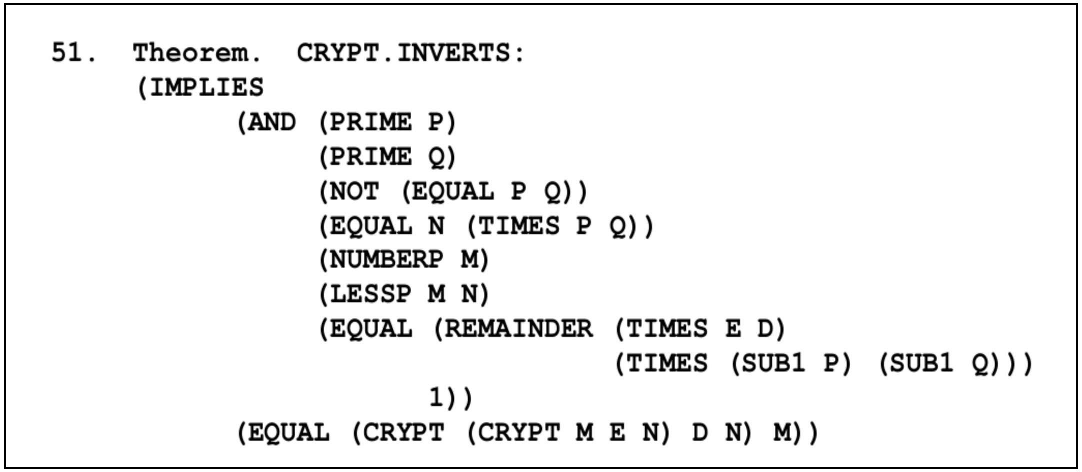
[JV84] Wolfgang Jürgensen and Son T. Vuong. Formal Specification and Validation of ISO Transport Protocol Components, using Petri Nets. *ACM SIGCOMM Computer Communication Review*, 14(2):75–82, 1984.

[MGL+83] B. Montel, D. Grissault, E. Le Mer, C. Robert, A. Sivet, J.M. Ayache, P. Azema, S. Bachmann, B. Berthomieu, B. Chezalviel-Pradin, J.P. Courtiat, M. Diaz, and J. Dufau. OVIDE: A Software Package for Verifying and Validating Petri Nets. In *Softfair Conference and Development Tools Techniques and Alternatives, Arlington, Virginia, USA*, pages 86–92, 1983.

[WWBG85] Geoffrey R. Wheeler, Michael C. Wilbur-Ham, Jonathan Billington, and J. A. Gilmour. Protocol Analysis using Numerical Petri Nets. In Grzegorz Rozenberg, editor, *6th European Workshop on Applications and Theory in Petri Nets (Advances in Petri Nets'85), Espoo, Finland*, volume 222 of *Lecture Notes in Computer Science*, pages 435–452. Springer, 1985.

1984

[Boyer-Moore-Shankar]



```
51.  Theorem.  CRYPT.INVERTS:
     (IMPLIES
           (AND (PRIME P)
                (PRIME Q)
                (NOT (EQUAL P Q))
                (EQUAL N (TIMES P Q))
                (NUMBERP M)
                (LESSP M N)
                (EQUAL (REMAINDER (TIMES E D)
                                  (TIMES (SUB1 P) (SUB1 Q)))
                       1))
           (EQUAL (CRYPT (CRYPT M E N) D N) M))
```

Automated proof checking using the NQTHM (Boyer-Moore) theorem prover of fundamental theorems of computer science:

– unsolvability of the halting problem,

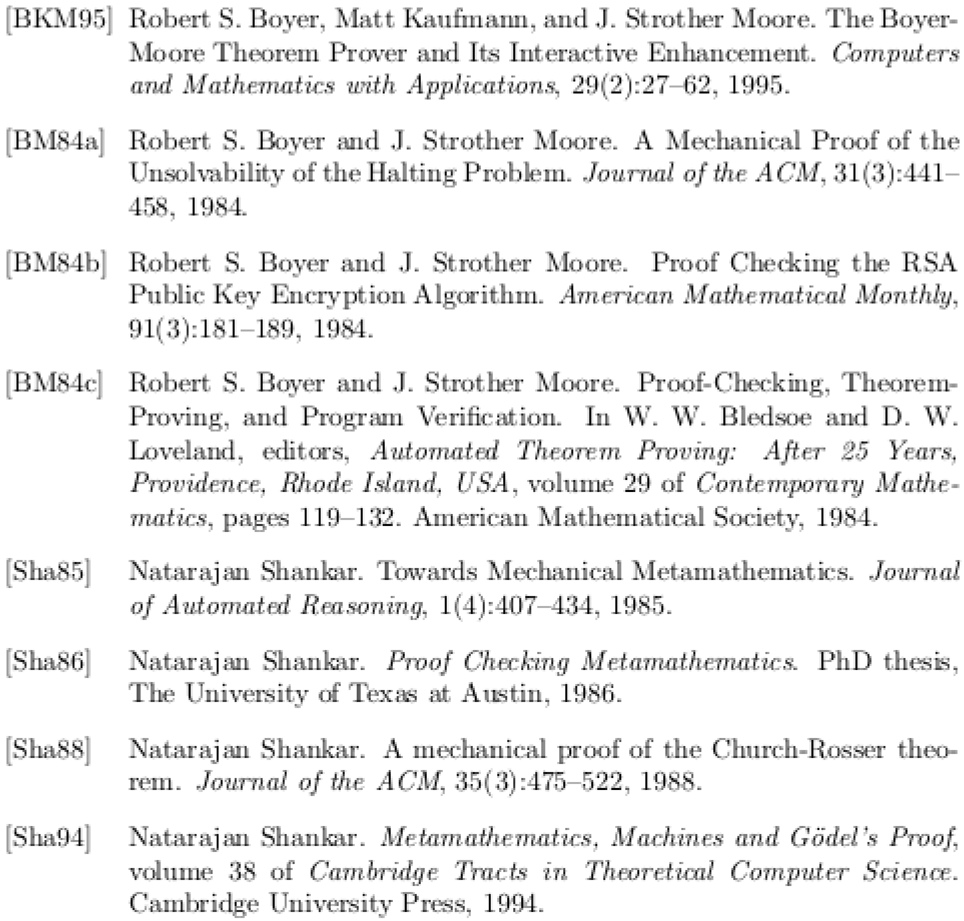
– Gödel's first incompleteness theorem

– Church-Rosser theorem of λ-calculus

and other theorems of practical value:

– invertibility of the RSA encryption algorithm

[1984]



[BKM95] Robert S. Boyer, Matt Kaufmann, and J. Strother Moore. The Boyer-Moore Theorem Prover and Its Interactive Enhancement. *Computers and Mathematics with Applications*, 29(2):27–62, 1995.

[BM84a] Robert S. Boyer and J. Strother Moore. A Mechanical Proof of the Unsolvability of the Halting Problem. *Journal of the ACM*, 31(3):441–458, 1984.

[BM84b] Robert S. Boyer and J. Strother Moore. Proof Checking the RSA Public Key Encryption Algorithm. *American Mathematical Monthly*, 91(3):181–189, 1984.

[BM84c] Robert S. Boyer and J. Strother Moore. Proof-Checking, Theorem-Proving, and Program Verification. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years, Providence, Rhode Island, USA*, volume 29 of *Contemporary Mathematics*, pages 119–132. American Mathematical Society, 1984.

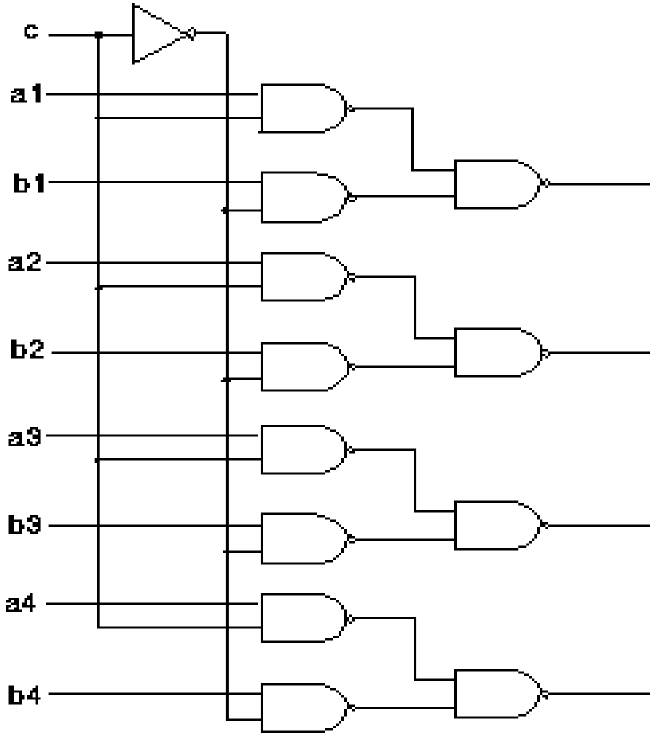
[Sha85] Natarajan Shankar. Towards Mechanical Metamathematics. *Journal of Automated Reasoning*, 1(4):407–434, 1985.

[Sha86] Natarajan Shankar. *Proof Checking Metamathematics*. PhD thesis, The University of Texas at Austin, 1986.

[Sha88] Natarajan Shankar. A mechanical proof of the Church-Rosser theorem. *Journal of the ACM*, 35(3):475–522, 1988.

[Sha94] Natarajan Shankar. *Metamathematics, Machines and Gödel's Proof*, volume 38 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1994.

1985

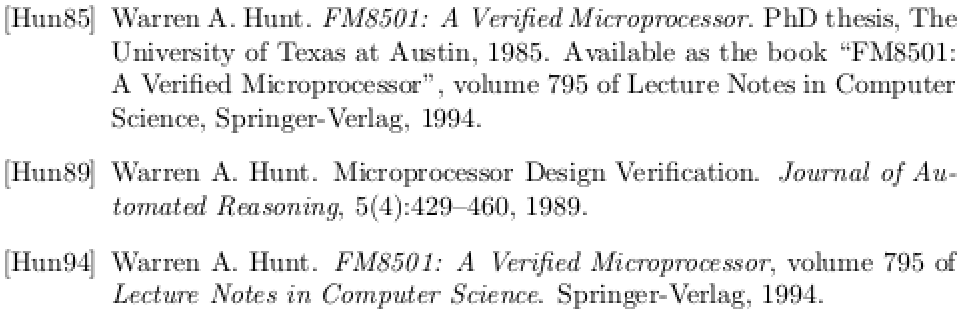


[Hunt]

Formal verification of the 16-bit FM8501 microprocessor using the NQTHM theorem prover

This was the first verified microprocessor, followed by many others

[1985]



[Hun85] Warren A. Hunt. *FM8501: A Verified Microprocessor*. PhD thesis, The University of Texas at Austin, 1985. Available as the book "FM8501: A Verified Microprocessor", volume 795 of Lecture Notes in Computer Science, Springer-Verlag, 1994.

[Hun89] Warren A. Hunt. Microprocessor Design Verification. *Journal of Automated Reasoning*, 5(4):429–460, 1989.

[Hun94] Warren A. Hunt. *FM8501: A Verified Microprocessor*, volume 795 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

1986

[West] + [Rudin, Zafiropulo]

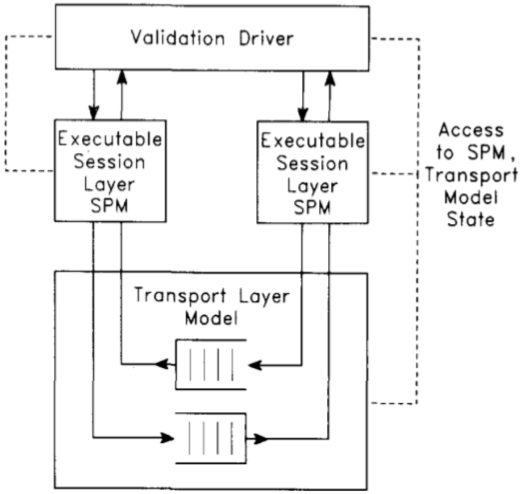

Validation Driver
Executable Session Layer SPM
Executable Session Layer SPM
Access to SPM, Transport Model State
Transport Layer Model

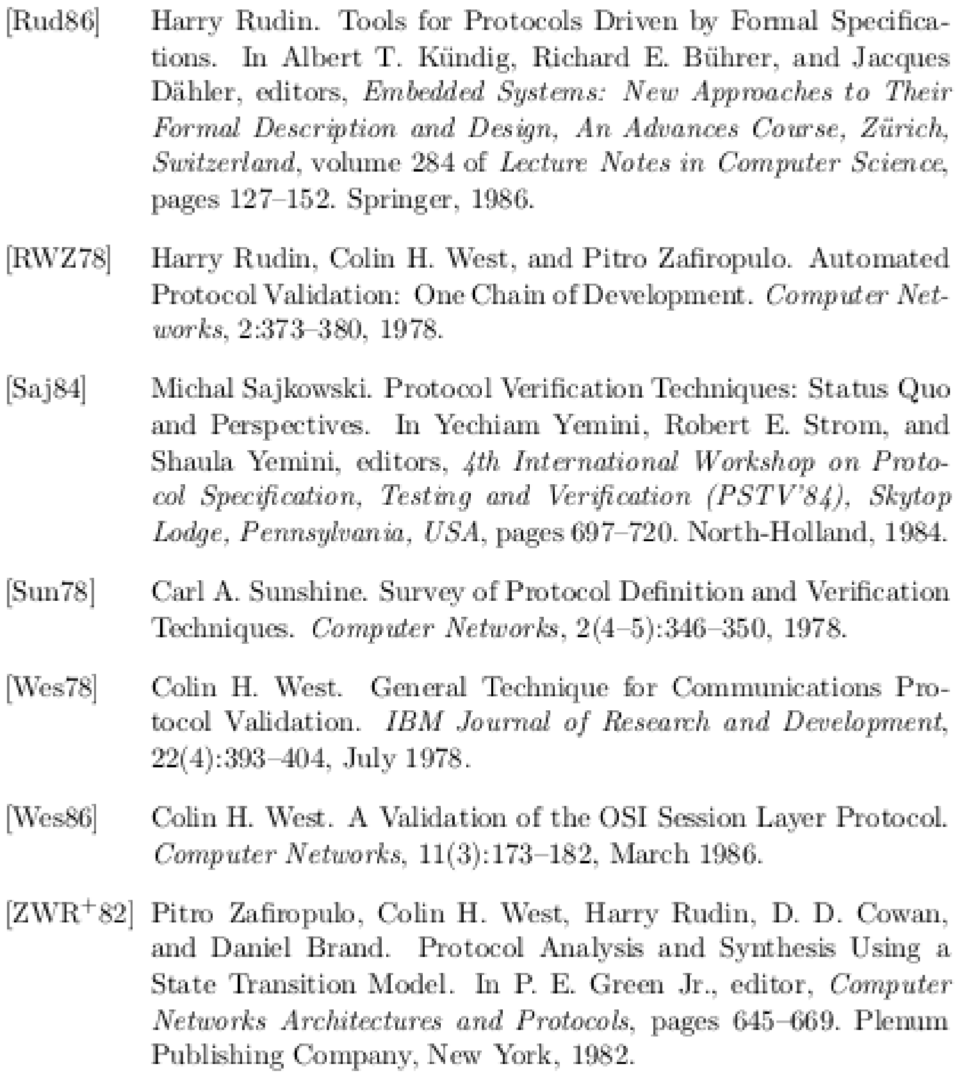
Fig. 2. The OSI session layer validation system.

Formal analysis of the (slightly simplified) OSI session layer protocol, using finite state machines communicating by bounded FIFO queues

Verification using automated protocol validation techniques based on state space exploration.

Various errors found and reported to ISO

[1986]



[Rud86] Harry Rudin. Tools for Protocols Driven by Formal Specifications. In Albert T. Kündig, Richard E. Bührer, and Jacques Dähler, editors, *Embedded Systems: New Approaches to Their Formal Description and Design, An Advances Course, Zürich, Switzerland*, volume 284 of *Lecture Notes in Computer Science*, pages 127–152. Springer, 1986.

[RWZ78] Harry Rudin, Colin H. West, and Pitro Zafiropulo. Automated Protocol Validation: One Chain of Development. *Computer Networks*, 2:373–380, 1978.

[Saj84] Michal Sajkowski. Protocol Verification Techniques: Status Quo and Perspectives. In Yechiam Yemini, Robert E. Strom, and Shaula Yemini, editors, *4th International Workshop on Protocol Specification, Testing and Verification (PSTV'84), Skytop Lodge, Pennsylvania, USA*, pages 697–720. North-Holland, 1984.

[Sun78] Carl A. Sunshine. Survey of Protocol Definition and Verification Techniques. *Computer Networks*, 2(4–5):346–350, 1978.

[Wes78] Colin H. West. General Technique for Communications Protocol Validation. *IBM Journal of Research and Development*, 22(4):393–404, July 1978.

[Wes86] Colin H. West. A Validation of the OSI Session Layer Protocol. *Computer Networks*, 11(3):173–182, March 1986.

[ZWR$^+$82] Pitro Zafiropulo, Colin H. West, Harry Rudin, D. D. Cowan, and Daniel Brand. Protocol Analysis and Synthesis Using a State Transition Model. In P. E. Green Jr., editor, *Computer Networks Architectures and Protocols*, pages 645–669. Plenum Publishing Company, New York, 1982.

1987

[Graf-Richier-Rodriguez-Sifakis-Voiron]

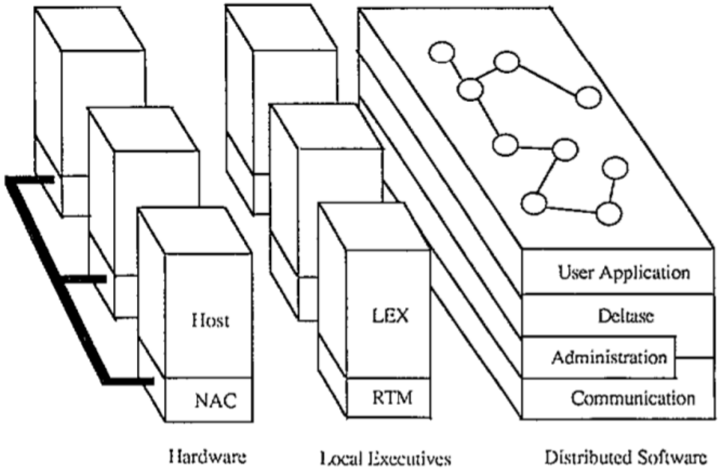
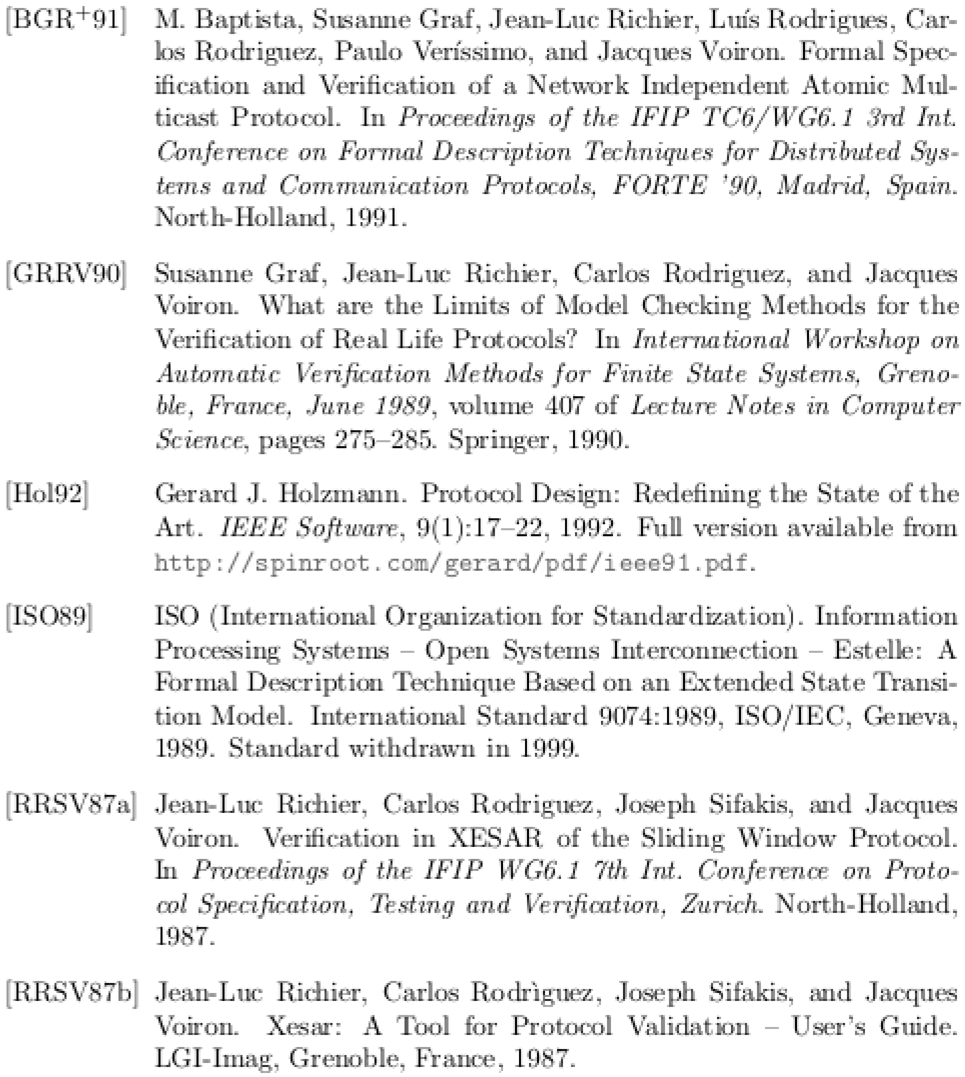


Figure 1. Abstract view of the Delta-4 architecture.

Specification in Estelle/R (a rendezvous-based variant of the protocol description language Estelle) and verification using the Xesar model checker of two protocols:

- a generic sliding window protocol, later intensively studied by the computer-aided verification community under the name "bounded retransmission protocol"
- an atomic multicast protocol for the Delta-4 distributed dependable architecture

[1987]



[BGR+91] M. Baptista, Susanne Graf, Jean-Luc Richier, Luís Rodrigues, Carlos Rodriguez, Paulo Veríssimo, and Jacques Voiron. Formal Specification and Verification of a Network Independent Atomic Multicast Protocol. In *Proceedings of the IFIP TC6/WG6.1 3rd Int. Conference on Formal Description Techniques for Distributed Systems and Communication Protocols, FORTE '90, Madrid, Spain.* North-Holland, 1991.

[GRRV90] Susanne Graf, Jean-Luc Richier, Carlos Rodriguez, and Jacques Voiron. What are the Limits of Model Checking Methods for the Verification of Real Life Protocols? In *International Workshop on Automatic Verification Methods for Finite State Systems, Grenoble, France, June 1989*, volume 407 of *Lecture Notes in Computer Science*, pages 275–285. Springer, 1990.

[Hol92] Gerard J. Holzmann. Protocol Design: Redefining the State of the Art. *IEEE Software*, 9(1):17–22, 1992. Full version available from `http://spinroot.com/gerard/pdf/ieee91.pdf`.

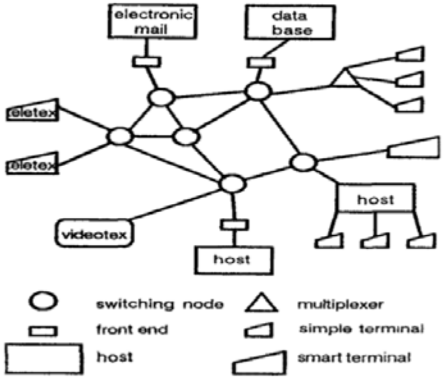
[ISO89] ISO (International Organization for Standardization). Information Processing Systems – Open Systems Interconnection – Estelle: A Formal Description Technique Based on an Extended State Transition Model. International Standard 9074:1989, ISO/IEC, Geneva, 1989. Standard withdrawn in 1999.

[RRSV87a] Jean-Luc Richier, Carlos Rodriguez, Joseph Sifakis, and Jacques Voiron. Verification in XESAR of the Sliding Window Protocol. In *Proceedings of the IFIP WG6.1 7th Int. Conference on Protocol Specification, Testing and Verification, Zurich*. North-Holland, 1987.

[RRSV87b] Jean-Luc Richier, Carlos Rodrìguez, Joseph Sifakis, and Jacques Voiron. Xesar: A Tool for Protocol Validation – User's Guide. LGI-Imag, Grenoble, France, 1987.

1988

[Ajubi-Aujla-Caneschi-Ferreira Pires-Freestone-Scollo-Turner-van de Heijden-van de Lagemaat-van Sinderen-Vissers-Widya]



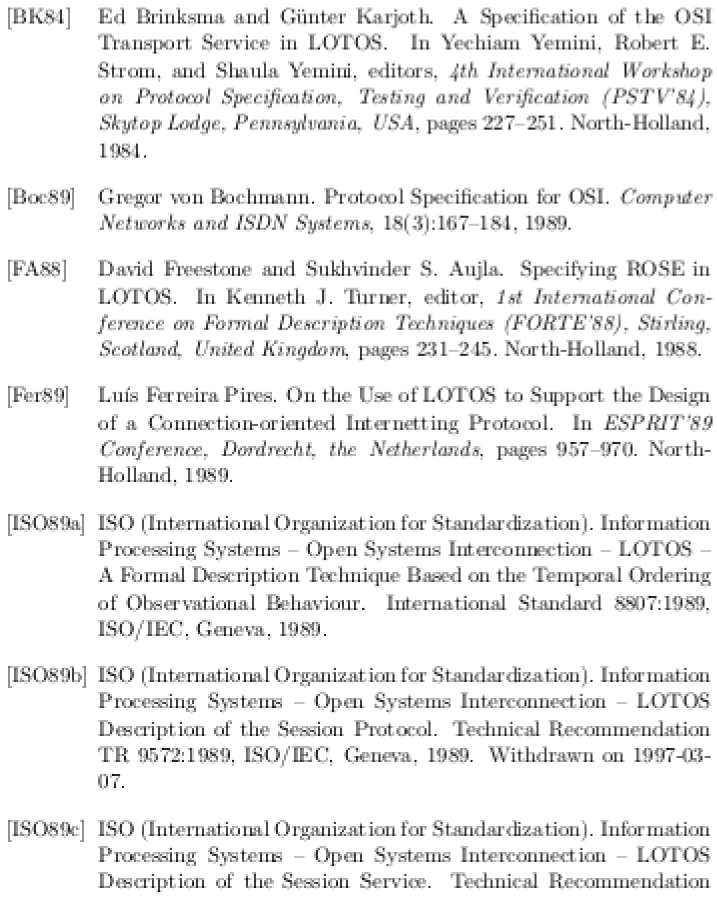
Configuration of an arbitrary OSI system.

Formal methods were used to specify OSI (Open System Interconnection) standards in a concise, unambiguous, implementation-neutral way.

LOTOS has been used intensively to specify :

- the service and protocol of the session layer
- the service and protocol of the transport layer
- the service and protocol of the network layer
- at the application layer
  - ROSE (Remote Operations Service Element) service
  - CCR (Commitment, Concurrency and Recovery) service and protocol
  - DTP (Distributed Transaction Processing) protocol

[1988]



[BK84] Ed Brinksma and Günter Karjoth. A Specification of the OSI Transport Service in LOTOS. In Yechiam Yemini, Robert E. Strom, and Shaula Yemini, editors, *4th International Workshop on Protocol Specification, Testing and Verification (PSTV'84), Skytop Lodge, Pennsylvania, USA*, pages 227–251. North-Holland, 1984.

[Boc89] Gregor von Bochmann. Protocol Specification for OSI. *Computer Networks and ISDN Systems*, 18(3):167–184, 1989.

[FA88] David Freestone and Sukhvinder S. Aujla. Specifying ROSE in LOTOS. In Kenneth J. Turner, editor, *1st International Conference on Formal Description Techniques (FORTE'88), Stirling, Scotland, United Kingdom*, pages 231–245. North-Holland, 1988.

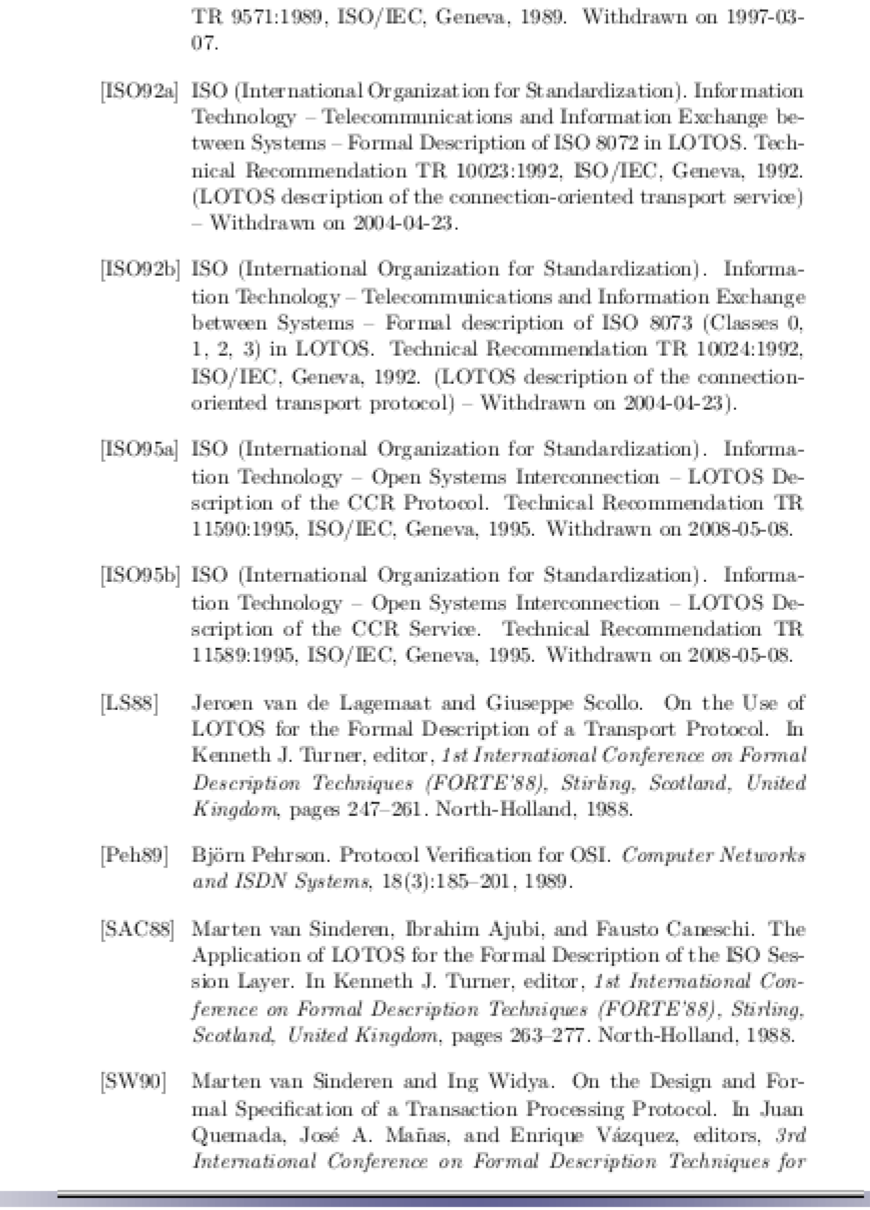
[Fer89] Luís Ferreira Pires. On the Use of LOTOS to Support the Design of a Connection-oriented Internetting Protocol. In *ESPRIT'89 Conference, Dordrecht, the Netherlands*, pages 957–970. North-Holland, 1989.

[ISO89a] ISO (International Organization for Standardization). Information Processing Systems – Open Systems Interconnection – LOTOS – A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour. International Standard 8807:1989, ISO/IEC, Geneva, 1989.

[ISO89b] ISO (International Organization for Standardization). Information Processing Systems – Open Systems Interconnection – LOTOS Description of the Session Protocol. Technical Recommendation TR 9572:1989, ISO/IEC, Geneva, 1989. Withdrawn on 1997-03-07.

[ISO89c] ISO (International Organization for Standardization). Information Processing Systems – Open Systems Interconnection – LOTOS Description of the Session Service. Technical Recommendation

[1988]



TR 9571:1989, ISO/IEC, Geneva, 1989. Withdrawn on 1997-03-07.

[ISO92a] ISO (International Organization for Standardization). Information Technology – Telecommunications and Information Exchange between Systems – Formal Description of ISO 8072 in LOTOS. Technical Recommendation TR 10023:1992, ISO/IEC, Geneva, 1992. (LOTOS description of the connection-oriented transport service) – Withdrawn on 2004-04-23.

[ISO92b] ISO (International Organization for Standardization). Information Technology – Telecommunications and Information Exchange between Systems – Formal description of ISO 8073 (Classes 0, 1, 2, 3) in LOTOS. Technical Recommendation TR 10024:1992, ISO/IEC, Geneva, 1992. (LOTOS description of the connection-oriented transport protocol) – Withdrawn on 2004-04-23).

[ISO95a] ISO (International Organization for Standardization). Information Technology – Open Systems Interconnection – LOTOS Description of the CCR Protocol. Technical Recommendation TR 11590:1995, ISO/IEC, Geneva, 1995. Withdrawn on 2008-05-08.

[ISO95b] ISO (International Organization for Standardization). Information Technology – Open Systems Interconnection – LOTOS Description of the CCR Service. Technical Recommendation TR 11589:1995, ISO/IEC, Geneva, 1995. Withdrawn on 2008-05-08.

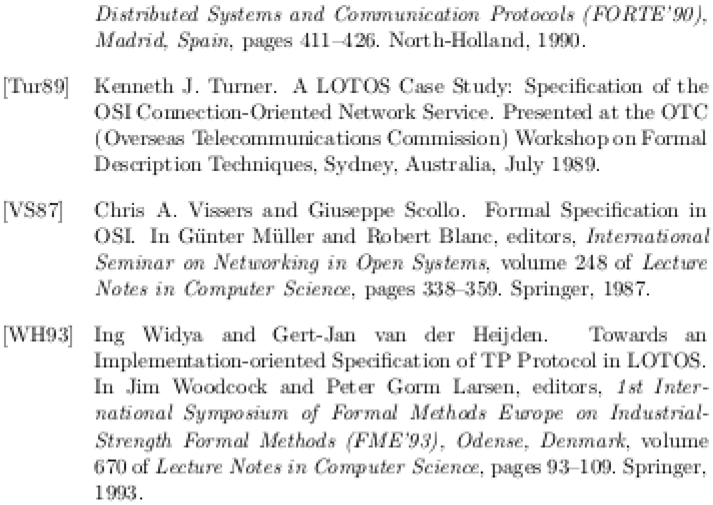
[LS88] Jeroen van de Lagemaat and Giuseppe Scollo. On the Use of LOTOS for the Formal Description of a Transport Protocol. In Kenneth J. Turner, editor, *1st International Conference on Formal Description Techniques (FORTE'88), Stirling, Scotland, United Kingdom*, pages 247–261. North-Holland, 1988.

[Peh89] Björn Pehrson. Protocol Verification for OSI. *Computer Networks and ISDN Systems*, 18(3):185–201, 1989.

[SAC88] Marten van Sinderen, Ibrahim Ajubi, and Fausto Caneschi. The Application of LOTOS for the Formal Description of the ISO Session Layer. In Kenneth J. Turner, editor, *1st International Conference on Formal Description Techniques (FORTE'88), Stirling, Scotland, United Kingdom*, pages 263–277. North-Holland, 1988.

[SW90] Marten van Sinderen and Ing Widya. On the Design and Formal Specification of a Transaction Processing Protocol. In Juan Quemada, José A. Mañas, and Enrique Vázquez, editors, *3rd International Conference on Formal Description Techniques for*

[1988]



Distributed Systems and Communication Protocols (FORTE'90), Madrid, Spain, pages 411–426. North-Holland, 1990.

[Tur89] Kenneth J. Turner. A LOTOS Case Study: Specification of the OSI Connection-Oriented Network Service. Presented at the OTC (Overseas Telecommunications Commission) Workshop on Formal Description Techniques, Sydney, Australia, July 1989.

[VS87] Chris A. Vissers and Giuseppe Scollo. Formal Specification in OSI. In Günter Müller and Robert Blanc, editors, *International Seminar on Networking in Open Systems*, volume 248 of *Lecture Notes in Computer Science*, pages 338–359. Springer, 1987.

[WH93] Ing Widya and Gert-Jan van der Heijden. Towards an Implementation-oriented Specification of TP Protocol in LOTOS. In Jim Woodcock and Peter Gorm Larsen, editors, *1st International Symposium of Formal Methods Europe on Industrial-Strength Formal Methods (FME'93), Odense, Denmark*, volume 670 of *Lecture Notes in Computer Science*, pages 93–109. Springer, 1993.

1989

[Stålmarck-Säflund-Borälv-Sheeran] [Groote-Koorn-Van Vlijmen] [Fokkink] [Eisner]

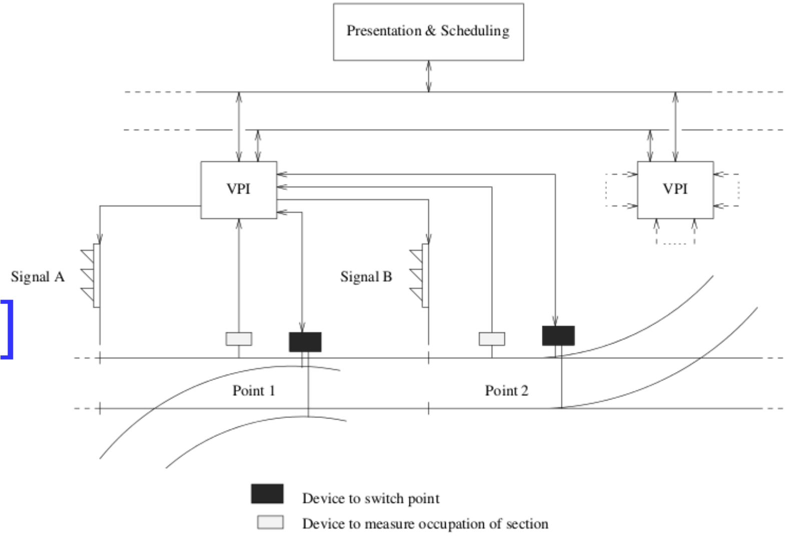
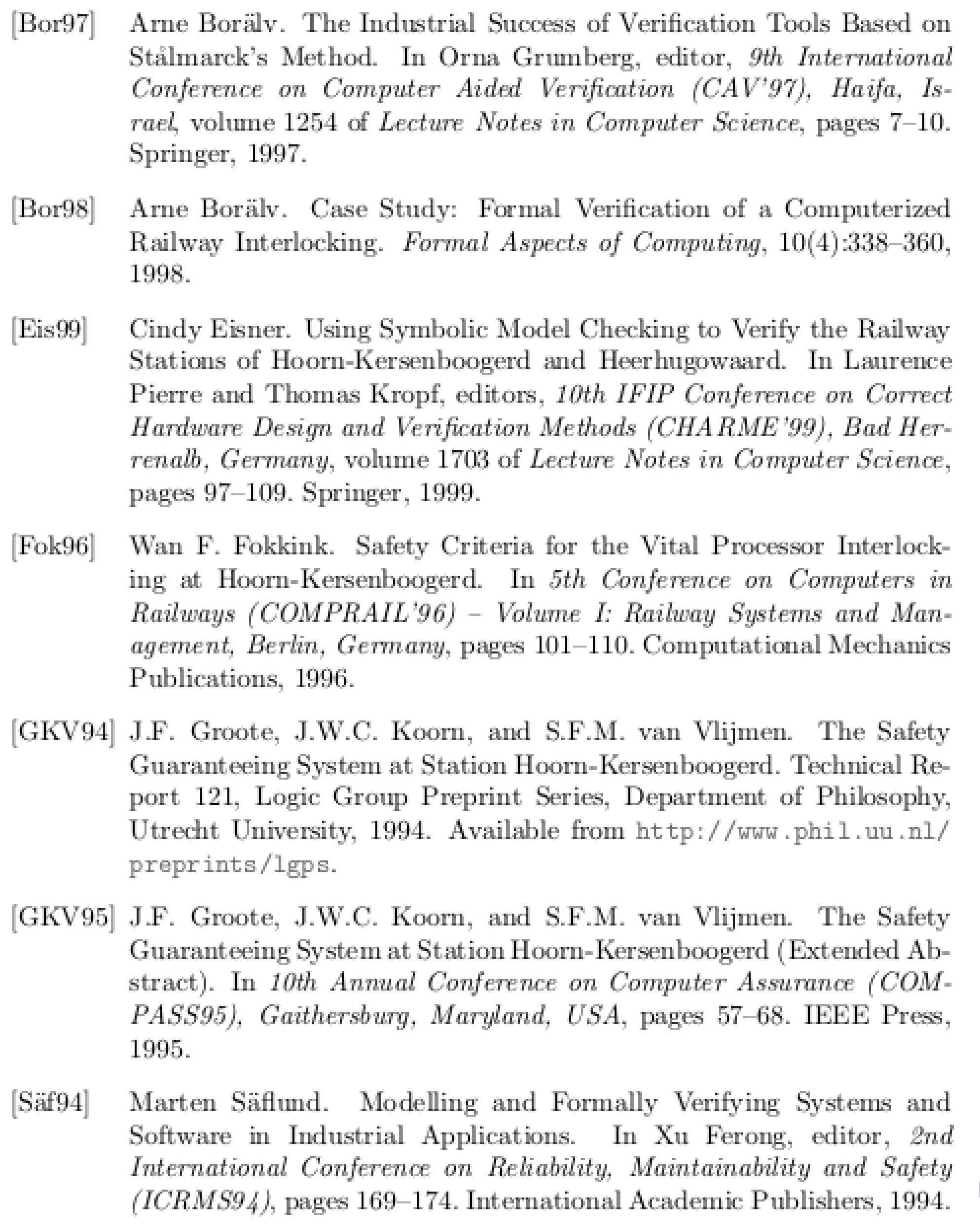

Presentation & Scheduling
VPI
VPI
Signal A
Signal B
Point 1
Point 2
■ Device to switch point
□ Device to measure occupation of section

Figure 1: VPIs and their environment

Formal verification, using a novel algorithm for efficiently proving large theorems of propositional logic, of safety-critical applications such as

- reverse flushing control in a nuclear plant's emergency cooling system
- landing gear control for a military aircraft
- railway signalling systems (interlocking verification)

[1989]



[Bor97] Arne Borälv. The Industrial Success of Verification Tools Based on Stålmarck's Method. In Orna Grumberg, editor, *9th International Conference on Computer Aided Verification (CAV'97), Haifa, Israel*, volume 1254 of *Lecture Notes in Computer Science*, pages 7–10. Springer, 1997.

[Bor98] Arne Borälv. Case Study: Formal Verification of a Computerized Railway Interlocking. *Formal Aspects of Computing*, 10(4):338–360, 1998.

[Eis99] Cindy Eisner. Using Symbolic Model Checking to Verify the Railway Stations of Hoorn-Kersenboogerd and Heerhugowaard. In Laurence Pierre and Thomas Kropf, editors, *10th IFIP Conference on Correct Hardware Design and Verification Methods (CHARME'99), Bad Herrenalb, Germany*, volume 1703 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1999.

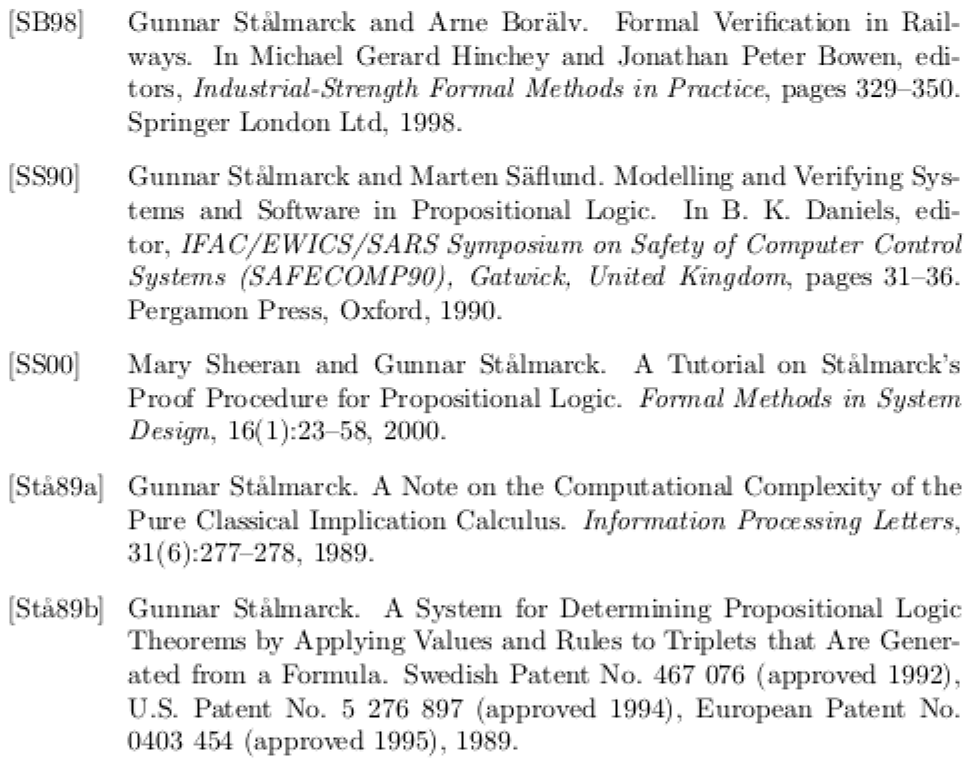
[Fok96] Wan F. Fokkink. Safety Criteria for the Vital Processor Interlocking at Hoorn-Kersenboogerd. In *5th Conference on Computers in Railways (COMPRAIL'96) – Volume I: Railway Systems and Management, Berlin, Germany*, pages 101–110. Computational Mechanics Publications, 1996.

[GKV94] J.F. Groote, J.W.C. Koorn, and S.F.M. van Vlijmen. The Safety Guaranteeing System at Station Hoorn-Kersenboogerd. Technical Report 121, Logic Group Preprint Series, Department of Philosophy, Utrecht University, 1994. Available from `http://www.phil.uu.nl/preprints/lgps`.

[GKV95] J.F. Groote, J.W.C. Koorn, and S.F.M. van Vlijmen. The Safety Guaranteeing System at Station Hoorn-Kersenboogerd (Extended Abstract). In *10th Annual Conference on Computer Assurance (COMPASS95), Gaithersburg, Maryland, USA*, pages 57–68. IEEE Press, 1995.

[Säf94] Marten Säflund. Modelling and Formally Verifying Systems and Software in Industrial Applications. In Xu Ferong, editor, *2nd International Conference on Reliability, Maintainability and Safety (ICRMS94)*, pages 169–174. International Academic Publishers, 1994.

[1989]



[SB98] Gunnar Stålmarck and Arne Borälv. Formal Verification in Railways. In Michael Gerard Hinchey and Jonathan Peter Bowen, editors, *Industrial-Strength Formal Methods in Practice*, pages 329–350. Springer London Ltd, 1998.

[SS90] Gunnar Stålmarck and Marten Säflund. Modelling and Verifying Systems and Software in Propositional Logic. In B. K. Daniels, editor, *IFAC/EWICS/SARS Symposium on Safety of Computer Control Systems (SAFECOMP90), Gatwick, United Kingdom*, pages 31–36. Pergamon Press, Oxford, 1990.

[SS00] Mary Sheeran and Gunnar Stålmarck. A Tutorial on Stålmarck's Proof Procedure for Propositional Logic. *Formal Methods in System Design*, 16(1):23–58, 2000.

[Stå89a] Gunnar Stålmarck. A Note on the Computational Complexity of the Pure Classical Implication Calculus. *Information Processing Letters*, 31(6):277–278, 1989.

[Stå89b] Gunnar Stålmarck. A System for Determining Propositional Logic Theorems by Applying Values and Rules to Triplets that Are Generated from a Formula. Swedish Patent No. 467 076 (approved 1992), U.S. Patent No. 5 276 897 (approved 1994), European Patent No. 0403 454 (approved 1995), 1989.

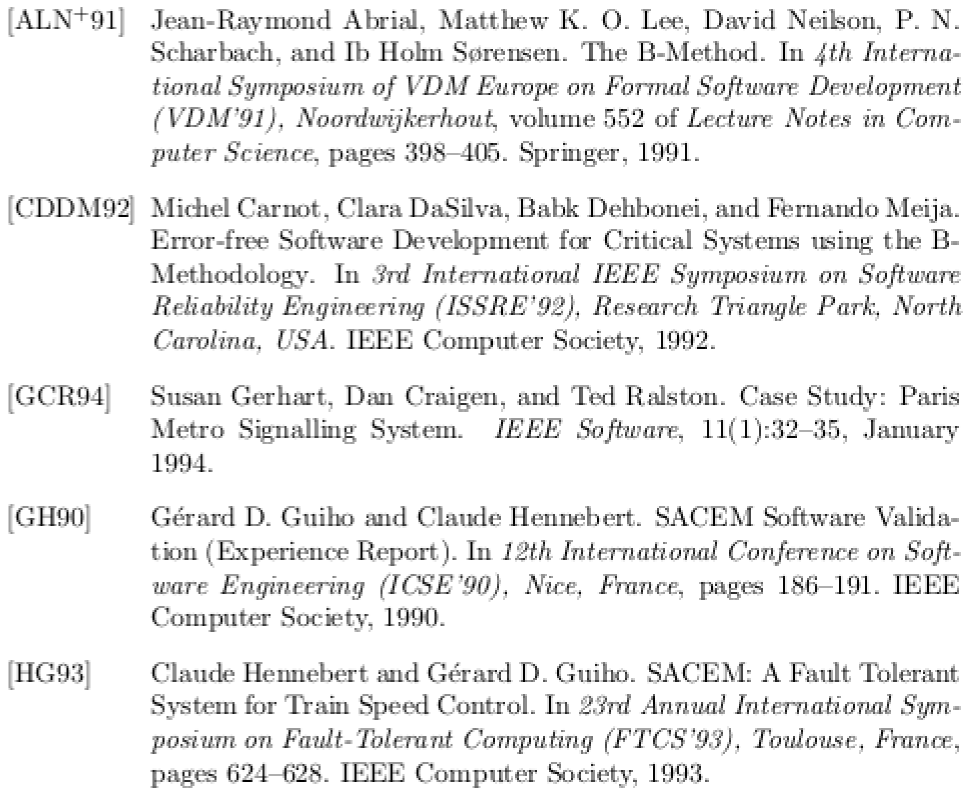
1990

[Guiho-Hennebert]

Formal specification using the B language and correctness proofs using Hoare-like logic (in addition to traditional code inspection and testing approaches) of SACEM, a fault-tolerant railway signalling system that controls train speed, signals drivers, and activates emergency brakes

SACEM was the first safety-critical software system certified by the French railway authority; it is used in Paris (800,000 passengers carried per day) and other cities in the world

[1990]



[ALN^{+}91] Jean-Raymond Abrial, Matthew K. O. Lee, David Neilson, P. N. Scharbach, and Ib Holm Sørensen. The B-Method. In *4th International Symposium of VDM Europe on Formal Software Development (VDM'91), Noordwijkerhout*, volume 552 of *Lecture Notes in Computer Science*, pages 398–405. Springer, 1991.

[CDDM92] Michel Carnot, Clara DaSilva, Babk Dehbonei, and Fernando Meija. Error-free Software Development for Critical Systems using the B-Methodology. In *3rd International IEEE Symposium on Software Reliability Engineering (ISSRE'92), Research Triangle Park, North Carolina, USA*. IEEE Computer Society, 1992.

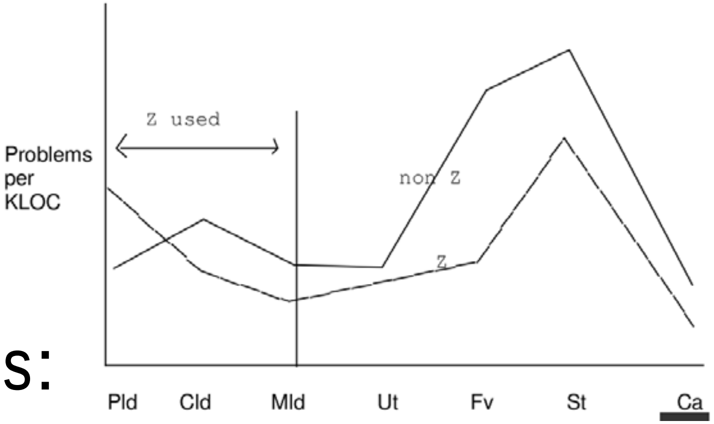
[GCR94] Susan Gerhart, Dan Craigen, and Ted Ralston. Case Study: Paris Metro Signalling System. *IEEE Software*, 11(1):32–35, January 1994.

[GH90] Gérard D. Guiho and Claude Hennebert. SACEM Software Validation (Experience Report). In *12th International Conference on Software Engineering (ICSE'90), Nice, France*, pages 186–191. IEEE Computer Society, 1990.

[HG93] Claude Hennebert and Gérard D. Guiho. SACEM: A Fault Tolerant System for Train Speed Control. In *23rd Annual International Symposium on Fault-Tolerant Computing (FTCS'93), Toulouse, France*, pages 624–628. IEEE Computer Society, 1993.

1991

[Houston-King]



Use of Z in two large IBM projects:

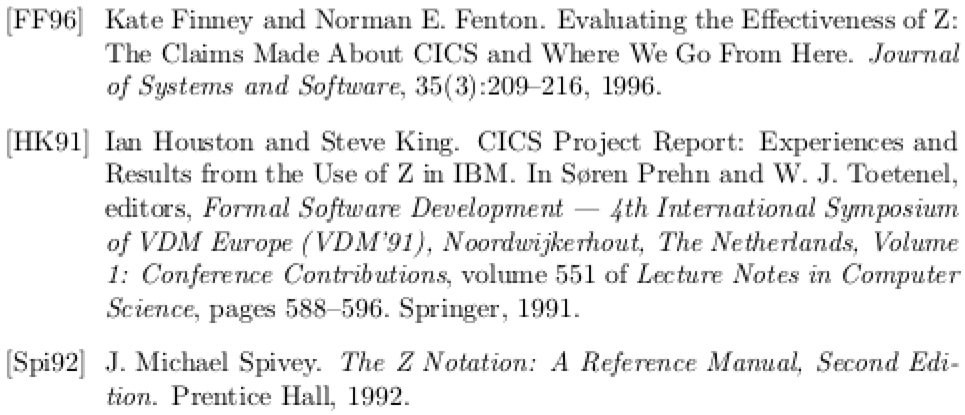
- a major new release IBM's CICS (Customer Information Control System) on-line transaction processing system
- the API (Application Programming Interface) of CICS.

Very few tools were used (only syntax and type checkers)

The authors report that the use of Z reduced the number of errors by a factor of 2.5 and saved 9% of the total development cost

But the significance of these conclusions was questioned later [Finney-Fenton]

[1991]



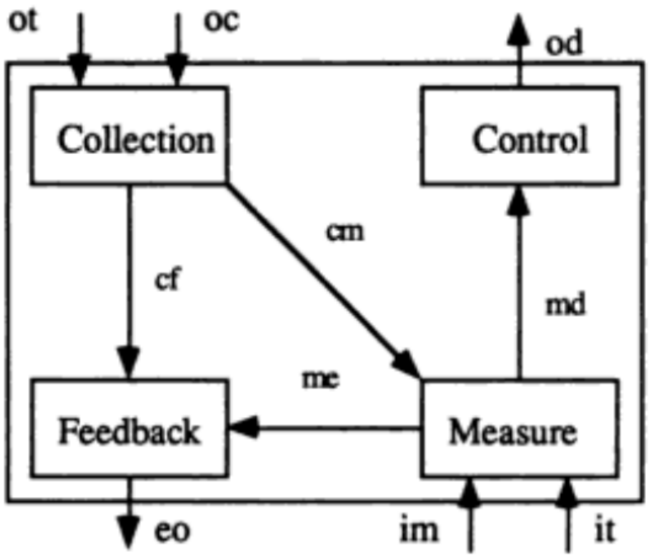
[FF96] Kate Finney and Norman E. Fenton. Evaluating the Effectiveness of Z: The Claims Made About CICS and Where We Go From Here. *Journal of Systems and Software*, 35(3):209–216, 1996.

[HK91] Ian Houston and Steve King. CICS Project Report: Experiences and Results from the Use of Z in IBM. In Søren Prehn and W. J. Toetenel, editors, *Formal Software Development — 4th International Symposium of VDM Europe (VDM'91), Noordwijkerhout, The Netherlands, Volume 1: Conference Contributions*, volume 551 of *Lecture Notes in Computer Science*, pages 588–596. Springer, 1991.

[Spi92] J. Michael Spivey. *The Z Notation: A Reference Manual, Second Edition.* Prentice Hall, 1992.

1992

[Paterno-Faconti] + [Duke-Harrison-Fornari-Mezzanotte-Sciacchitano-Löwgren] [Markopoulos]

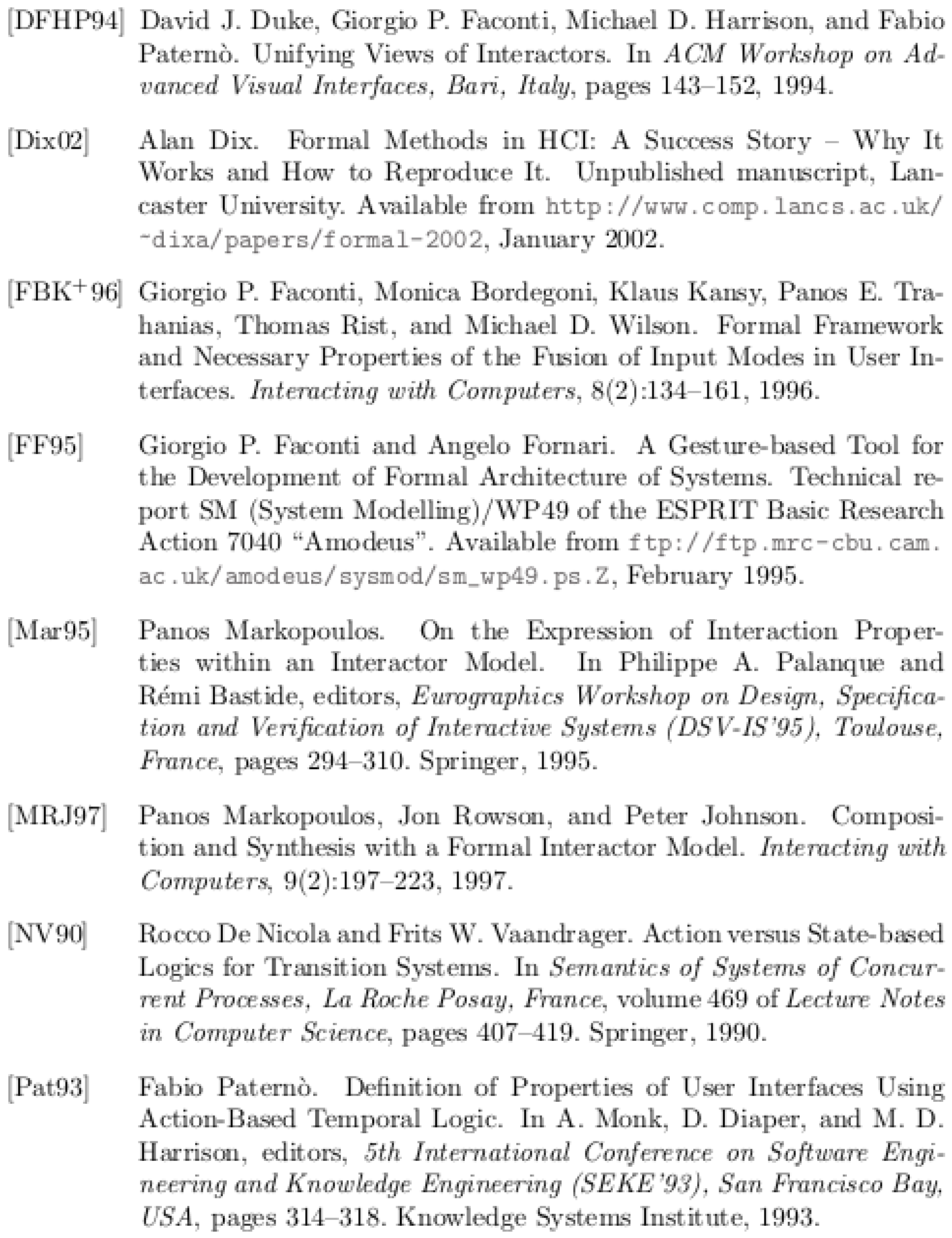


The Architecture of an Interactor

Formalization, using LOTOS and ACTL logic of the concept of "interactor", a software architectural model used to build complex user interface software

Several applications, e.g., MATIS, a multimodal interactive system enabling users to get information about flight schedules using speech, mouse and keyboard, or a combination of them

[1992]



[DFHP94] David J. Duke, Giorgio P. Faconti, Michael D. Harrison, and Fabio Paternò. Unifying Views of Interactors. In *ACM Workshop on Advanced Visual Interfaces, Bari, Italy*, pages 143–152, 1994.

[Dix02] Alan Dix. Formal Methods in HCI: A Success Story – Why It Works and How to Reproduce It. Unpublished manuscript, Lancaster University. Available from `http://www.comp.lancs.ac.uk/~dixa/papers/formal-2002`, January 2002.

[FBK+96] Giorgio P. Faconti, Monica Bordegoni, Klaus Kansy, Panos E. Trahanias, Thomas Rist, and Michael D. Wilson. Formal Framework and Necessary Properties of the Fusion of Input Modes in User Interfaces. *Interacting with Computers*, 8(2):134–161, 1996.

[FF95] Giorgio P. Faconti and Angelo Fornari. A Gesture-based Tool for the Development of Formal Architecture of Systems. Technical report SM (System Modelling)/WP49 of the ESPRIT Basic Research Action 7040 “Amodeus”. Available from `ftp://ftp.mrc-cbu.cam.ac.uk/amodeus/sysmod/sm_wp49.ps.Z`, February 1995.

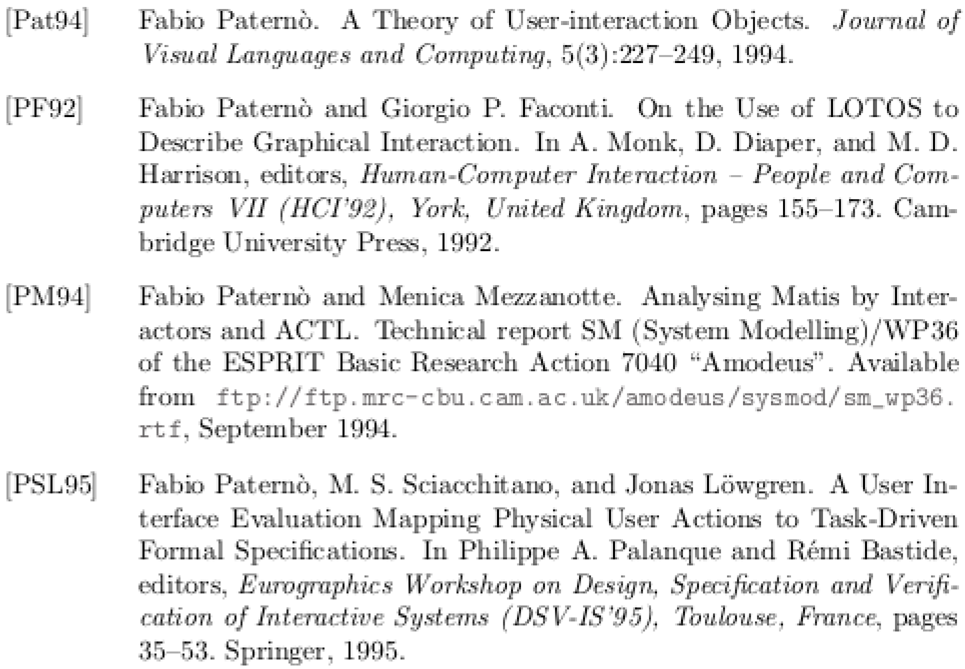
[Mar95] Panos Markopoulos. On the Expression of Interaction Properties within an Interactor Model. In Philippe A. Palanque and Rémi Bastide, editors, *Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'95), Toulouse, France*, pages 294–310. Springer, 1995.

[MRJ97] Panos Markopoulos, Jon Rowson, and Peter Johnson. Composition and Synthesis with a Formal Interactor Model. *Interacting with Computers*, 9(2):197–223, 1997.

[NV90] Rocco De Nicola and Frits W. Vaandrager. Action versus State-based Logics for Transition Systems. In *Semantics of Systems of Concurrent Processes, La Roche Posay, France*, volume 469 of *Lecture Notes in Computer Science*, pages 407–419. Springer, 1990.

[Pat93] Fabio Paternò. Definition of Properties of User Interfaces Using Action-Based Temporal Logic. In A. Monk, D. Diaper, and M. D. Harrison, editors, *5th International Conference on Software Engineering and Knowledge Engineering (SEKE'93), San Francisco Bay, USA*, pages 314–318. Knowledge Systems Institute, 1993.

[1992]



[Pat94] Fabio Paternò. A Theory of User-interaction Objects. *Journal of Visual Languages and Computing*, 5(3):227–249, 1994.

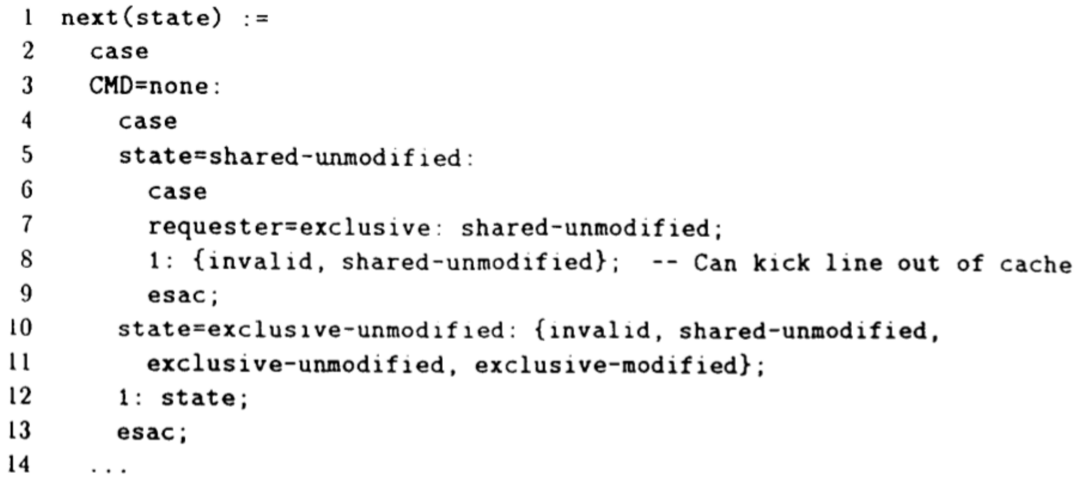
[PF92] Fabio Paternò and Giorgio P. Faconti. On the Use of LOTOS to Describe Graphical Interaction. In A. Monk, D. Diaper, and M. D. Harrison, editors, *Human-Computer Interaction – People and Computers VII (HCI'92), York, United Kingdom*, pages 155–173. Cambridge University Press, 1992.

[PM94] Fabio Paternò and Menica Mezzanotte. Analysing Matis by Interactors and ACTL. Technical report SM (System Modelling)/WP36 of the ESPRIT Basic Research Action 7040 "Amodeus". Available from `ftp://ftp.mrc-cbu.cam.ac.uk/amodeus/sysmod/sm_wp36.rtf`, September 1994.

[PSL95] Fabio Paternò, M. S. Sciacchitano, and Jonas Löwgren. A User Interface Evaluation Mapping Physical User Actions to Task-Driven Formal Specifications. In Philippe A. Palanque and Rémi Bastide, editors, *Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'95), Toulouse, France*, pages 35–53. Springer, 1995.

1993

[Clarke-Grumberg-Hiraishi-Jha-Long-McMillan-Ness]



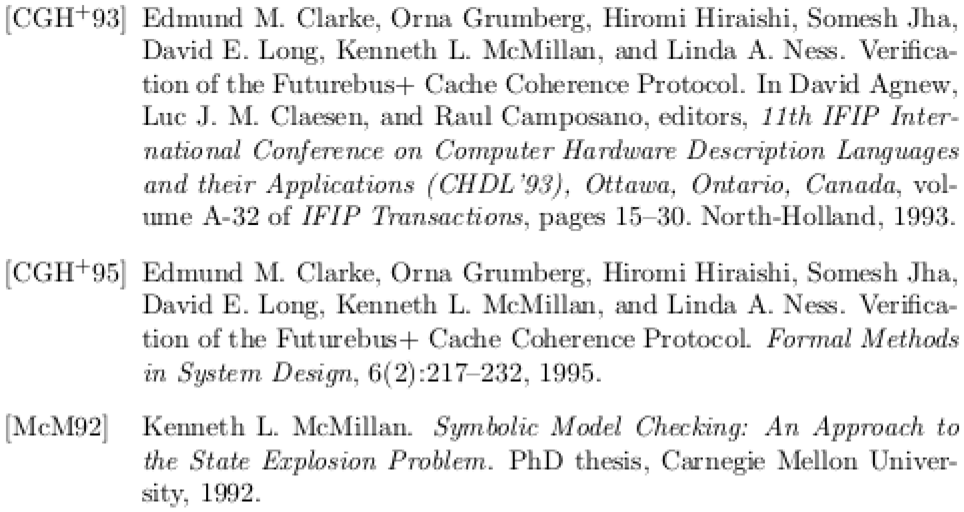
```
next(state) :=
  case
  CMD=none:
    case
    state=shared-unmodified:
      case
      requester=exclusive: shared-unmodified;
      1: {invalid, shared-unmodified};  -- Can kick line out of cache
      esac;
    state=exclusive-unmodified: {invalid, shared-unmodified,
      exclusive-unmodified, exclusive-modified};
    1: state;
    esac;
  ...
```

Formal specification and verification of the cache coherence protocol of IEEE standard 896.1-1991 "Futurebus+" using the SMV symbolic model checker

Several design errors previously undetected were found

First time that a formal verification tool was used to find errors in an IEEE standard

[1993]



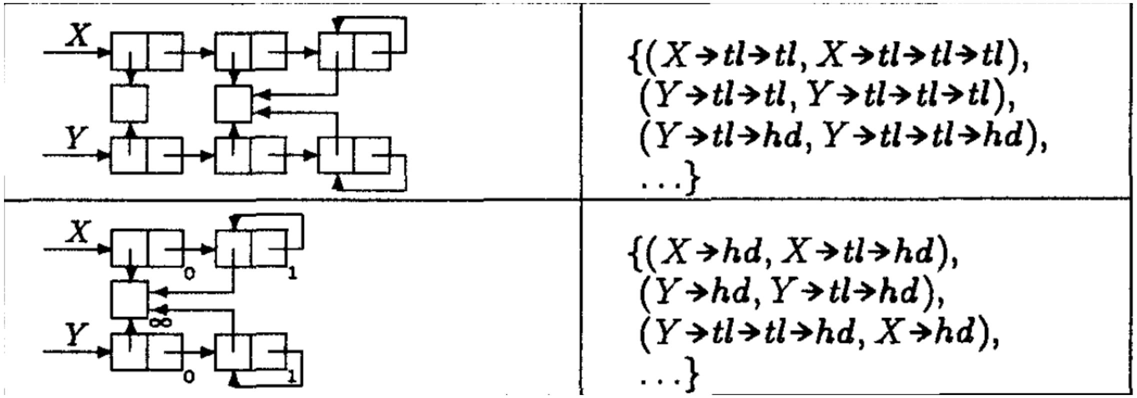
[CGH+93] Edmund M. Clarke, Orna Grumberg, Hiromi Hiraishi, Somesh Jha, David E. Long, Kenneth L. McMillan, and Linda A. Ness. Verification of the Futurebus+ Cache Coherence Protocol. In David Agnew, Luc J. M. Claesen, and Raul Camposano, editors, *11th IFIP International Conference on Computer Hardware Description Languages and their Applications (CHDL'93), Ottawa, Ontario, Canada*, volume A-32 of *IFIP Transactions*, pages 15–30. North-Holland, 1993.

[CGH+95] Edmund M. Clarke, Orna Grumberg, Hiromi Hiraishi, Somesh Jha, David E. Long, Kenneth L. McMillan, and Linda A. Ness. Verification of the Futurebus+ Cache Coherence Protocol. *Formal Methods in System Design*, 6(2):217–232, 1995.

[McM92] Kenneth L. McMillan. *Symbolic Model Checking: An Approach to the State Explosion Problem*. PhD thesis, Carnegie Mellon University, 1992.

1994

[Andersen] [Deutsch] [Evans-Guttag-Horning-Tan]



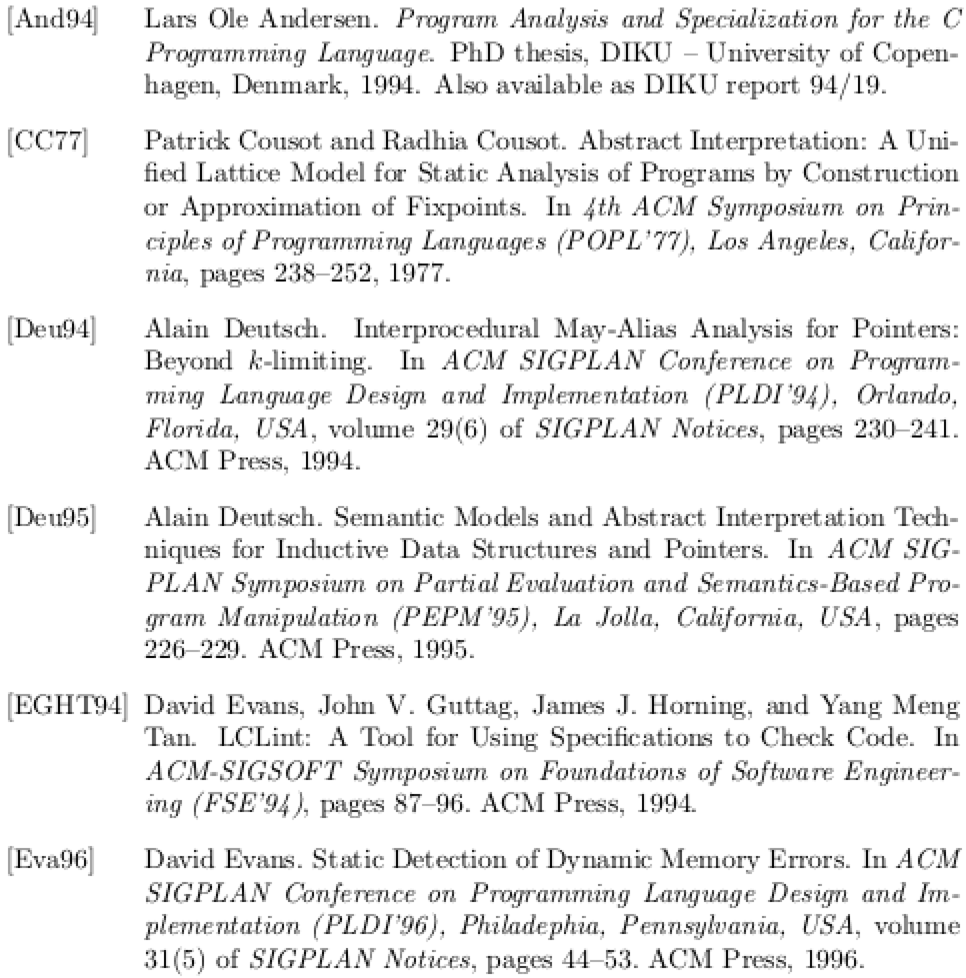
$\{(X\to tl\to tl,\ X\to tl\to tl\to tl),$
$(Y\to tl\to tl,\ Y\to tl\to tl\to tl),$
$(Y\to tl\to hd,\ Y\to tl\to tl\to hd),$
$\ldots\}$

$\{(X\to hd,\ X\to tl\to hd),$
$(Y\to hd,\ Y\to tl\to hd),$
$(Y\to tl\to tl\to hd,\ X\to hd),$
$\ldots\}$

Applications of the abstract interpretation to build static analyzers for C programs:

- LCLint annotation-assisted static checker (later extended to check dynamic memory allocation and buffer overflow vulnerabilities, and renamed into Splint)
- IABC static analysis tool for pointer manipulation and aliasing, which later went to marked under the name Polyspace Verifier

[1994]



[And94] Lars Ole Andersen. *Program Analysis and Specialization for the C Programming Language*. PhD thesis, DIKU – University of Copenhagen, Denmark, 1994. Also available as DIKU report 94/19.

[CC77] Patrick Cousot and Radhia Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *4th ACM Symposium on Principles of Programming Languages (POPL'77), Los Angeles, California*, pages 238–252, 1977.

[Deu94] Alain Deutsch. Interprocedural May-Alias Analysis for Pointers: Beyond *k*-limiting. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'94), Orlando, Florida, USA*, volume 29(6) of *SIGPLAN Notices*, pages 230–241. ACM Press, 1994.

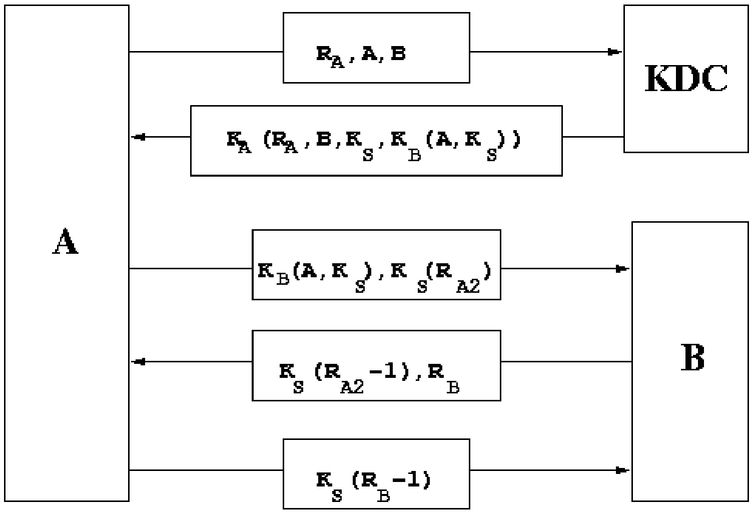
[Deu95] Alain Deutsch. Semantic Models and Abstract Interpretation Techniques for Inductive Data Structures and Pointers. In *ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM'95), La Jolla, California, USA*, pages 226–229. ACM Press, 1995.

[EGHT94] David Evans, John V. Guttag, James J. Horning, and Yang Meng Tan. LCLint: A Tool for Using Specifications to Check Code. In *ACM-SIGSOFT Symposium on Foundations of Software Engineering (FSE'94)*, pages 87–96. ACM Press, 1994.

[Eva96] David Evans. Static Detection of Dynamic Memory Errors. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'96), Philadephia, Pennsylvania, USA*, volume 31(5) of *SIGPLAN Notices*, pages 44–53. ACM Press, 1996.

1995

[Lowe]

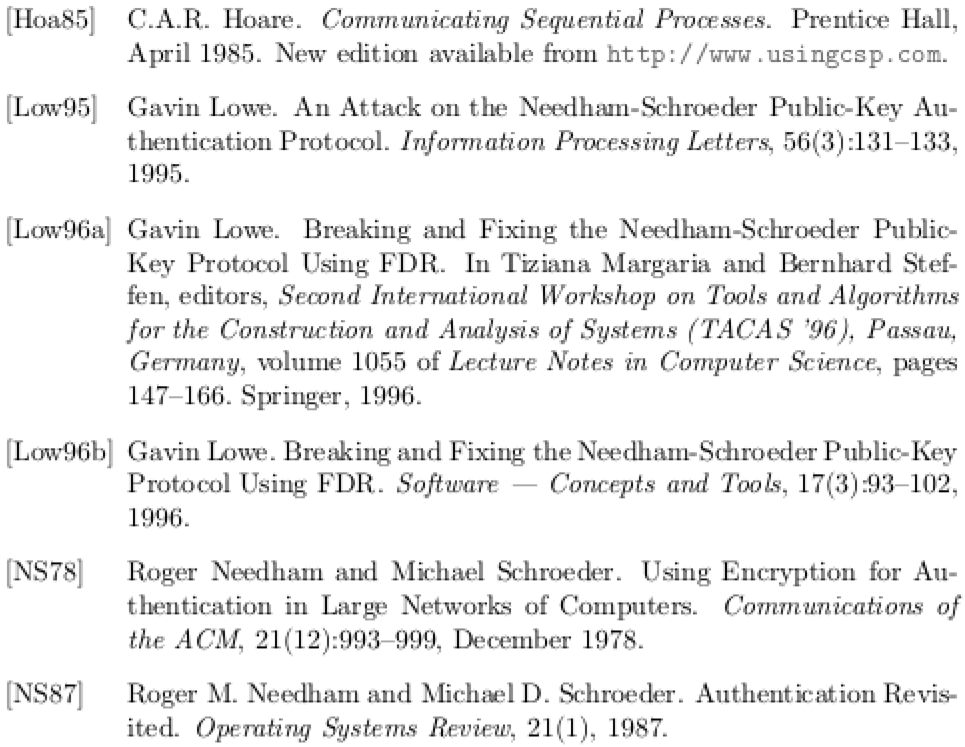


The Needham-Schroeder Authentication Protocol

Discovery, using CSP and FDR of an unknown, subtle "man-in-the-middle" attack in the classical Needham-Schroeder public-key protocol, which forms the basis of Kerberos authentication

This fueled a lot of research on formal methods and tools for the analysis of security protocols

[1995]



[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, April 1985. New edition available from `http://www.usingcsp.com`.

[Low95] Gavin Lowe. An Attack on the Needham-Schroeder Public-Key Authentication Protocol. *Information Processing Letters*, 56(3):131–133, 1995.

[Low96a] Gavin Lowe. Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR. In Tiziana Margaria and Bernhard Steffen, editors, *Second International Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '96), Passau, Germany*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer, 1996.

[Low96b] Gavin Lowe. Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR. *Software — Concepts and Tools*, 17(3):93–102, 1996.

[NS78] Roger Needham and Michael Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, December 1978.

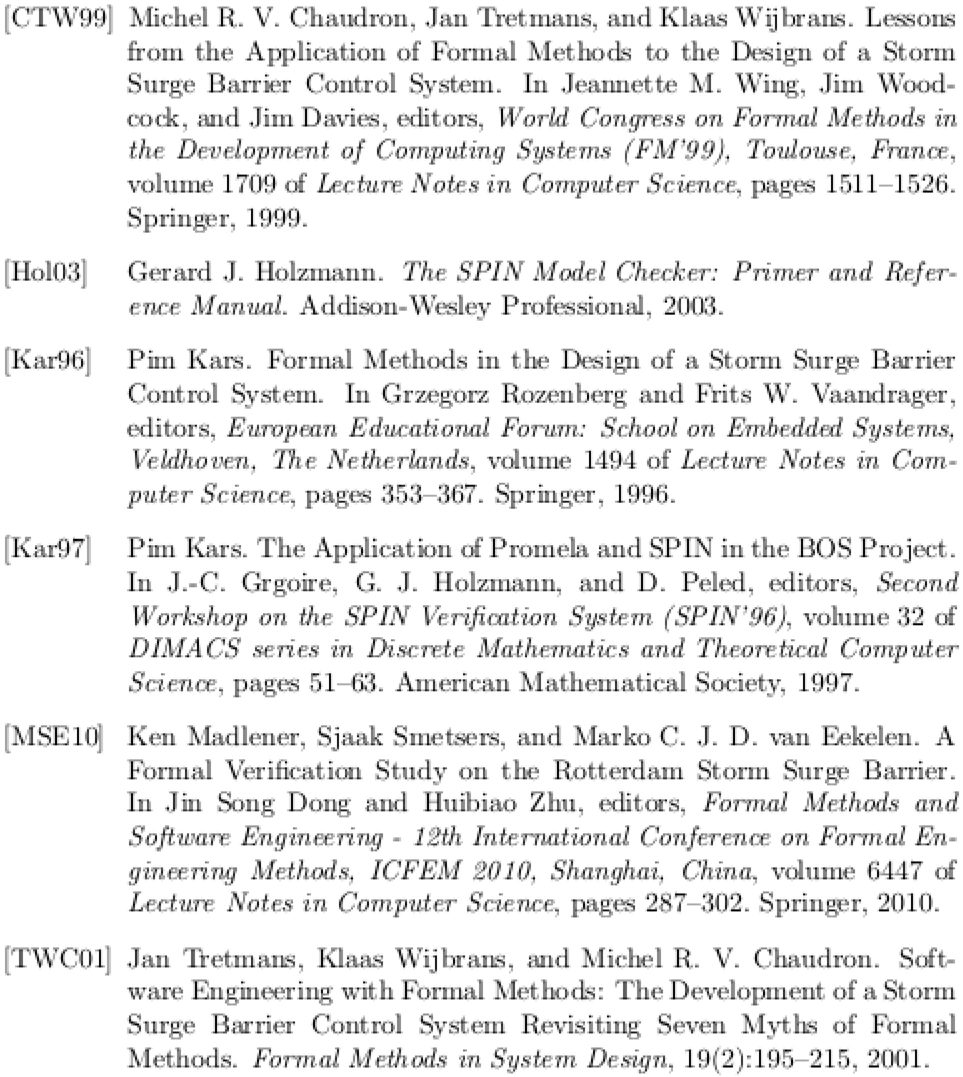
[NS87] Roger M. Needham and Michael D. Schroeder. Authentication Revisited. *Operating Systems Review*, 21(1), 1987.

1996

[Kars] [Chaudron-Tretmans-Wijbrans] [Madlener-Smetsers-van Eekelen]

Specification using Z and Promela, and model checking using SPIN of the software controlling the storm surge barrier that protects Rotterdam from flooding, a life-critical application certified at the highest safety integrity level (SIL4)

[1996]



[CTW99] Michel R. V. Chaudron, Jan Tretmans, and Klaas Wijbrans. Lessons from the Application of Formal Methods to the Design of a Storm Surge Barrier Control System. In Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *World Congress on Formal Methods in the Development of Computing Systems (FM'99), Toulouse, France*, volume 1709 of *Lecture Notes in Computer Science*, pages 1511–1526. Springer, 1999.

[Hol03] Gerard J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 2003.

[Kar96] Pim Kars. Formal Methods in the Design of a Storm Surge Barrier Control System. In Grzegorz Rozenberg and Frits W. Vaandrager, editors, *European Educational Forum: School on Embedded Systems, Veldhoven, The Netherlands*, volume 1494 of *Lecture Notes in Computer Science*, pages 353–367. Springer, 1996.

[Kar97] Pim Kars. The Application of Promela and SPIN in the BOS Project. In J.-C. Grgoire, G. J. Holzmann, and D. Peled, editors, *Second Workshop on the SPIN Verification System (SPIN'96)*, volume 32 of *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, pages 51–63. American Mathematical Society, 1997.

[MSE10] Ken Madlener, Sjaak Smetsers, and Marko C. J. D. van Eekelen. A Formal Verification Study on the Rotterdam Storm Surge Barrier. In Jin Song Dong and Huibiao Zhu, editors, *Formal Methods and Software Engineering - 12th International Conference on Formal Engineering Methods, ICFEM 2010, Shanghai, China*, volume 6447 of *Lecture Notes in Computer Science*, pages 287–302. Springer, 2010.

[TWC01] Jan Tretmans, Klaas Wijbrans, and Michel R. V. Chaudron. Software Engineering with Formal Methods: The Development of a Storm Surge Barrier Control System Revisiting Seven Myths of Formal Methods. *Formal Methods in System Design*, 19(2):195–215, 2001.

1997

[Luttik]
[Kühne-Hooman-de Roever]
[Sighireanu-Mateescu]

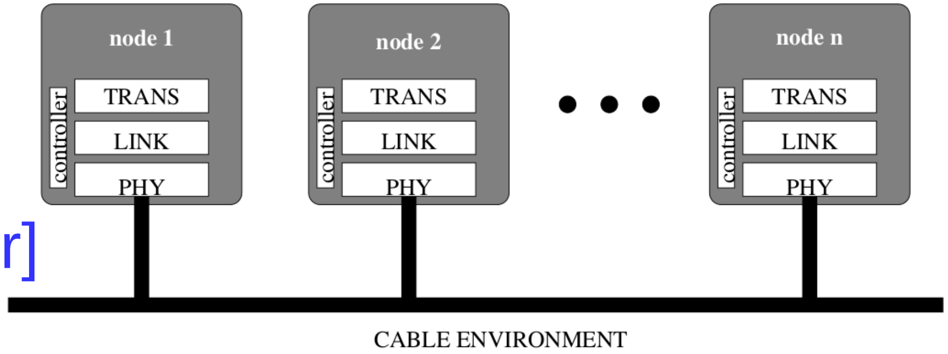


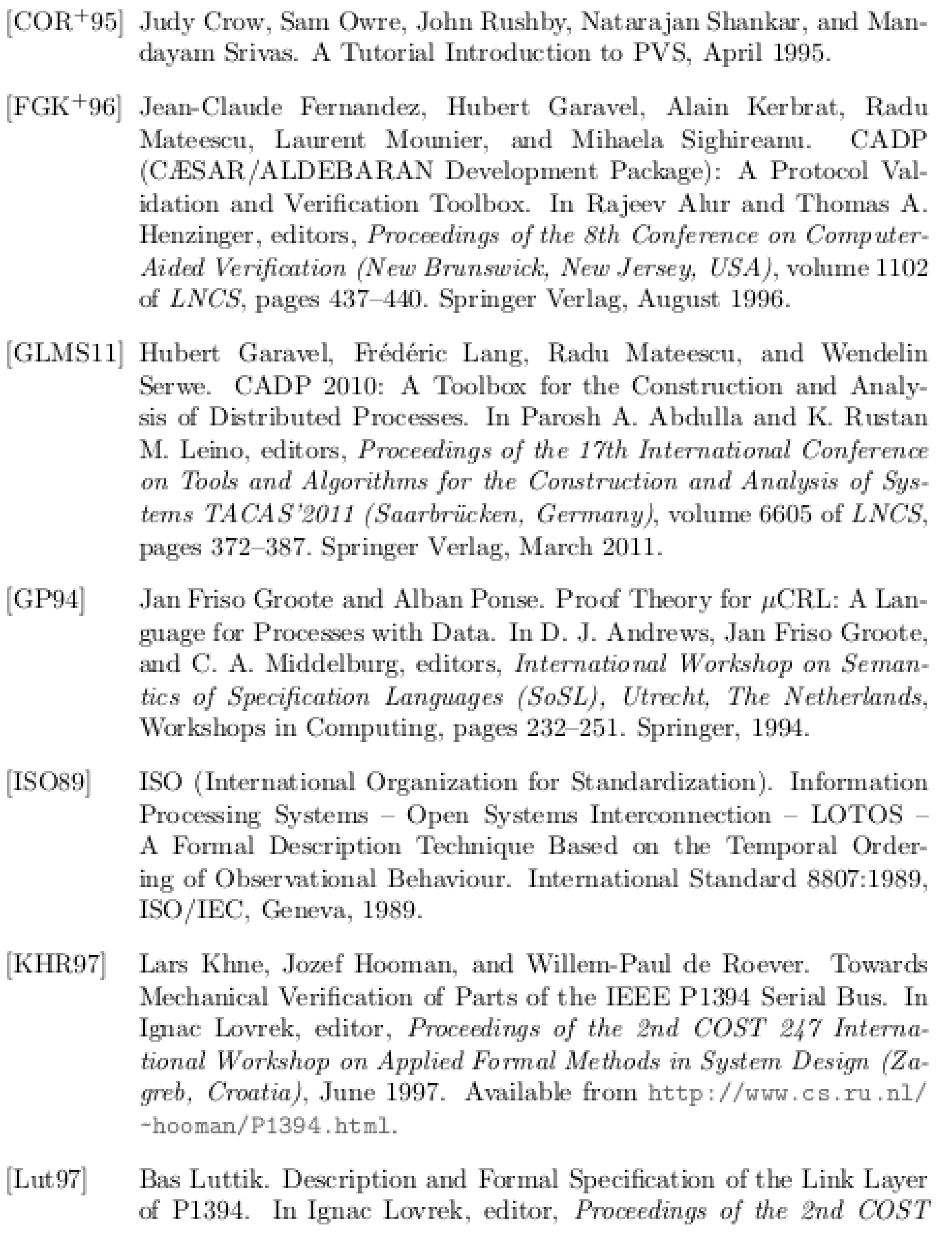
Figure 1: Serial Bus architecture.

Specification and analysis, using various formal methods, of the asynchronous mode of the Link Layer protocol of the IEEE Standard 1394 "Firewire" high-speed serial bus. Two problems found:

- a missing handling of pending requests discovered independently using PVS and μCRL
- a deadlock discovered using LOTOS and CADP in only one person.month without prior knowledge of the protocol

During the next decade, other IEEE 1394 protocols (root contention, tree identity, leader election, etc.) have been intensely scrutinized

[1997]



[COR+95] Judy Crow, Sam Owre, John Rushby, Natarajan Shankar, and Mandayam Srivas. A Tutorial Introduction to PVS, April 1995.

[FGK+96] Jean-Claude Fernandez, Hubert Garavel, Alain Kerbrat, Radu Mateescu, Laurent Mounier, and Mihaela Sighireanu. CADP (CÆSAR/ALDEBARAN Development Package): A Protocol Validation and Verification Toolbox. In Rajeev Alur and Thomas A. Henzinger, editors, *Proceedings of the 8th Conference on Computer-Aided Verification (New Brunswick, New Jersey, USA)*, volume 1102 of *LNCS*, pages 437–440. Springer Verlag, August 1996.

[GLMS11] Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. CADP 2010: A Toolbox for the Construction and Analysis of Distributed Processes. In Parosh A. Abdulla and K. Rustan M. Leino, editors, *Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS'2011 (Saarbrücken, Germany)*, volume 6605 of *LNCS*, pages 372–387. Springer Verlag, March 2011.

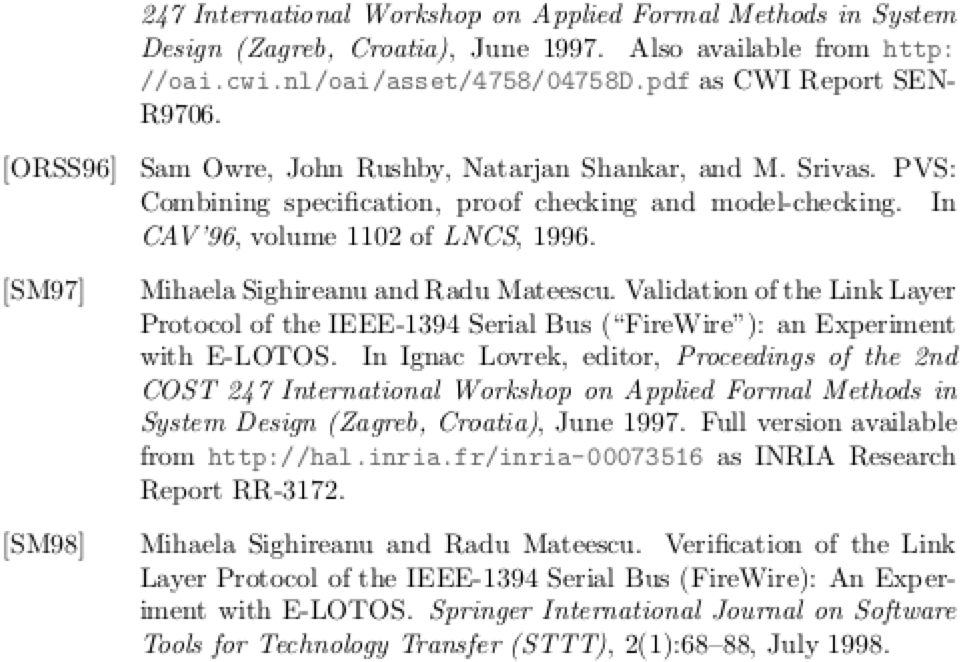
[GP94] Jan Friso Groote and Alban Ponse. Proof Theory for μCRL: A Language for Processes with Data. In D. J. Andrews, Jan Friso Groote, and C. A. Middelburg, editors, *International Workshop on Semantics of Specification Languages (SoSL), Utrecht, The Netherlands*, Workshops in Computing, pages 232–251. Springer, 1994.

[ISO89] ISO (International Organization for Standardization). Information Processing Systems – Open Systems Interconnection – LOTOS – A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour. International Standard 8807:1989, ISO/IEC, Geneva, 1989.

[KHR97] Lars Khne, Jozef Hooman, and Willem-Paul de Roever. Towards Mechanical Verification of Parts of the IEEE P1394 Serial Bus. In Ignac Lovrek, editor, *Proceedings of the 2nd COST 247 International Workshop on Applied Formal Methods in System Design (Zagreb, Croatia)*, June 1997. Available from `http://www.cs.ru.nl/~hooman/P1394.html`.

[Lut97] Bas Luttik. Description and Formal Specification of the Link Layer of P1394. In Ignac Lovrek, editor, *Proceedings of the 2nd COST*

[1997]



247 International Workshop on Applied Formal Methods in System Design (Zagreb, Croatia), June 1997. Also available from http://oai.cwi.nl/oai/asset/4758/04758D.pdf as CWI Report SEN-R9706.

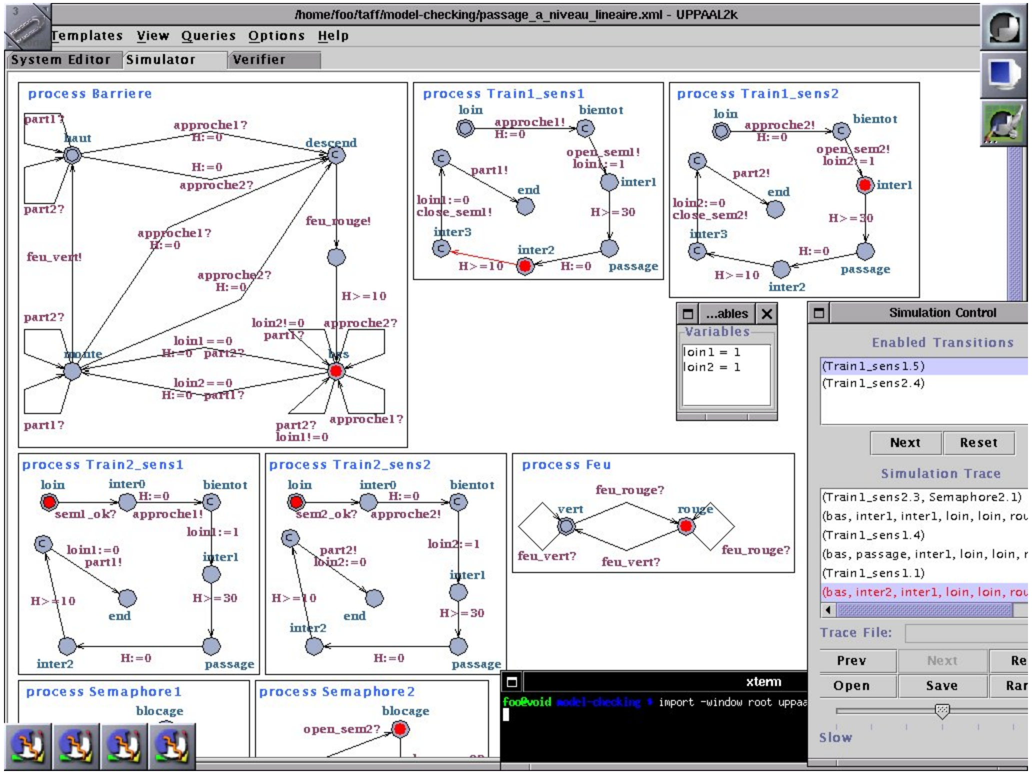
[ORSS96] Sam Owre, John Rushby, Natarjan Shankar, and M. Srivas. PVS: Combining specification, proof checking and model-checking. In *CAV'96*, volume 1102 of *LNCS*, 1996.

[SM97] Mihaela Sighireanu and Radu Mateescu. Validation of the Link Layer Protocol of the IEEE-1394 Serial Bus ("FireWire"): an Experiment with E-LOTOS. In Ignac Lovrek, editor, *Proceedings of the 2nd COST 247 International Workshop on Applied Formal Methods in System Design (Zagreb, Croatia)*, June 1997. Full version available from http://hal.inria.fr/inria-00073516 as INRIA Research Report RR-3172.

[SM98] Mihaela Sighireanu and Radu Mateescu. Verification of the Link Layer Protocol of the IEEE-1394 Serial Bus (FireWire): An Experiment with E-LOTOS. *Springer International Journal on Software Tools for Technology Transfer (STTT)*, 2(1):68–88, July 1998.

1998

[Bowman-Faconti-Katoen-Latella-Massink]
[Lindahl-Pettersson-Yi]
[Tripakis-Yovine]



```
loin1 = 1
loin2 = 1
```

```
(Train1_sens1.5)
(Train1_sens2.4)
```

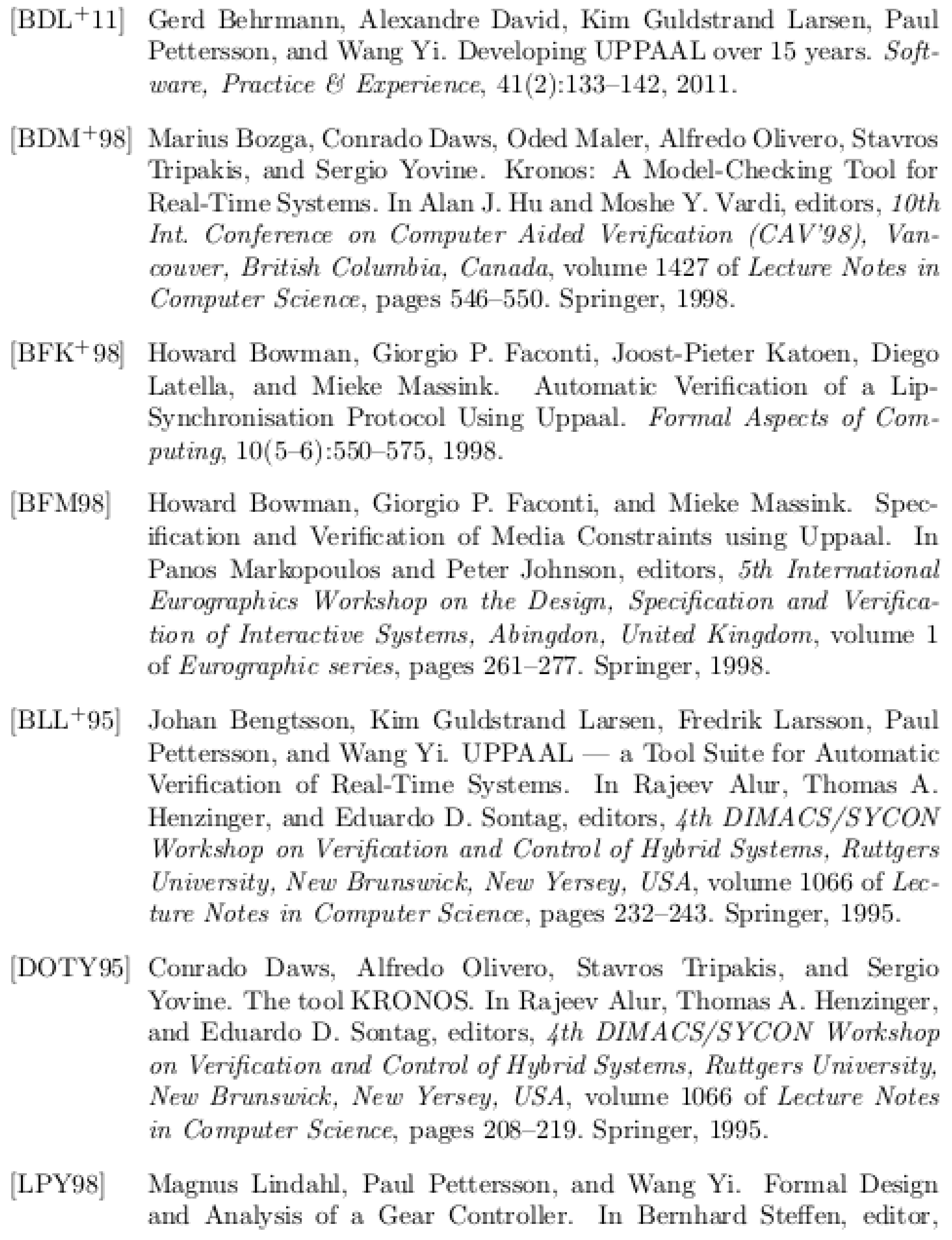
```
(Train1_sens2.3, Semaphore2.1)
(bas, inter1, inter1, loin, loin, rou
(Train1_sens1.4)
(bas, passage, inter1, loin, loin, r
(Train1_sens1.1)
(bas, inter2, inter1, loin, loin, rou
```

```
foo@void model-checking $ import -window root uppaa
```

Automated verification of several real-time protocols using:

- Kronos [Daws-Olivero-Tripakis-Yovine-Bozga-Maler]
- Uppaal [Bengtsson-Larsen-Larsson-Pettersson-Yi]

[1998]



[BDL+11] Gerd Behrmann, Alexandre David, Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. Developing UPPAAL over 15 years. *Software, Practice & Experience*, 41(2):133–142, 2011.

[BDM+98] Marius Bozga, Conrado Daws, Oded Maler, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. Kronos: A Model-Checking Tool for Real-Time Systems. In Alan J. Hu and Moshe Y. Vardi, editors, *10th Int. Conference on Computer Aided Verification (CAV'98), Vancouver, British Columbia, Canada*, volume 1427 of *Lecture Notes in Computer Science*, pages 546–550. Springer, 1998.

[BFK+98] Howard Bowman, Giorgio P. Faconti, Joost-Pieter Katoen, Diego Latella, and Mieke Massink. Automatic Verification of a Lip-Synchronisation Protocol Using Uppaal. *Formal Aspects of Computing*, 10(5–6):550–575, 1998.

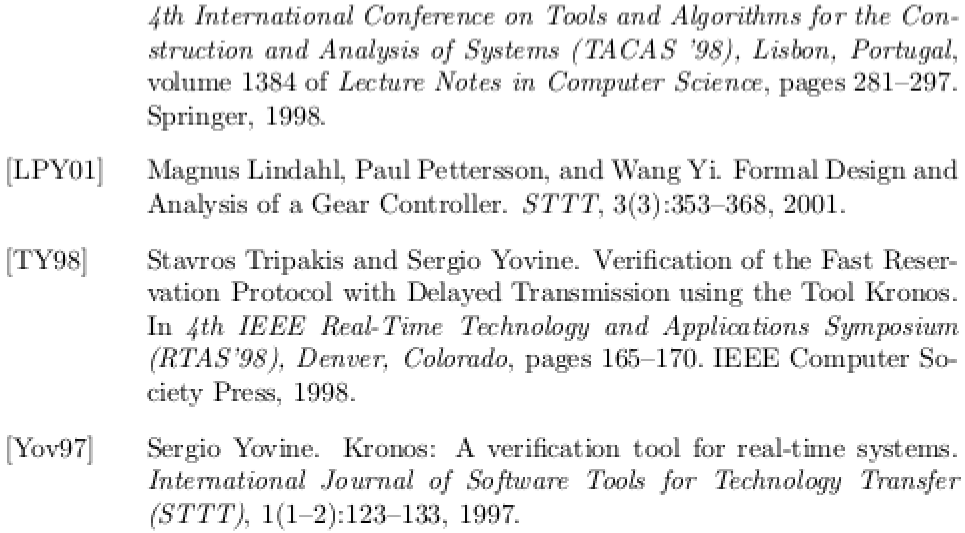
[BFM98] Howard Bowman, Giorgio P. Faconti, and Mieke Massink. Specification and Verification of Media Constraints using Uppaal. In Panos Markopoulos and Peter Johnson, editors, *5th International Eurographics Workshop on the Design, Specification and Verification of Interactive Systems, Abingdon, United Kingdom*, volume 1 of *Eurographic series*, pages 261–277. Springer, 1998.

[BLL+95] Johan Bengtsson, Kim Guldstrand Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL — a Tool Suite for Automatic Verification of Real-Time Systems. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *4th DIMACS/SYCON Workshop on Verification and Control of Hybrid Systems, Ruttgers University, New Brunswick, New Yersey, USA*, volume 1066 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 1995.

[DOTY95] Conrado Daws, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. The tool KRONOS. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *4th DIMACS/SYCON Workshop on Verification and Control of Hybrid Systems, Ruttgers University, New Brunswick, New Yersey, USA*, volume 1066 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 1995.

[LPY98] Magnus Lindahl, Paul Pettersson, and Wang Yi. Formal Design and Analysis of a Gear Controller. In Bernhard Steffen, editor,

[1998]



4th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '98), Lisbon, Portugal, volume 1384 of *Lecture Notes in Computer Science*, pages 281–297. Springer, 1998.

[LPY01] Magnus Lindahl, Paul Pettersson, and Wang Yi. Formal Design and Analysis of a Gear Controller. *STTT*, 3(3):353–368, 2001.

[TY98] Stavros Tripakis and Sergio Yovine. Verification of the Fast Reservation Protocol with Delayed Transmission using the Tool Kronos. In *4th IEEE Real-Time Technology and Applications Symposium (RTAS'98), Denver, Colorado*, pages 165–170. IEEE Computer Society Press, 1998.

[Yov97] Sergio Yovine. Kronos: A verification tool for real-time systems. *International Journal of Software Tools for Technology Transfer (STTT)*, 1(1–2):123–133, 1997.

1999

[Pfeifer-Schwier-von Henke-Rushby-Sorea-Steiner]

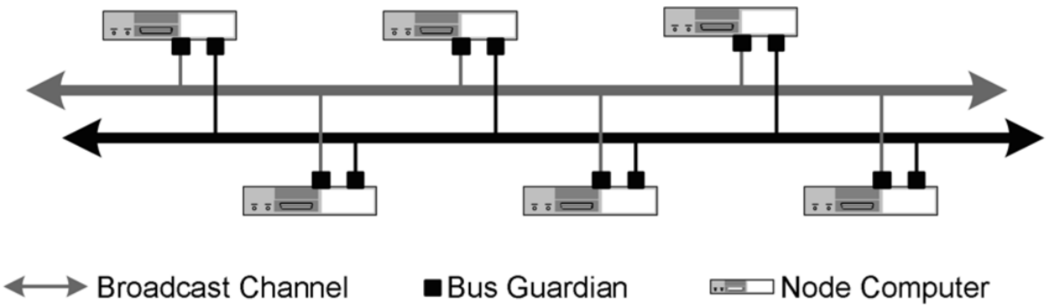


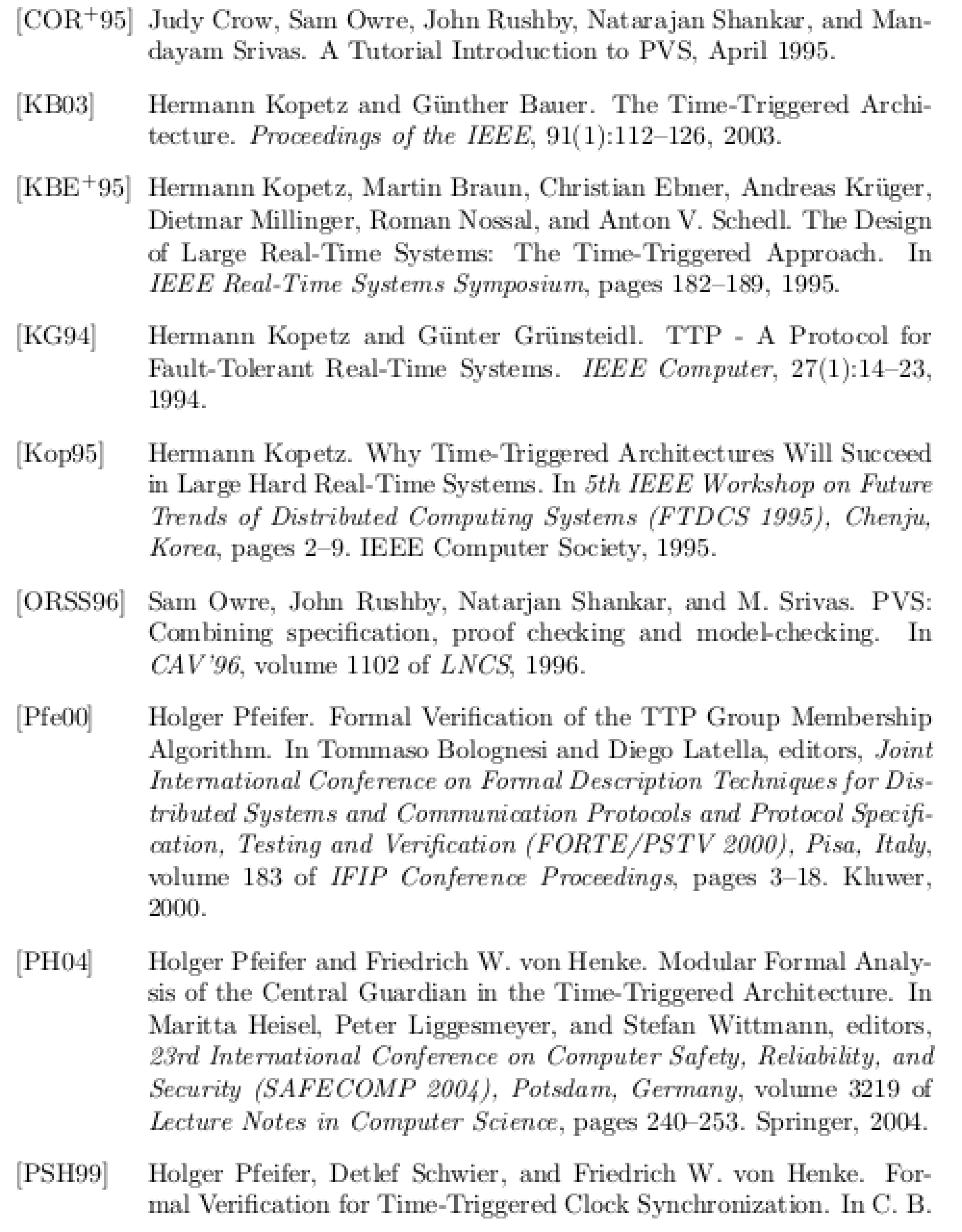
Fig. 5. Topology of TTA-bus

TTA (Time-Triggered Architecture) is a communication bus infrastructure guaranteeing dependability, predictability, and real-time requirements [Kopetz-Bauer-Braun-Gründsteidl-etc]

TTA and similar architectures are used for distributed-control safety-critical applications in automotive, aerospace, railways, industrial automation and process control, medical systems, etc.

Formal verification using PVS of several key protocols of TTA

[1999]



[COR+95] Judy Crow, Sam Owre, John Rushby, Natarajan Shankar, and Mandayam Srivas. A Tutorial Introduction to PVS, April 1995.

[KB03] Hermann Kopetz and Günther Bauer. The Time-Triggered Architecture. *Proceedings of the IEEE*, 91(1):112–126, 2003.

[KBE+95] Hermann Kopetz, Martin Braun, Christian Ebner, Andreas Krüger, Dietmar Millinger, Roman Nossal, and Anton V. Schedl. The Design of Large Real-Time Systems: The Time-Triggered Approach. In *IEEE Real-Time Systems Symposium*, pages 182–189, 1995.

[KG94] Hermann Kopetz and Günter Grünsteidl. TTP - A Protocol for Fault-Tolerant Real-Time Systems. *IEEE Computer*, 27(1):14–23, 1994.

[Kop95] Hermann Kopetz. Why Time-Triggered Architectures Will Succeed in Large Hard Real-Time Systems. In *5th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 1995), Chenju, Korea*, pages 2–9. IEEE Computer Society, 1995.

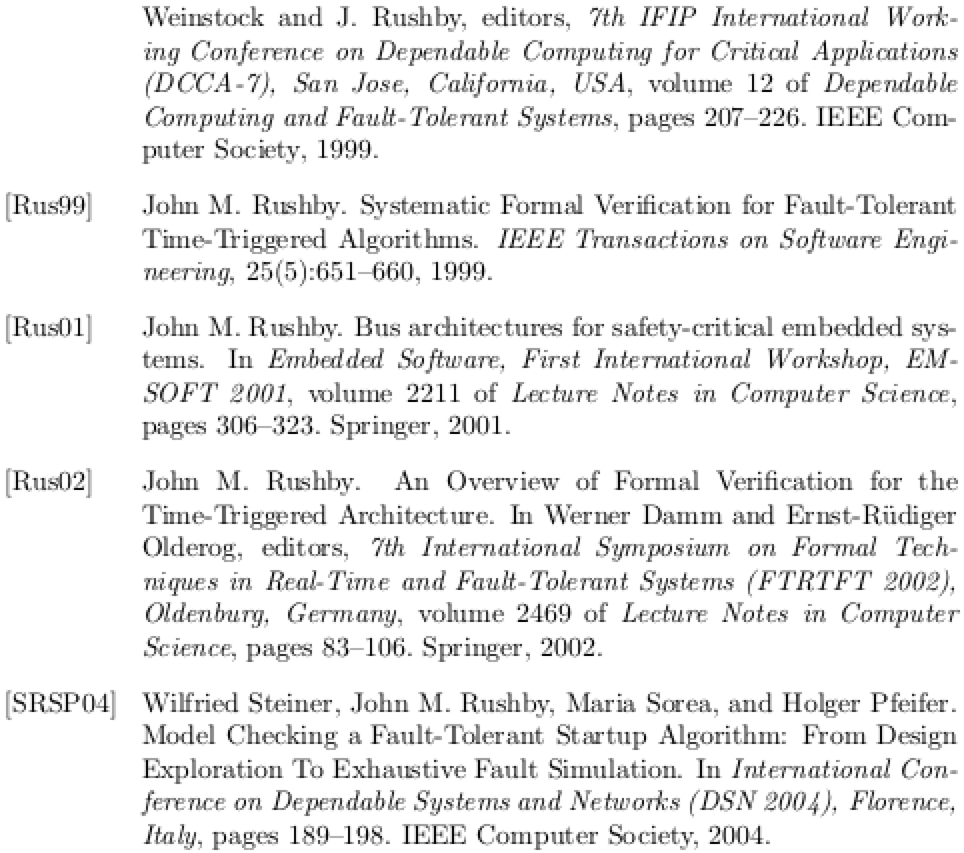
[ORSS96] Sam Owre, John Rushby, Natarjan Shankar, and M. Srivas. PVS: Combining specification, proof checking and model-checking. In *CAV'96*, volume 1102 of *LNCS*, 1996.

[Pfe00] Holger Pfeifer. Formal Verification of the TTP Group Membership Algorithm. In Tommaso Bolognesi and Diego Latella, editors, *Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols and Protocol Specification, Testing and Verification (FORTE/PSTV 2000), Pisa, Italy*, volume 183 of *IFIP Conference Proceedings*, pages 3–18. Kluwer, 2000.

[PH04] Holger Pfeifer and Friedrich W. von Henke. Modular Formal Analysis of the Central Guardian in the Time-Triggered Architecture. In Maritta Heisel, Peter Liggesmeyer, and Stefan Wittmann, editors, *23rd International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2004), Potsdam, Germany*, volume 3219 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2004.

[PSH99] Holger Pfeifer, Detlef Schwier, and Friedrich W. von Henke. Formal Verification for Time-Triggered Clock Synchronization. In C. B.

[1999]



Weinstock and J. Rushby, editors, *7th IFIP International Working Conference on Dependable Computing for Critical Applications (DCCA-7), San Jose, California, USA*, volume 12 of *Dependable Computing and Fault-Tolerant Systems*, pages 207–226. IEEE Computer Society, 1999.

[Rus99] John M. Rushby. Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms. *IEEE Transactions on Software Engineering*, 25(5):651–660, 1999.

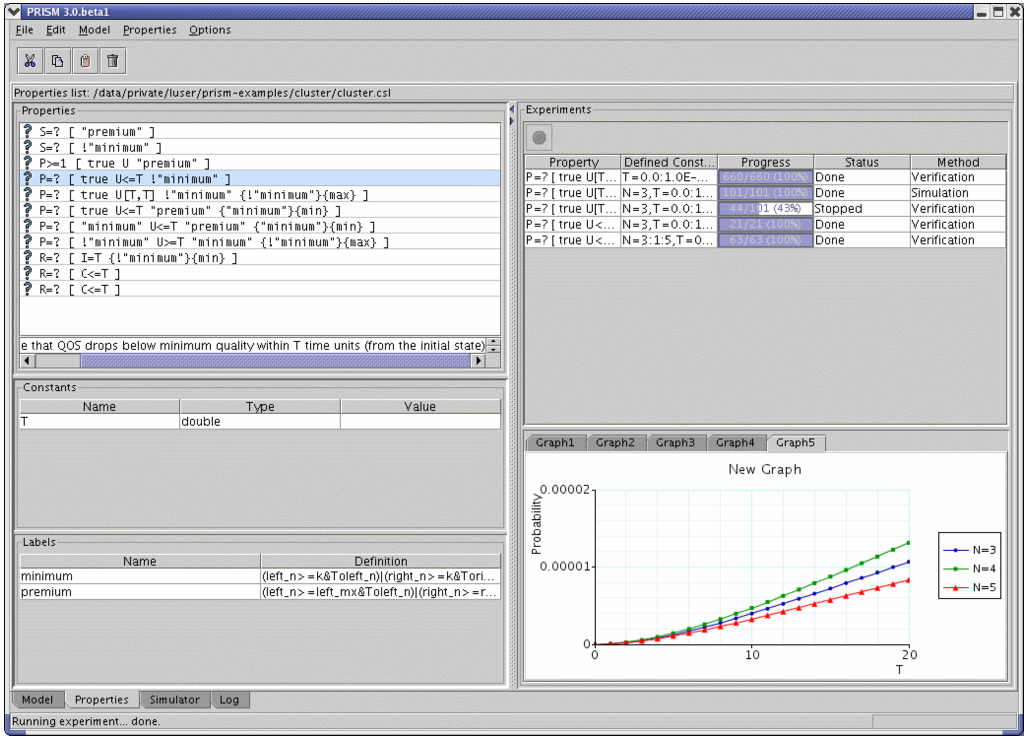
[Rus01] John M. Rushby. Bus architectures for safety-critical embedded systems. In *Embedded Software, First International Workshop, EMSOFT 2001*, volume 2211 of *Lecture Notes in Computer Science*, pages 306–323. Springer, 2001.

[Rus02] John M. Rushby. An Overview of Formal Verification for the Time-Triggered Architecture. In Werner Damm and Ernst-Rüdiger Olderog, editors, *7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT 2002), Oldenburg, Germany*, volume 2469 of *Lecture Notes in Computer Science*, pages 83–106. Springer, 2002.

[SRSP04] Wilfried Steiner, John M. Rushby, Maria Sorea, and Holger Pfeifer. Model Checking a Fault-Tolerant Startup Algorithm: From Design Exploration To Exhaustive Fault Simulation. In *International Conference on Dependable Systems and Networks (DSN 2004), Florence, Italy*, pages 189–198. IEEE Computer Society, 2004.

2000

[Kwiatkowska-Norman-Parker-Segala]


PRISM 3.0.beta1
File Edit Model Properties Options
Properties
Experiments
Graph1 Graph2 Graph3 Graph4 Graph5
New Graph
Probability
0.00002
0.00001
0
0
10
20
T
N=3
N=4
N=5
Model Properties Simulator Log

Properties list: /data/private/luser/prism-examples/cluster/cluster.csl

```
S=? [ "premium" ]
S=? [ !"minimum" ]
P>=1 [ true U "premium" ]
P=? [ true U<=T !"minimum" ]
P=? [ true U[T,T] !"minimum" {!"minimum"}{max} ]
P=? [ true U<=T "premium" {"minimum"}{min} ]
P=? [ "minimum" U<=T "premium" {"minimum"}{min} ]
P=? [ !"minimum" U>=T "minimum" {!"minimum"}{max} ]
R=? [ I=T {!"minimum"}{min} ]
R=? [ C<=T ]
R=? [ C<=T ]
```

e that QOS drops below minimum quality within T time units (from the initial state)

Constants

Name	Type	Value
T	double	

Labels

Name	Definition
minimum	(left_n>=k&Toleft_n)|(right_n>=k&Tori...
premium	(left_n>=left_mx&Toleft_n)|(right_n>=r...

Experiments

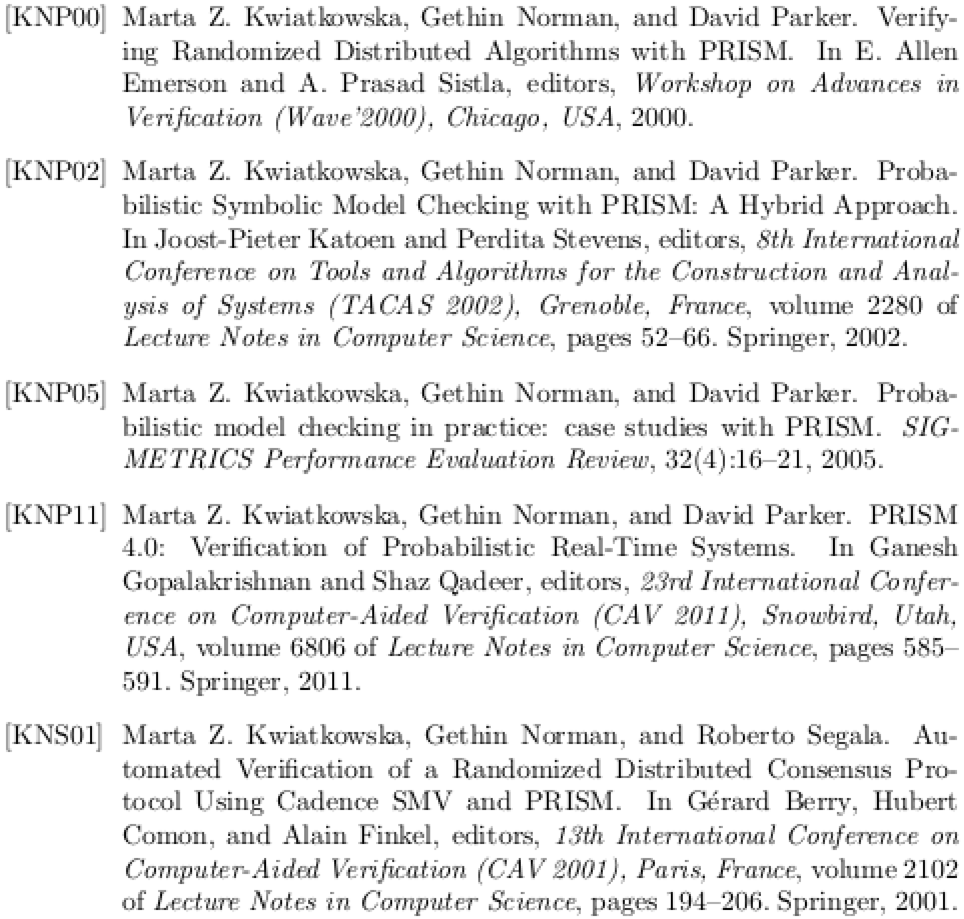
Property	Defined Const...	Progress	Status	Method
P=? [true U[T...	T=0.0:1.0E-...	660/660 (100%)	Done	Verification
P=? [true U[T...	N=3,T=0.0:1...	101/101 (100%)	Done	Simulation
P=? [true U[T...	N=3,T=0.0:1...	44/101 (43%)	Stopped	Verification
P=? [true U<...	N=3,T=0.0:1...	21/21 (100%)	Done	Verification
P=? [true U<...	N=3:1:5,T=0...	63/63 (100%)	Done	Verification

Running experiment... done.

Automated validation of several randomized distributed algorithms (taken from the literature) using the PRISM probabilistic model checker

PRISM has been used to analyze case studies in many different application domains

[2000]



[KNP00] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Verifying Randomized Distributed Algorithms with PRISM. In E. Allen Emerson and A. Prasad Sistla, editors, *Workshop on Advances in Verification (Wave'2000), Chicago, USA*, 2000.

[KNP02] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Probabilistic Symbolic Model Checking with PRISM: A Hybrid Approach. In Joost-Pieter Katoen and Perdita Stevens, editors, *8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2002), Grenoble, France*, volume 2280 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2002.

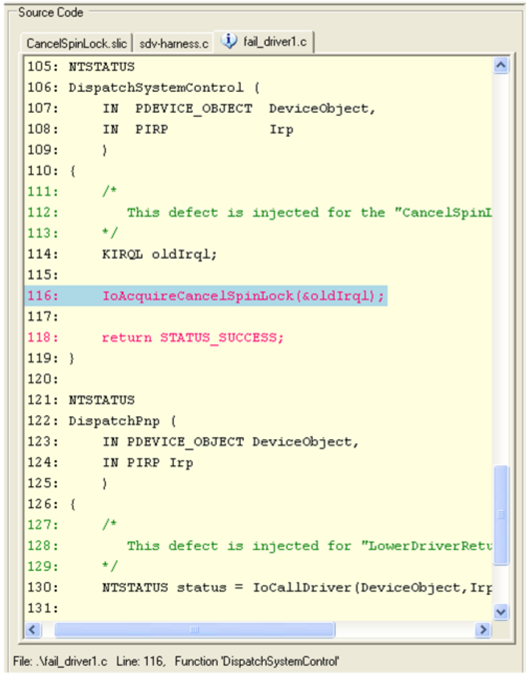
[KNP05] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Probabilistic model checking in practice: case studies with PRISM. *SIGMETRICS Performance Evaluation Review*, 32(4):16–21, 2005.

[KNP11] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *23rd International Conference on Computer-Aided Verification (CAV 2011), Snowbird, Utah, USA*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

[KNS01] Marta Z. Kwiatkowska, Gethin Norman, and Roberto Segala. Automated Verification of a Randomized Distributed Consensus Protocol Using Cadence SMV and PRISM. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *13th International Conference on Computer-Aided Verification (CAV 2001), Paris, France*, volume 2102 of *Lecture Notes in Computer Science*, pages 194–206. Springer, 2001.

2001

[Ball-Bounimova-Chaki-Kumar-Levin-Lichtenberg-Rajamani]



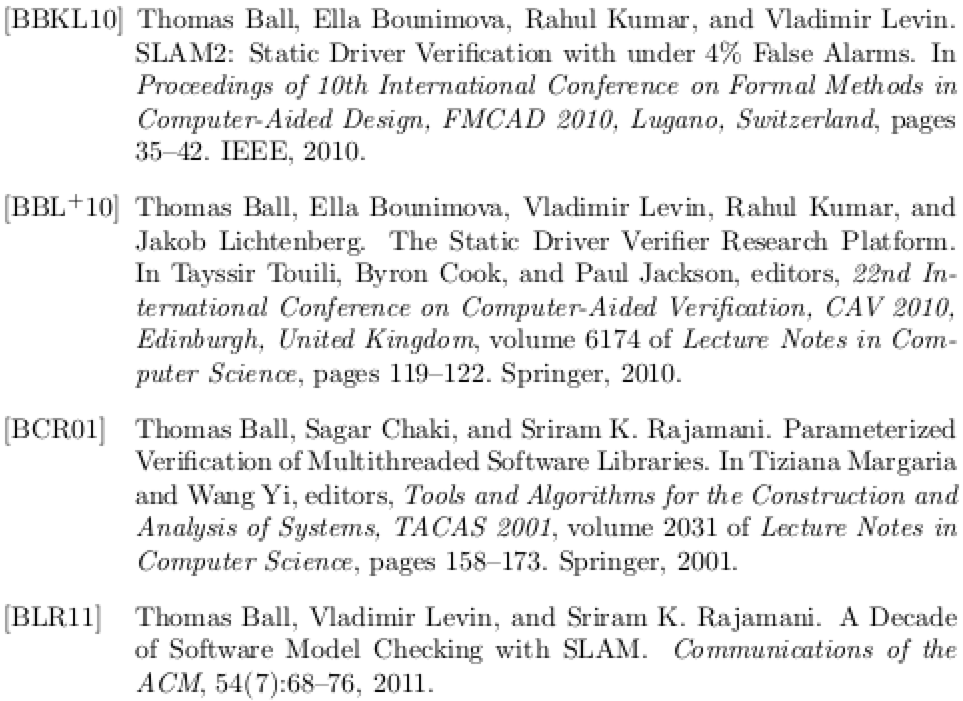
```
105: NTSTATUS
106: DispatchSystemControl (
107:     IN  PDEVICE_OBJECT  DeviceObject,
108:     IN  PIRP            Irp
109:     )
110: {
111:     /*
112:        This defect is injected for the "CancelSpinL
113:     */
114:     KIRQL oldIrql;
115:
116:     IoAcquireCancelSpinLock(&oldIrql);
117:
118:     return STATUS_SUCCESS;
119: }
120:
121: NTSTATUS
122: DispatchPnp (
123:     IN PDEVICE_OBJECT DeviceObject,
124:     IN PIRP Irp
125:     )
126: {
127:     /*
128:        This defect is injected for "LowerDriverRetu
129:     */
130:     NTSTATUS status = IoCallDriver(DeviceObject,Irp
131:
```

File: .\fail_driver1.c Line: 116, Function 'DispatchSystemControl'

Development of a verification platform (based on static analysis and symbolic model checking) for analyzing the source code of Microsoft Windows drivers (and more generally any source C code)

Check whether the invocations of API (Application Programming Interfaces) primitives obey rules for proper use

[2001]



[BBKL10] Thomas Ball, Ella Bounimova, Rahul Kumar, and Vladimir Levin. SLAM2: Static Driver Verification with under 4% False Alarms. In *Proceedings of 10th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2010, Lugano, Switzerland*, pages 35–42. IEEE, 2010.

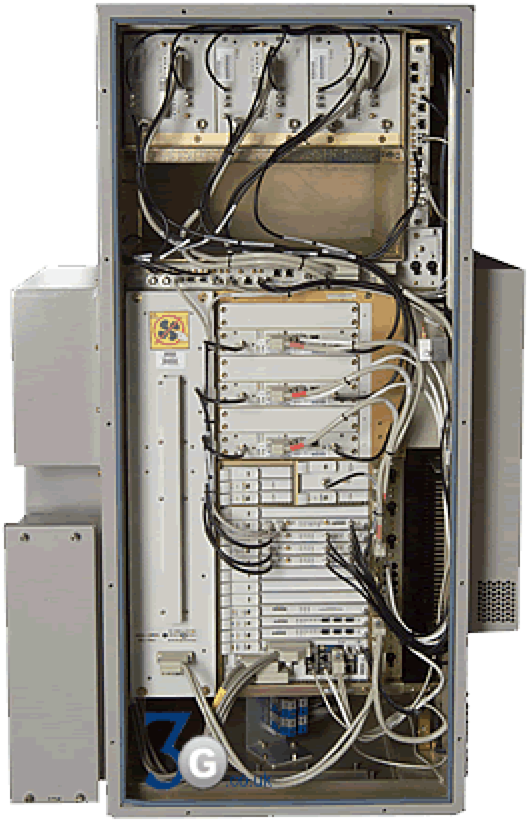
[BBL$^+$10] Thomas Ball, Ella Bounimova, Vladimir Levin, Rahul Kumar, and Jakob Lichtenberg. The Static Driver Verifier Research Platform. In Tayssir Touili, Byron Cook, and Paul Jackson, editors, *22nd International Conference on Computer-Aided Verification, CAV 2010, Edinburgh, United Kingdom*, volume 6174 of *Lecture Notes in Computer Science*, pages 119–122. Springer, 2010.

[BCR01] Thomas Ball, Sagar Chaki, and Sriram K. Rajamani. Parameterized Verification of Multithreaded Software Libraries. In Tiziana Margaria and Wang Yi, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2001*, volume 2031 of *Lecture Notes in Computer Science*, pages 158–173. Springer, 2001.

[BLR11] Thomas Ball, Vladimir Levin, and Sriram K. Rajamani. A Decade of Software Model Checking with SLAM. *Communications of the ACM*, 54(7):68–76, 2011.

2002

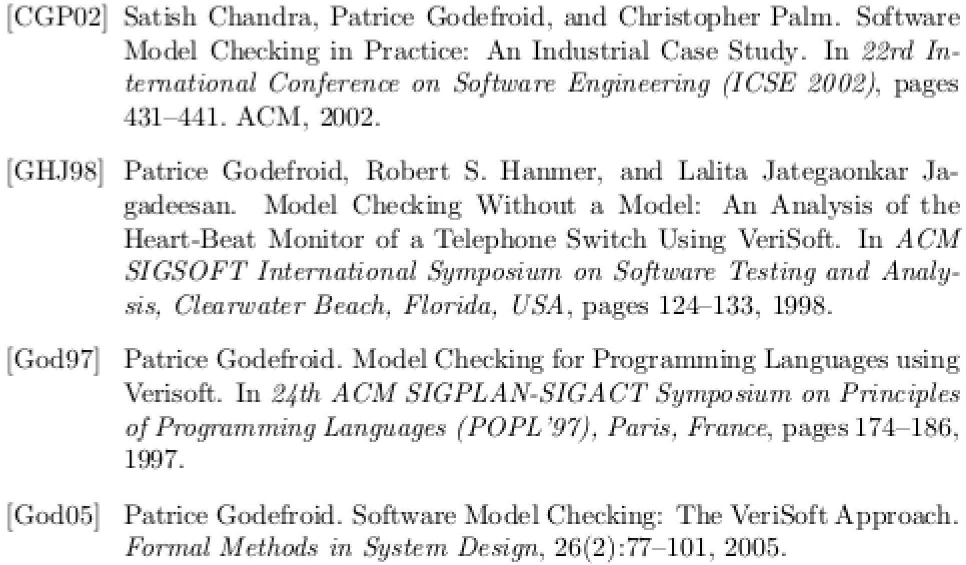
[Chandra-Godefroid-Palm]



Automated analysis of Lucent's CDMA base station call-processing software library (100,000s lines of C/C++ code) using the VeriSoft tool for systematic state space exploration

Detection of several critical bugs

[2002]



[CGP02] Satish Chandra, Patrice Godefroid, and Christopher Palm. Software Model Checking in Practice: An Industrial Case Study. In *22rd International Conference on Software Engineering (ICSE 2002)*, pages 431–441. ACM, 2002.

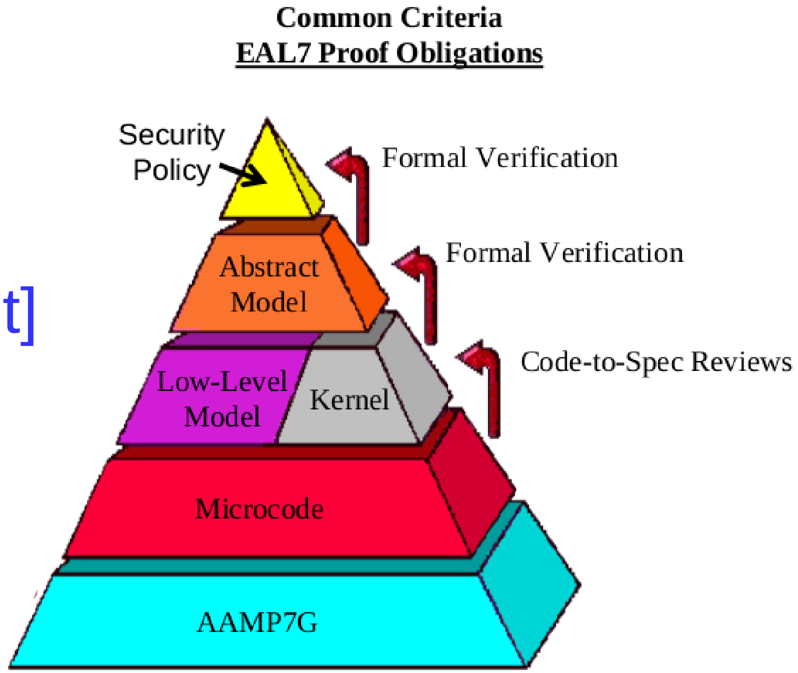
[GHJ98] Patrice Godefroid, Robert S. Hanmer, and Lalita Jategaonkar Jagadeesan. Model Checking Without a Model: An Analysis of the Heart-Beat Monitor of a Telephone Switch Using VeriSoft. In *ACM SIGSOFT International Symposium on Software Testing and Analysis, Clearwater Beach, Florida, USA*, pages 124–133, 1998.

[God97] Patrice Godefroid. Model Checking for Programming Languages using Verisoft. In *24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'97), Paris, France*, pages 174–186, 1997.

[God05] Patrice Godefroid. Software Model Checking: The VeriSoft Approach. *Formal Methods in System Design*, 26(2):77–101, 2005.

2003

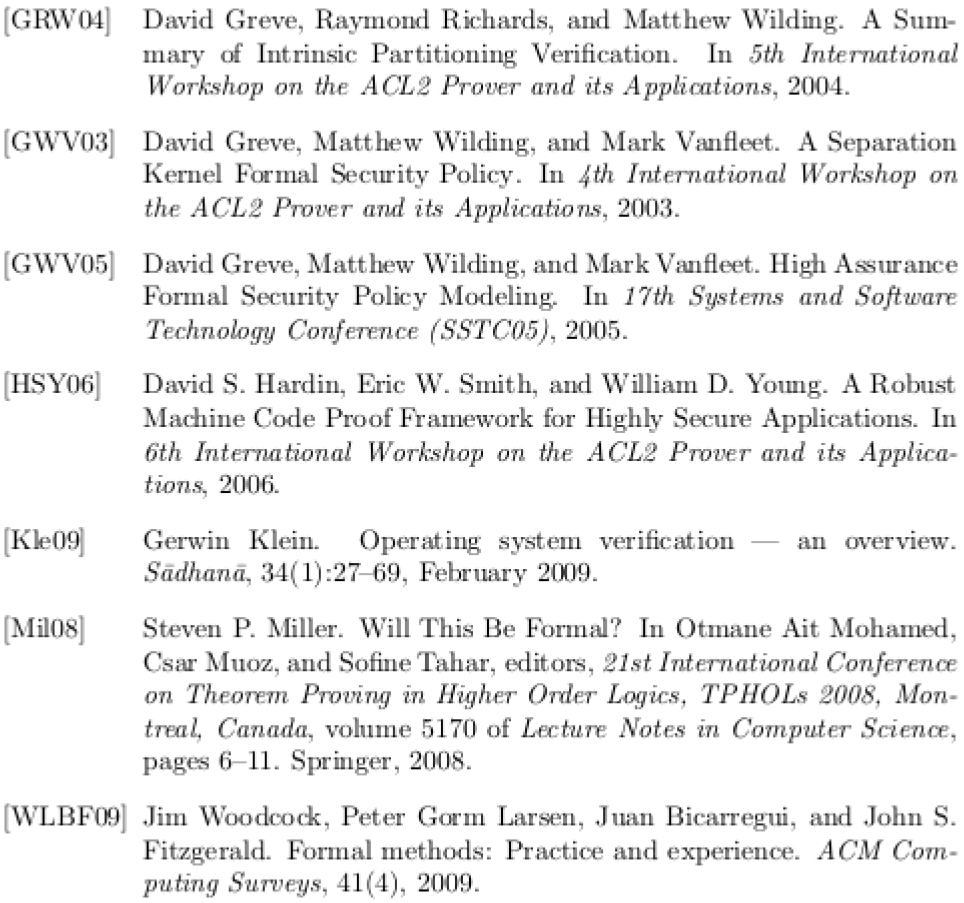
[Greve-Richards-Wilding-Vanfleet]
[Hardin-Smith-Young]



Formal proof using the ACL2 theorem prover that the microcode of the Rockwell Collins AAMP7 microprocessor respects a security policy corresponding to a static separation kernel

The microprocessor received a MILS Certificate from NSA to concurrently process Unclassified through Top Secret codeword information

[2003]



[GRW04] David Greve, Raymond Richards, and Matthew Wilding. A Summary of Intrinsic Partitioning Verification. In *5th International Workshop on the ACL2 Prover and its Applications*, 2004.

[GWV03] David Greve, Matthew Wilding, and Mark Vanfleet. A Separation Kernel Formal Security Policy. In *4th International Workshop on the ACL2 Prover and its Applications*, 2003.

[GWV05] David Greve, Matthew Wilding, and Mark Vanfleet. High Assurance Formal Security Policy Modeling. In *17th Systems and Software Technology Conference (SSTC05)*, 2005.

[HSY06] David S. Hardin, Eric W. Smith, and William D. Young. A Robust Machine Code Proof Framework for Highly Secure Applications. In *6th International Workshop on the ACL2 Prover and its Applications*, 2006.

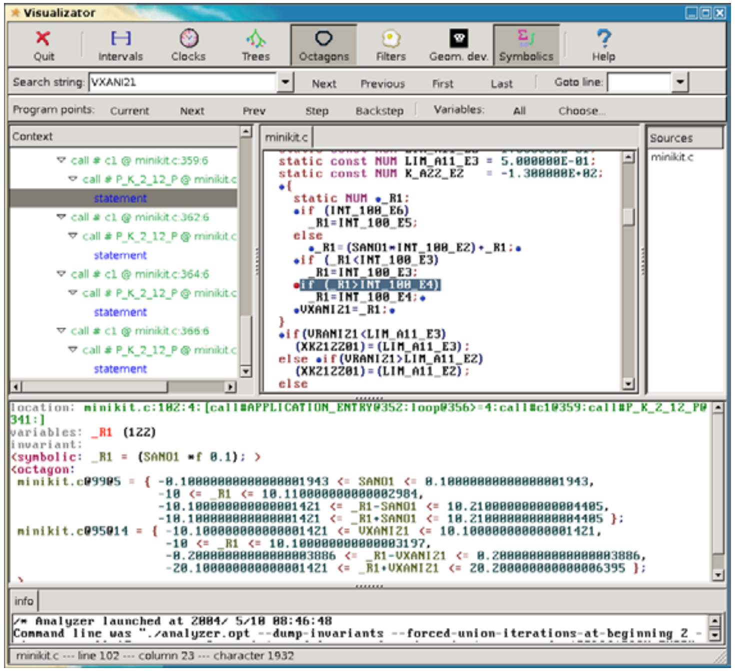
[Kle09] Gerwin Klein. Operating system verification — an overview. *Sādhanā*, 34(1):27–69, February 2009.

[Mil08] Steven P. Miller. Will This Be Formal? In Otmane Ait Mohamed, Csar Muoz, and Sofine Tahar, editors, *21st International Conference on Theorem Proving in Higher Order Logics, TPHOLs 2008, Montreal, Canada*, volume 5170 of *Lecture Notes in Computer Science*, pages 6–11. Springer, 2008.

[WLBF09] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John S. Fitzgerald. Formal methods: Practice and experience. *ACM Computing Surveys*, 41(4), 2009.

2004

[Blanchet-Cousot-Cousot-Feret-Mauborgne-Miné-Monniaux-Rival] [Delmas-Sourys]



```
static const NUM LIM_A11_E3 = 5.000000E-01;
static const NUM K_A22_E2   = -1.300000E+02;
•{
  static NUM •_R1;
  •if (INT_100_E6)
    _R1=INT_100_E5;
  else
    •_R1=(SAN01*INT_100_E2)+_R1;•
  •if (_R1<INT_100_E3)
    _R1=INT_100_E3;
  •if (_R1>INT_100_E4)
    _R1=INT_100_E4;•
  •VXANI21=_R1;•
}
•if(VRANI21<LIM_A11_E3)
  (XK212Z01)=(LIM_A11_E3);
else •if(VRANI21>LIM_A11_E2)
  (XK212Z01)=(LIM_A11_E2);
else
```

```
location: minikit.c:102:4:[call#APPLICATION_ENTRY@352:loop@356>=4:call#c1@359:call#P_K_2_12_P@
341:]
variables: _R1 (122)
invariant:
<symbolic: _R1 = (SAN01 *f 0.1); >
<octagon:
 minikit.c@9905 = { -0.1000000000000001943 <= SAN01 <= 0.1000000000000001943,
                    -10 <= _R1 <= 10.110000000000002984,
                    -10.100000000000001421 <= _R1-SAN01 <= 10.210000000000004405,
                    -10.100000000000001421 <= _R1+SAN01 <= 10.210000000000004405 };
 minikit.c@95014 = { -10.100000000000001421 <= VXANI21 <= 10.100000000000001421,
                     -10 <= _R1 <= 10.100000000000003197,
                     -0.2000000000000003886 <= _R1-VXANI21 <= 0.2000000000000003886,
                     -20.100000000000001421 <= _R1+VXANI21 <= 20.200000000000006395 };
>
```

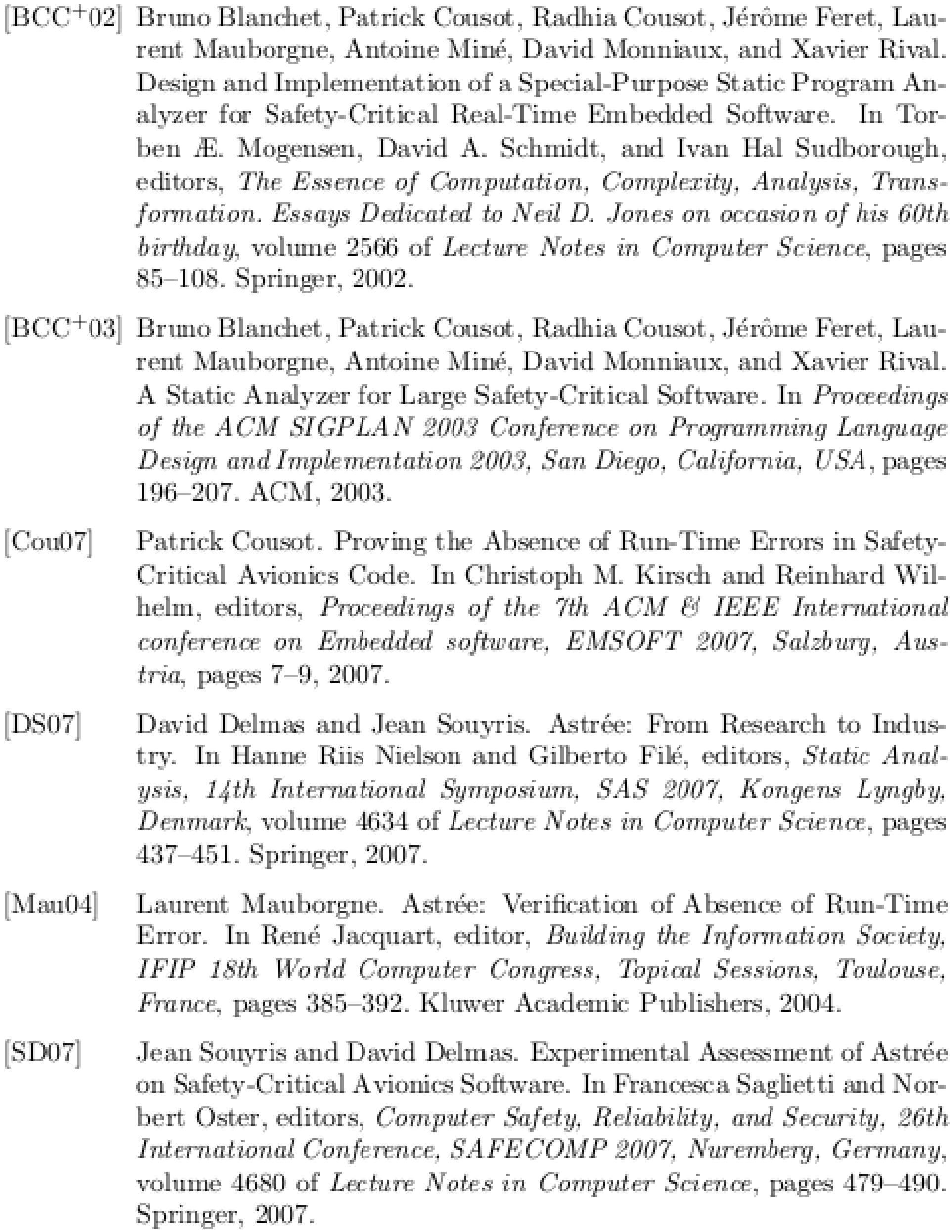
```
/* Analyzer launched at 2004/ 5/10 08:46:48
Command line was "./analyzer.opt --dump-invariants --forced-union-iterations-at-beginning 2 -
```

minikit.c --- line 102 --- column 23 --- character 1932

Proof, using the Astrée static analyzer based on abstract interpretation, of the absence of any run-time error in several safety-critical C programs of Airbus:

- primary flight-control software for the A340 fly-by-wire system
- electric flight-control codes for the A380 series

[2004]



[BCC+02] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. Design and Implementation of a Special-Purpose Static Program Analyzer for Safety-Critical Real-Time Embedded Software. In Torben Æ. Mogensen, David A. Schmidt, and Ivan Hal Sudborough, editors, *The Essence of Computation, Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones on occasion of his 60th birthday*, volume 2566 of *Lecture Notes in Computer Science*, pages 85–108. Springer, 2002.

[BCC+03] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. A Static Analyzer for Large Safety-Critical Software. In *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation 2003, San Diego, California, USA*, pages 196–207. ACM, 2003.

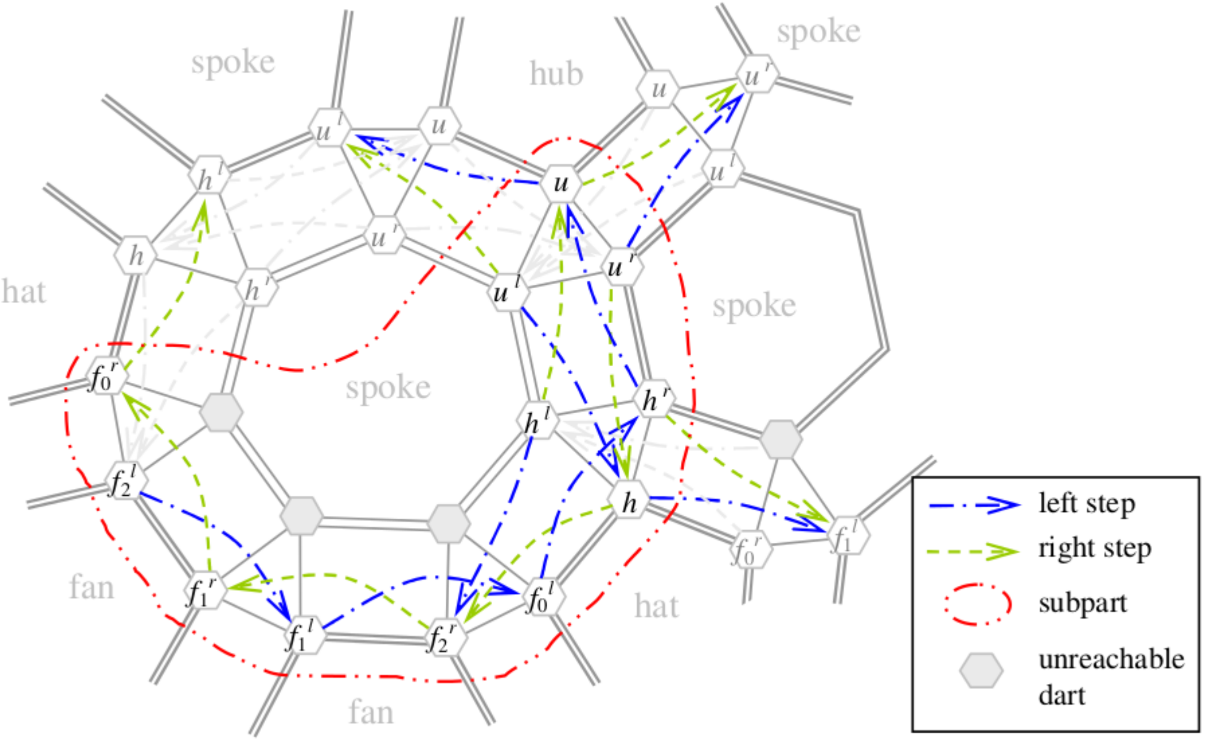
[Cou07] Patrick Cousot. Proving the Absence of Run-Time Errors in Safety-Critical Avionics Code. In Christoph M. Kirsch and Reinhard Wilhelm, editors, *Proceedings of the 7th ACM & IEEE International conference on Embedded software, EMSOFT 2007, Salzburg, Austria*, pages 7–9, 2007.

[DS07] David Delmas and Jean Souyris. Astrée: From Research to Industry. In Hanne Riis Nielson and Gilberto Filé, editors, *Static Analysis, 14th International Symposium, SAS 2007, Kongens Lyngby, Denmark*, volume 4634 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2007.

[Mau04] Laurent Mauborgne. Astrée: Verification of Absence of Run-Time Error. In René Jacquart, editor, *Building the Information Society, IFIP 18th World Computer Congress, Topical Sessions, Toulouse, France*, pages 385–392. Kluwer Academic Publishers, 2004.

[SD07] Jean Souyris and David Delmas. Experimental Assessment of Astrée on Safety-Critical Avionics Software. In Francesca Saglietti and Norbert Oster, editors, *Computer Safety, Reliability, and Security, 26th International Conference, SAFECOMP 2007, Nuremberg, Germany*, volume 4680 of *Lecture Notes in Computer Science*, pages 479–490. Springer, 2007.

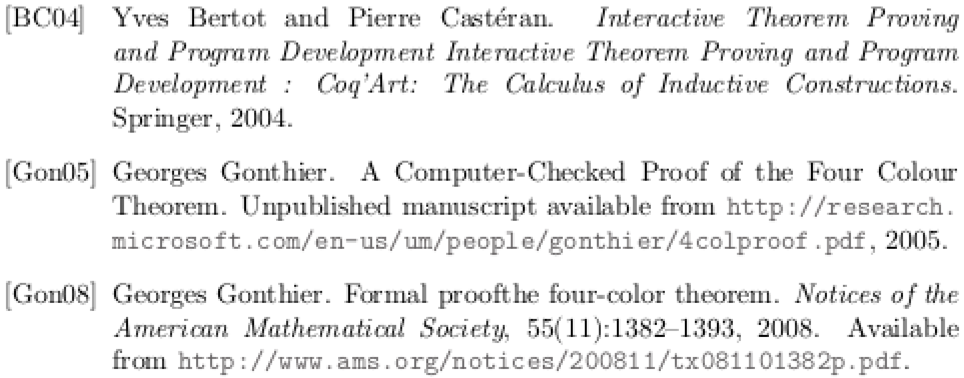
2005



[Gonthier]

Computer-checked proof using Coq of the "four color theorem", the second most famous unsolved problem in mathematics

[2005]



[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development Interactive Theorem Proving and Program Development : Coq'Art: The Calculus of Inductive Constructions.* Springer, 2004.

[Gon05] Georges Gonthier. A Computer-Checked Proof of the Four Colour Theorem. Unpublished manuscript available from `http://research.microsoft.com/en-us/um/people/gonthier/4colproof.pdf`, 2005.

[Gon08] Georges Gonthier. Formal proofthe four-color theorem. *Notices of the American Mathematical Society*, 55(11):1382–1393, 2008. Available from `http://www.ams.org/notices/200811/tx081101382p.pdf`.

2006

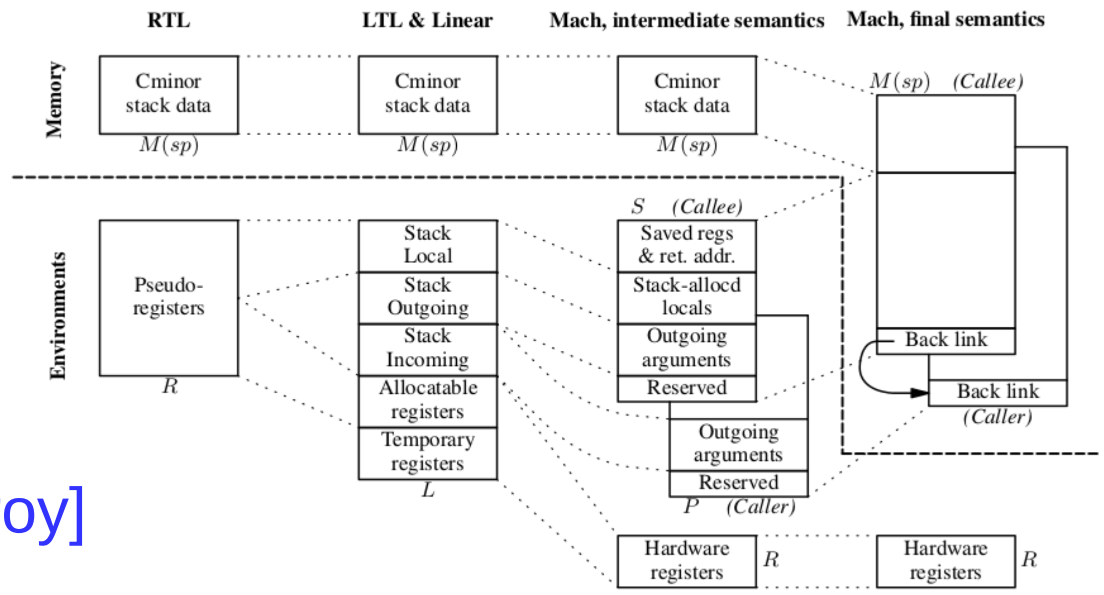
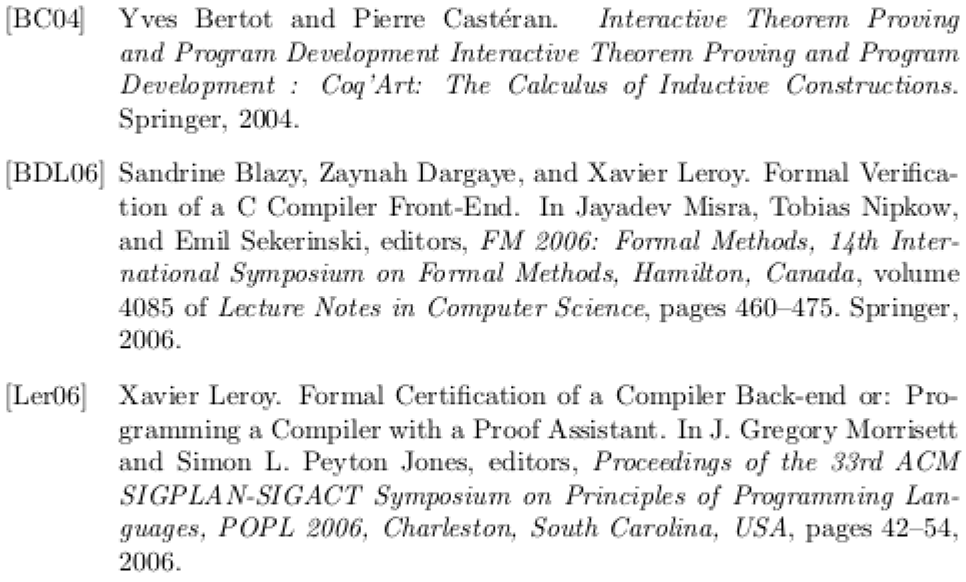


Figure 1. Overview of register allocation and introduction of activation records. For each intermediate language, the placement of function-local data is outlined, either in the memory-allocated activation record (top part) or in non memory-resident execution environments (bottom part).

[Blazy-Dargaye-Leroy]

Formal verification using Coq of a C compiler (front-end and back-end) with a realistic subset of the C language usable for critical embedded software

[2006]

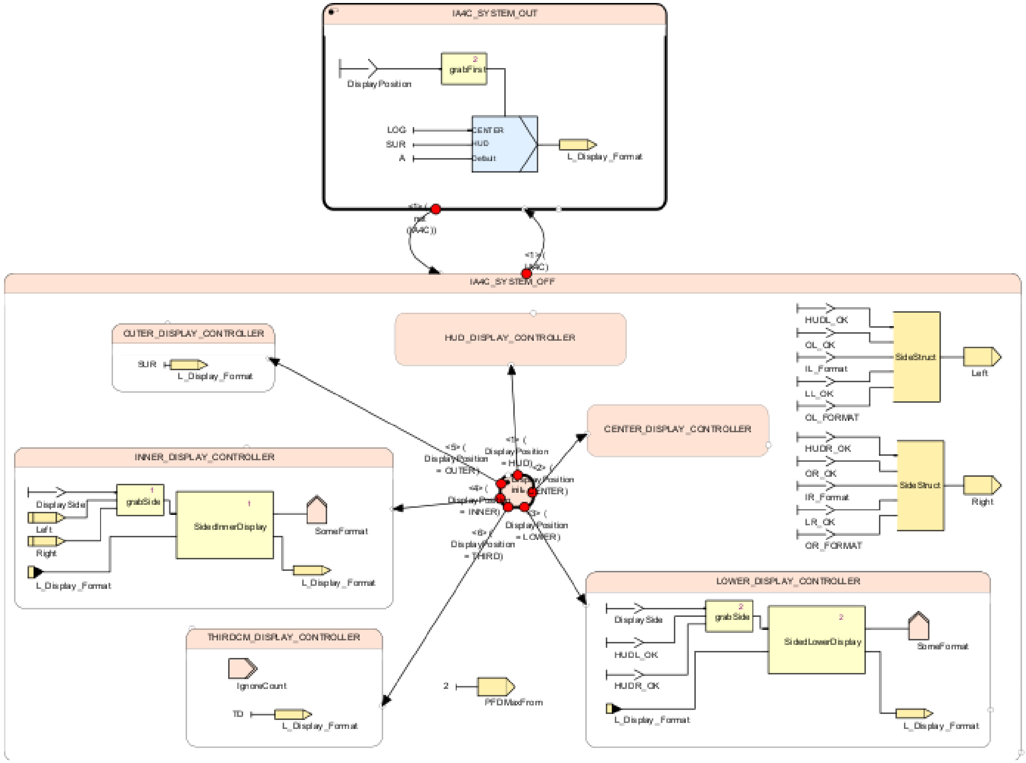


[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development Interactive Theorem Proving and Program Development : Coq'Art: The Calculus of Inductive Constructions.* Springer, 2004.

[BDL06] Sandrine Blazy, Zaynah Dargaye, and Xavier Leroy. Formal Verification of a C Compiler Front-End. In Jayadev Misra, Tobias Nipkow, and Emil Sekerinski, editors, *FM 2006: Formal Methods, 14th International Symposium on Formal Methods, Hamilton, Canada*, volume 4085 of *Lecture Notes in Computer Science*, pages 460–475. Springer, 2006.

[Ler06] Xavier Leroy. Formal Certification of a Compiler Back-end or: Programming a Compiler with a Proof Assistant. In J. Gregory Morrisett and Simon L. Peyton Jones, editors, *Proceedings of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2006, Charleston, South Carolina, USA*, pages 42–54, 2006.

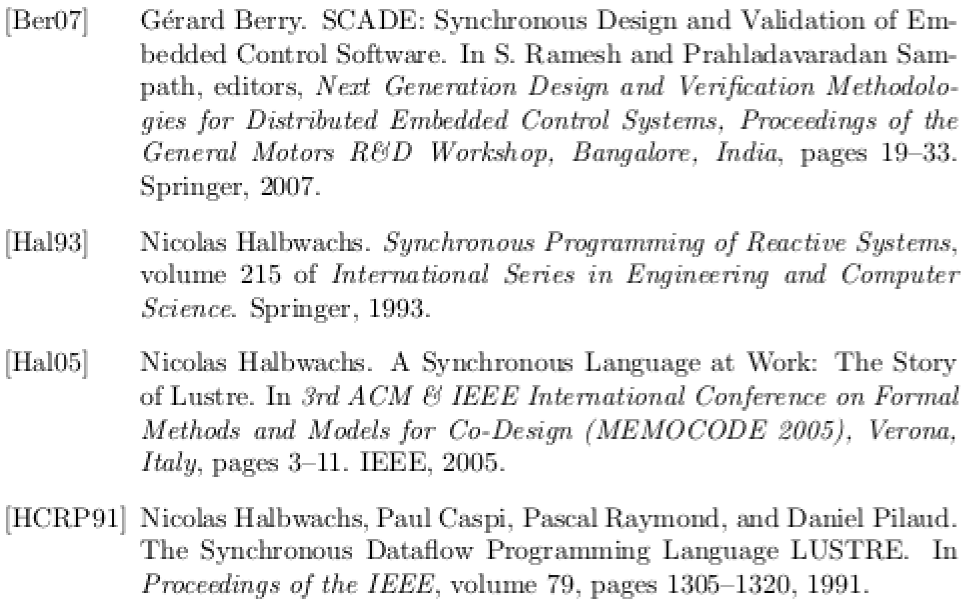
2007



[Berry]
[Halbwachs-Caspi-Raymond-Pilaud]

Design, validation, and implementation of avionics, automotive, railway, and other safety-critical applications using the SCADE tools for the synchronous language Lustre

[2007]



[Ber07] Gérard Berry. SCADE: Synchronous Design and Validation of Embedded Control Software. In S. Ramesh and Prahladavaradan Sampath, editors, *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems, Proceedings of the General Motors R&D Workshop, Bangalore, India*, pages 19–33. Springer, 2007.

[Hal93] Nicolas Halbwachs. *Synchronous Programming of Reactive Systems*, volume 215 of *International Series in Engineering and Computer Science*. Springer, 1993.

[Hal05] Nicolas Halbwachs. A Synchronous Language at Work: The Story of Lustre. In *3rd ACM & IEEE International Conference on Formal Methods and Models for Co-Design (MEMOCODE 2005), Verona, Italy*, pages 3–11. IEEE, 2005.

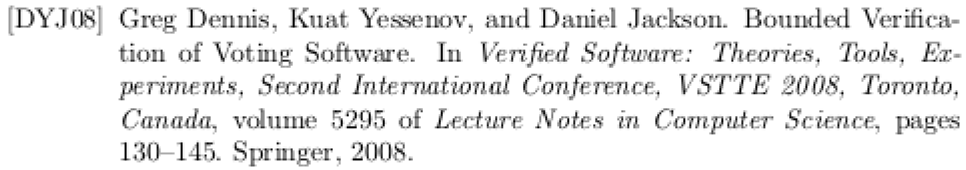
[HCRP91] Nicolas Halbwachs, Paul Caspi, Pascal Raymond, and Daniel Pilaud. The Synchronous Dataflow Programming Language LUSTRE. In *Proceedings of the IEEE*, volume 79, pages 1305–1320, 1991.

2008

[Dennis-Yessenov-Jackson]

Formal verification using JML and ESC/Java in the vote-tallying part of the KOA open source software used for remote voting in Dutch public elections: discovery of specification errors and programming bugs undetected so far

[2008]



[DYJ08] Greg Dennis, Kuat Yessenov, and Daniel Jackson. Bounded Verification of Voting Software. In *Verified Software: Theories, Tools, Experiments, Second International Conference, VSTTE 2008, Toronto, Canada*, volume 5295 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2008.

2009

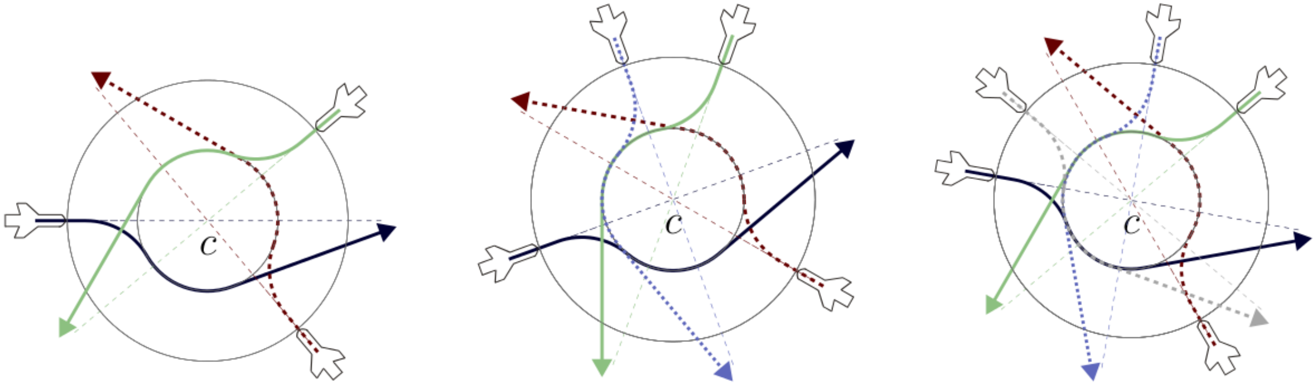
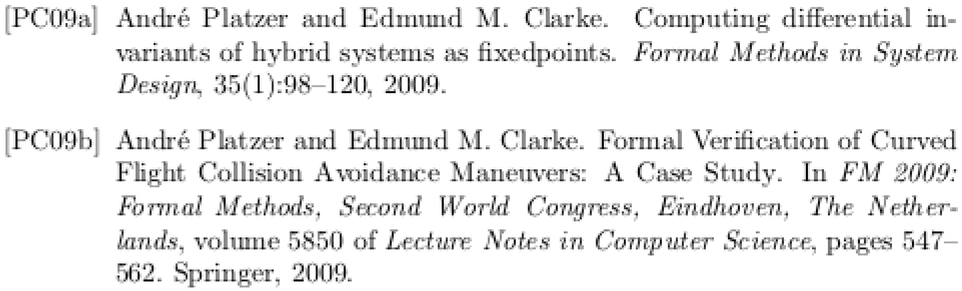


Figure 8: Flyable aircraft roundabout (multiple aircraft)

[Platzer-Clarke]

Formal verification of curved flight collision avoidance maneuvers using the KeYmaera verification tool for hybrid systems

[2009]



[PC09a] André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. *Formal Methods in System Design*, 35(1):98–120, 2009.

[PC09b] André Platzer and Edmund M. Clarke. Formal Verification of Curved Flight Collision Avoidance Maneuvers: A Case Study. In *FM 2009: Formal Methods, Second World Congress, Eindhoven, The Netherlands*, volume 5850 of *Lecture Notes in Computer Science*, pages 547–562. Springer, 2009.

2010

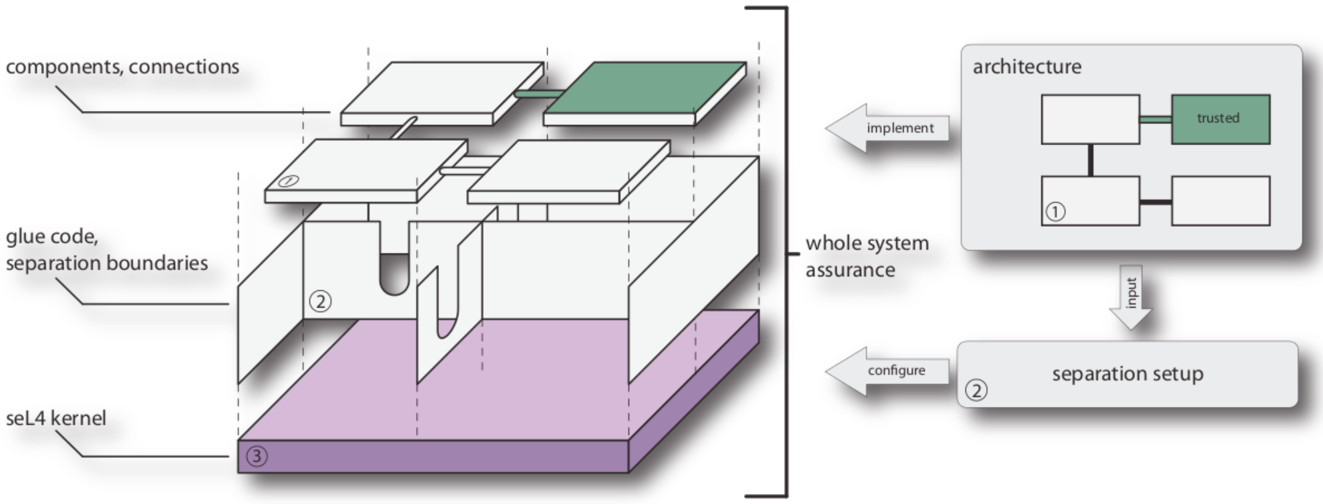
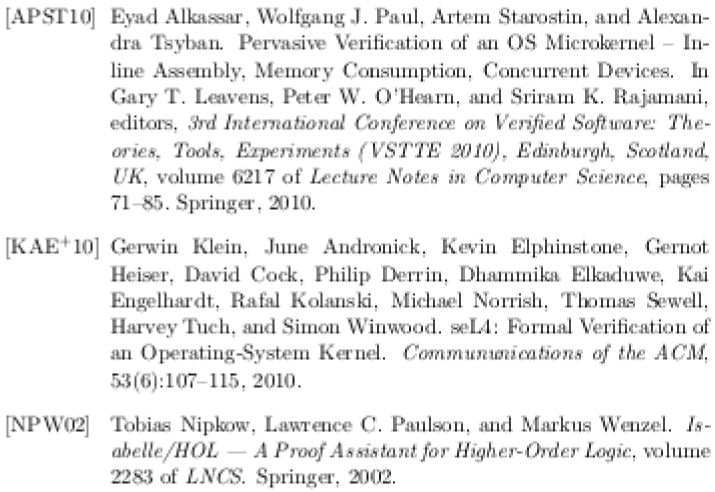


Figure 3: seL4-based system with multiple independent levels of assurance

[Klein-Andronick-Elphinstone-Heiser-Cock-Derrin-Elkaduwe-Engelhardt-Kolanski-Norrish-Sewell-Tuch-Winwood]

Formal verification of the seL4 general purpose operating-system micro-kernel using the Isabelle/HOL theorer prover

[2010]



[APST10] Eyad Alkassar, Wolfgang J. Paul, Artem Starostin, and Alexandra Tsyban. Pervasive Verification of an OS Microkernel – Inline Assembly, Memory Consumption, Concurrent Devices. In Gary T. Leavens, Peter W. O'Hearn, and Sriram K. Rajamani, editors, *3rd International Conference on Verified Software: Theories, Tools, Experiments (VSTTE 2010), Edinburgh, Scotland, UK*, volume 6217 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2010.

[KAE+10] Gerwin Klein, June Andronick, Kevin Elphinstone, Gernot Heiser, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal Verification of an Operating-System Kernel. *Communications of the ACM*, 53(6):107–115, 2010.

[NPW02] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

2011

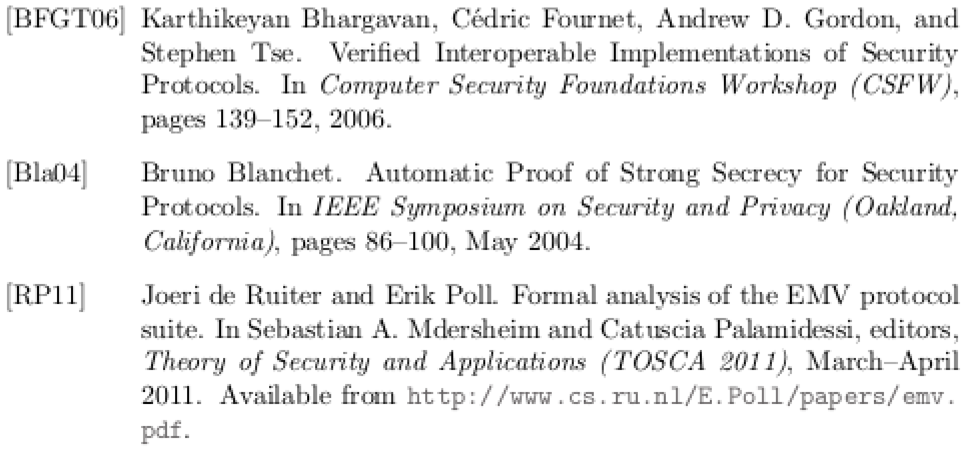
[de Ruiter-Poll]

Formal modelling of the EMV (Europay-MasterCard-Visa) protocol suite in the F# language

Automated analysis of these protocols by joint use of:

- FS2PV translator [Barghavan-Fournet-Gordon-Tse]
- ProVerif [Blanchet]

[2011]



[BFGT06] Karthikeyan Bhargavan, Cédric Fournet, Andrew D. Gordon, and Stephen Tse. Verified Interoperable Implementations of Security Protocols. In *Computer Security Foundations Workshop (CSFW)*, pages 139–152, 2006.

[Bla04] Bruno Blanchet. Automatic Proof of Strong Secrecy for Security Protocols. In *IEEE Symposium on Security and Privacy (Oakland, California)*, pages 86–100, May 2004.

[RP11] Joeri de Ruiter and Erik Poll. Formal analysis of the EMV protocol suite. In Sebastian A. Mdersheim and Catuscia Palamidessi, editors, *Theory of Security and Applications (TOSCA 2011)*, March–April 2011. Available from `http://www.cs.ru.nl/E.Poll/papers/emv.pdf`.

Conclusion

Conclusion

- About formal methods:
  - Much has been done in 30 years
  - Great diversity of applications
  - Key issue: how to incorporate formal methods in standard engineering practice?
- About this study:
  - A careful selection of success stories
  - Not limitative: there are more success stories
  - Perhaps biased towards over-published works
  - Other lists could be made:
    - 30 most useful fundamental results
    - 30 best software tools