Formal Methods Technology Transfer: A View from NASA

James L. Caldwell

Flight Electronics Technology Division
NASA Langley Research Center
Hampton, Virginia 23681

26 February 1996

Abstract

Since 1988 NASA Langley Research Center has supported a formal methods research group. From its inception, a primary goal of the program has been the transfer of formal methods technology into aerospace industries focusing on applications in commercial air transport. The overall program has been described elsewhere. An up-to-date and extensive account of the program can be found at the NASA Langley Formal Methods Home Page on the World-Wide Web[1]. In this paper I remark on the technology transfer strategy and its evolution.

1 Introduction

In 1988, Ricky Butler at NASA Langley Research Center (Langley) founded a formal methods program. From its inception, a primary goal of the program has been to transfer formal methods technology and formally justified fault-tolerant algorithms and architectures into the aerospace industry, with a focus on commercial air transport.

At the time the formal methods program was conceived, it was clear that the future for commercial air transports was tied to fly-by-wire digital control. The design and validation of digital flight controls pose challenging problems. To obtain FAA certification, systems must be built to ultra-reliable standards, a level of reliability quantified as an error rate of less than 10^{-10} per hour [26]. Individual hardware devices, including power supplies, can not be built to these levels of reliability, a fact that generates the requirement for fault-tolerant

[1] http://atb-www.larc.nasa.gov/fm.html

designs based on redundant electrically isolated systems. Because adding fault-tolerance mechanisms to a system further increases complexity, the expected gain in reliability clearly depends on the correctness of the mechanism. The strict reliability requirements and the subtlety and complexity of the interactions between the distributed components rule out testing as the primary means of assurance. Indeed, for software, testing can not establish reliability to these levels [11]. Based on this argument, the most intellectually defensible method to achieve the high levels of reliability (but not to quantify them) seemed to rest with logical and deductive methods: formal methods.

The Langley research program was founded on the assumption that issues of fault-tolerance and redundancy management should be considered independently of the problem of functional correctness of the control law codes. We are not control theorists nor do we believe control theorists should be responsible for implementing the required fault-tolerance. This view is not universal; however, there is ample evidence to support the position that, even for embedded systems, the concerns of fault-tolerance and the application codes are largely orthogonal.

Initially, our sights were set on obtaining and disseminating results which could be used by the (often skeptical) engineers who design these systems. Our ultimate goal was to transfer the methods into the hands of engineers in industry.

The overall program has been described in some detail elsewhere; both from the programmatic point of view of NASA Langley [7, 6], and from the perspective of SRI researchers who have been partners in the effort from the beginning [46]. An up-to-date account of the program can be found at the NASA Langley Formal Methods Home Page on the World-Wide Web. In this paper we concentrate on aspects of technology transfer and discuss the evolution of the strategy.

2 The NASA Approach to Formal Methods

Historically, the NASA program has concentrated on the most rigorous application of the methods: machine checked proofs. Unlike some well-known European efforts [34, 60] where formal specification is emphasized —often without correctness proofs or with pencil and paper proofs but not ones checked by machine— the Langley program has concentrated on proofs of correctness checked by machine. We admit the advantage of formal specification but prefer the assurance of mechanically checked proofs. [2]

The early Langley program did not fund the development of new founda-

[2]The emphasis in the U.S. on mechanically checked proof can be partially attributed to the National Security Agency's (NSA) funding support for the field. The NSA standard for computer security, the so-called "Orange Book", required formal verifications as part of the certification process for the highest level secure devices. This requirement drove NSA funded researchers and others in the U.S. toward machine checked proof.

tional theories or the development of general purpose tools. This is important research but within constraints on our own funding we have concentrated on applying existing tools to problems we believe are both tractable and important to industry. The emphasis on a particular class of systems and specific problems has focused the research and amplified the results. With the stated goals of technology transfer in mind and with limited program funds, the use of existing tools was initially perceived as a necessary but undesirable constraint. It has, in retrospect, accounted for some measure of our success. By concentrating attention on results instead of tools we believe we have had more impact than we might have had otherwise. In fact, SRI has attributed the experience gained under our program with a significant influence on the design of their PVS system [46].

3 Evolution of a Technology Transfer Model

Our strategy for technology transfer has evolved as the program has matured. We can divide the program into two periods. The first, from 1989 to the summer of 1992, was spent exploring the methods, learning to apply the provers, and building a portfolio of results. The second period, from the autumn of 1992 until the present, has been spent largely on collaborative technology transfer projects with industrial partners.

3.1 The early period: 1988 to 1992

Butler began by building a team. I was hired in Fall 1988, joining Butler and two others, Jon Sjogren and Sally Johnson (both of whom have subsequently left the program for other positions.) In December 1988, a request for proposals was published soliciting contractual partners in the newly formed formal methods program. In the fall of 1989 contracts were awarded to the following companies: Computational Logic Incorporated (CLI) in Austin, Texas; Odyssey Research Associates (ORA) in Ithaca, New York; and Stanford Research Institute International (SRI) in Menlo Park, California. In December of the same year Ben DiVito joined the in-house group as a contractor with VíGYAN. In 1990, Paul Miner and Victor Carreño, both NASA civil servants, joined the formal methods team.

One impetus for the formation of the group was the initiation of a Memorandum Of Understanding between NASA Langley and the developers of the Viper microprocessor at the Royal Signals and Radar Establishment (RSRE) at Malvern, England. Butler and Sjogren had performed a study of the RSRE hardware verification methodology in SRI's EHDM prover [12]. The first work performed by CLI was an evaluation of the overall RSRE methodology [5].

The first tasks undertaken toward the development of a basis of formally verified fault-tolerant algorithms were a continuation of earlier work initiated in the SIFT project [28]. SIFT had been developed at SRI under Langley sponsorship. The SIFT fault-tolerant architecture was based on exact-match voting of results by synchronized replicated processors. The lowest level building-blocks of such systems are implementations, in both hardware and software, of Byzantine resilient fault-tolerant algorithms for distributed clock synchronization, and interactive consistency. These are intellectually challenging problems requiring ingenious solutions. As such, they are often the ones engineers have most difficulty convincing themselves they've got right. In the first years of the program, formal specifications and verifications for these building blocks were undertaken and completed in the following systems: EHDM and later in PVS at SRI [55, 50, 59]; Nqthm, the Boyer Moore prover, at CLI [1, 2, 45, 44, 68]; and Clio and the Penelope Ada verification system at ORA [61, 4, 63, 30, 62].

Concurrently we gained experience at Langley using a number of the prover systems including EHDM, Nqthm, and HOL. During that time we were primarily occupied with the design and verification (in EHDM) of a fault-tolerant architecture designed to withstand transient faults [23, 8, 22, 21, 20, 41]. Generalizing our models, Rushby at SRI verified transient recovery for an abstract architecture [48, 49].

Through Langley support, researchers at Boeing worked with Carl Levitt's group at the University of California at Davis. The Davis group worked in the HOL system. Boeing and Davis researchers collaborated to verify a set of hardware devices, including a microprocessor, a floating-point coprocessor based on a scaled down Intel 8087, a direct memory access (DMA) controller modeled on a scaled down Intel 8237A, and a set of memory-management units [47, 35, 56, 58, 57, 27]. Under partial sponsorship of this project Windley developed his HOL theory of generic interpreters and formalized a method for composing verified devices into systems [65, 66, 67].

During this early period the Langley formal methods group sponsored two workshops. The first was held in August 1990 [13]. It served two purposes: to build cooperation and communication between geographically isolated members of the formal methods team; and to give them an overview of the state-of-the-practice in digital flight control by a series of talks from industry. The second workshop, held in August 1992, focused on presenting the body of results just cited to the aerospace community [33].

3.2 Toward a model for technology transfer.

The Boeing and Draper projects initiated in the first period represented our first efforts to actually transfer formal methods into external organizations. These projects formed a basis of comparison for future efforts. The Boeing project was initiated to demonstrate the feasibility of verifying hardware; the results were verifications of a number of components having properties not addressed in previous hardware verification efforts.

The work by Bickford and Srivas of ORA with engineers from Charles Stark

Draper Labs (Draper) in verifying the "scoreboard" circuit of the Fault Tolerant Parallel Processor (FTPP) was significant because the verification uncovered errors previously unknown to the Draper engineers. Because Bickford and Srivas were not experts in fault-tolerant hardware design, the discovery of errors through the formal analysis was especially significant. As has happened on many projects before and since, the analysis of failed proofs not only revealed flaws in the design but suggested solutions. In the case of the scoreboard, the failed proofs showed the assumptions about the initial state were too weak. This in turn revealed that the initialization of the circuitry was flawed. When the scoreboard chip was fabricated, it worked. Draper reported that this was the first time they had fabricated a new design that worked on the first iteration. They attributed the success to their use of VHDL and to ORA's application of formal methods. Based on the experience gained in the FTPP verification ORA began work on a VHDL verification tool which was awarded funding by the U.S. Air Force.

As examples of technology transfer, both projects were counted as successful. In both cases, the industrial partners reported increased confidence and commitment to formal methods; and in both cases, the application of formal methods advanced the state of the practice and resulted in the development of new methods. On the other hand, these projects did not have direct impact on actual products. In both cases the industrial partners were research organizations in their own right. In the Boeing work done before 1992, the work was closer to actual engineering development processes than previous Langley efforts, but the early projects only verified paper designs that were never realized in hardware. In the Draper collaboration, errors were uncovered in a physically realized hardware device; but the FTPP was itself a research project and was never incorporated into any systems. In both projects, NASA funded the work of both the formal methods experts and, under other contracts, the time of engineers at Boeing and Draper. These weaknesses may well be inevitable for first attempts at technology transfer.

The significance of these projects for our program lay in their role as models for future technology transfer projects. Perhaps most importantly, in both projects, there were individuals on the industry side who were anxious to apply formal methods in his problem area and who were generally optimistic about the chances for success. In both projects, a NASA person acting as technical monitor established and coordinated contact between the external formal methods group and the industrial partner. Also, the formal methods contractors studied the problem with applications engineers from the industry and then worked the verifications independently. To improve the results of the technology transfer process, for future projects, we planned to seek industrial partners willing to pay for the expenses (time, travel, and equipment) incurred by their own engineering staffs and hoped to find partners who would begin to apply the methods themselves.

3.3 Application of the Model: 1992 - 1996

There were some personnel and contract changes in the second period. In 1992 Michael Holloway joined the Langley team. A component of the original staffing plan was for civil servant researchers to pursue advanced degrees in the field under the graduate study program supported by NASA. Under this program, Caldwell attended Cornell University studying computer science and logic, Carreño attended Cambridge University in England working with Mike Gordon in his HOL group, and Miner attended Indiana University in the computer science and applied logic program. All three have since returned to Langley and are currently working to finish dissertations. The existing contracts with CLI, ORA and SRI expired in 1994; a new request for proposals was published in May of 1994. Contracts were awarded later in the same year to ORA and SRI.

The second workshop marked the beginning of a new phase in the program. The Draper and Boeing projects continued and a number of new projects and partnerships were initiated which were based on the emerging technology transfer model.

Using the techniques developed earlier, the U.C. Davis and Boeing group verified a microprocessor realizing the VIPER instruction set [36]. Within Boeing, Fura at Boeing Defense went on to apply the methodology to a proprietary memory/bus interface device [27] developed for use in Boeing's Fault-tolerant Embedded Processor (FTEP).

ORA continued work with Draper on the FTPP, applying a prototype of their VHDL verification system [3] to the VHDL designs of the Scoreboard circuit of the FTPP. The ORA VHDL tool was developed under other government funding but, like SRI's PVS system, the design of the tool was driven by the results of applied verifications funded under the Langley program.

ORA has worked with Honeywell Air Transport Systems Division in Phoenix Arizona since 1993. They have been working to incorporate formal methods in the software development process at Honeywell. They started by constructing a formal model of a navigation system. More recently, the collaboration has resulted in the development of TableWise [32], a tool based on a generalized BDD model used to analyze decision tables. Honeywell has applied it to the analysis of tables used in auto-pilot applications. The project continues into 1996 with ORA considering marketing a tool based on the prototype built for Honeywell.

Since 1993, ORA has also been working with Union Switch and Signal in Pittsburgh, Pennsylvania on railroad switching applications. In the first phase of this work ORA formalized a high level model of railway switching networks and provided a formal definition of safety; a proof that trains operating under the standard fixed-block control scheme are safe was also performed. The second phase of work analyzed correctness of a compilation process within a proprietary CAD system developed by US&S for use by applications engineers. This work continues into 1996.

Under Langley sponsorship, a collaboration between SRI and Collins Commercial Avionics division of Rockwell International was started in 1993. The collaboration between Srivas, now at SRI, and Miller of Collins resulted in the specification and a partial verification of the micro-code of the AAMP5 microprocessor [64, 39]. Butler and Miner at Langley constructed a PVS library for a bit-vector type in support of the effort. The most significant results from the effort were the discovery of two errors seeded by Collins engineers and of two other previously unknown errors [64]. The newly discovered errors were corrected before the first round of fabrication. The success of the collaboration has lead to a follow-on project to verify a new member of the AAMP microprocessor family. Collins engineers are taking a more active role in applying the prover and are performing a majority of the formal proofs in the new project themselves.

SRI also collaborated with Allied Signal in developing fault-tolerant algorithms for a hybrid fault model; researchers at Allied Signal had noticed that the standard Byzantine resilient algorithm for interactive consistency could be made more robust. This work lead to numerous publications as SRI modified and reverified previously verified algorithms under the new fault-model [37, 29, 38, 53, 52]. Aside from the development of the new algorithms, which was important work in its own right, this work has significance for formal methods because SRI was able to effectively reuse the artifacts of their earlier verifications.

Under FAA sponsorship through the Langley program, Rushby of SRI wrote an extended report on the application of formal methods to validation of digital systems [51]. This huge document was condensed into a chapter on formal methods [54] for the FAA Digital Systems Validation Handbook[16].

Also, in 1992, a team was formed to study applications and transfer of formal methods into NASA space programs. The team consists of researchers and practitioners from Langley, Johnson Space Center, the Jet Propulsion Lab, Loral Space Information Systems, SRI, and VíGYAN. Portions of the Space Shuttle flight control requirements and change requests have formalized and analyzed [25, 18, 24, 17, 19].

At Langley, NASA researchers continued work on refining the fault-tolerant architecture developed earlier by specifying lower levels of the hierarchy and formally proving they are implementations via refinement mappings [9, 10]. Miner continued work on extending the SRI clock-synchronization results and implementing a circuit in silicon [40, 43]. Models of the IEEE floating point standards were formalized in HOL and PVS [14, 42, 15].

A third workshop was held in May of 1995 reporting on work completed to date [31].

3.4 New projects

New projects starting in 1996 include the development of a formal model for time, space, and I/O partitioning intended for application to integrated avionics computing systems. This research is funded through Langley by the FAA in response to the industrial move toward integration of non-critical tasks with flight-critical tasks on the same computing platform. In a related task the SafeBus protocol, developed at Honeywell and used in the Boeing 777, will be analyzed for similar partitioning properties.

4 Reflections on Technology Transfer

Technology transfer is notoriously difficult. Engineers in established disciplines rarely welcome change into processes that have been successful. This inertia is prudent and rational. The model of technology transfer we have applied has allowed our industrial partners to apply formal methods with little risk and minimal cost.

4.1 Specification, Proof, and Pushbutton

Formal specifications (which in our industrial projects has primarily meant PVS specifications) have been readily accepted by engineers and they have few problems reading them. Also, formal specifications have helped find many discrepancies and errors. The two non-seeded bugs found in the AAMP5 project were discovered during the specification phase [64]. The advantage of formal specification has been observed in nearly all of our projects and agrees with experiences gained elsewhere. It should be remarked that specifications in PVS can be type-checked[3] and they usually are. The transition from reading to writing formal specifications can be difficult [17]; but there is no reason to believe this obstacle can not be overcome through training. Langley researchers recently gave a PVS training session for Loral requirements engineers that was well received. Miller reports [64] the Collins engineers had no difficulties in learning the PVS language.

Formal proofs seem to pose another level of challenge; however, at Collins, engineers are using the PVS prover. In fact, Miller is quite optimistic and reported the Collins engineers had few difficulties learning to use the prover [64]. Typically, they have relied on powerful strategies developed for their particular application and they have applied them by modifying proofs previously developed at SRI. In their application, micro-code verification, this approach, based on what might be called proof templates, is feasible because the correctness proofs for instructions within a single instruction class are all quite similar. The

[3] The type system supported by PVS is undecidable. Because of this, type-checking typically generates proof obligations, typically these type-check conditions (TCC's) only require trivial interactions with the prover.

first verification of an instruction from a new class may be difficult, but once found, the verification of the others typically follow a similar pattern. As experience within Collins grows they tackle more and varied proof obligations on their own. Their experiences in learning to use the PVS prover parallel our own early experiences with EHDM at Langley.

As mentioned above, ORA's work with Honeywell resulted in the development of the TableWise tool. Collaborating with system engineers from Honeywell, ORA was able to isolate a problem area that was of concern to Honeywell engineers and that was amenable to automatic techniques. ORA responded with a specialized analysis tool that applies formal techniques but requires no knowledge of formal methods on the part of the user. This project illustrates the power of applying formal techniques to a single component of an industrial problem. This is in contrast to attempts at grand solutions; an open ear and a flexible tool box of techniques may be one of the most valuable assets for industrial applications of formal methods.

In many ways, the Honeywell and Collins projects represent opposite ends of the spectrum of formal methods application. ORA's work on the Honeywell project is applied to software development, especially to the earlier stages of the life-cycle. The Collins work is micro-code verification. It seems fair to say that the technology transfer at Collins has more depth. The Collins engineers perform proofs in PVS, while the Honeywell engineers needn't even know they're applying a formal method. On the other hand, the transfer to Honeywell has more breadth in that TableWise tool is usable by anyone, while at Collins only those engineers having received the necessary training can contribute productively to formal methods projects. These two projects have applied formal methods in significantly different ways and offer competing models of transferring formal methods into industry. Future work under the Langley program will explore the tradeoffs in these two approaches.

5 Conclusion

In every case we have applied formal methods to industrial problems, previously undiscovered errors have been reported. Remarkably, this has even been true of applications of the methods to algorithms published in respected journals. In the end, this is the value added by formal methods, the discovery of errors and improved confidence in correctness.

Acknowledgments
The success of the NASA program belongs to the entire team, and especially to Ricky Butler. Thanks for editing and comments on this paper go to my other colleagues at NASA: Victor Carreño; Michael Holloway; Paul Miner; and also to Ben DiVito at VíGYAN.

References

[1] William R. Bevier and William D. Young. Machine checked proofs of the design and implementation of a fault-tolerant circuit. NASA Contractor Report 182099, November 1990.

[2] William R. Bevier and William D. Young. The proof of correctness of a fault-tolerant circuit design. In *Second IFIP Conference on Dependable Computing For Critical Applications*, pages 107–114, Tucson, Arizona, February 1991.

[3] Mark Bickford. A formal semantics for a subset of VHDL and its use in analysis of the ftpp scoreboard circuit. NASA Contractor Report 191577, April 1994.

[4] Mark Bickford and Mandayam Srivas. Verification of the FtCayuga fault-tolerant microprocessor system (Volume 2: Formal specification and correctness theorems). NASA Contractor Report 187574, July 1991.

[5] Bishop Brock and Warren A. Hunt, Jr. Report on the formal specification and partial verification of the VIPER microprocessor. NASA Contractor Report 187540, July 1991.

[6] Ricky W. Butler. NASA Langley's research program in formal methods. In *6th Annual Conference on Computer Assurance (COMPASS 91)*, Gaithersburg, MD, June 1991.

[7] Ricky W. Butler, James L. Caldwell, Victor A. Carreno, C. Michael Holloway, Paul S. Miner, and Ben L. Di Vito. NASA Langley's research and technology transfer program in formal methods. In *Tenth Annual Conference on Computer Assurance (COMPASS 95)*, Gaithersburg, MD, June 1995.

[8] Ricky W. Butler, James L. Caldwell, and Ben L. Di Vito. Design strategy for a formally verified reliable computing platform. In *6th Annual Conference on Computer Assurance (COMPASS 91)*, Gaithersburg, MD, June 1991.

[9] Ricky W. Butler and Ben L. Di Vito. Formal design and verification of a reliable computing platform for real-time control (Phase 2 results). NASA Technical Memorandum 104196, January 1992.

[10] Ricky W. Butler, Ben L. Di Vito, and C. Michael Holloway. Formal design and verification of a reliable computing platform for real-time control (Phase 3 results). NASA Technical Memorandum 109140, August 1994.

[11] Ricky W. Butler and George B. Finelli. The infeasibility of quantifying the reliability of life-critical real-time software. *IEEE Transactions on Software Engineering*, 19(1):3–12, January 1993.

[12] Ricky W. Butler and Jon A. Sjogren. Hardware proofs using EHDM and the RSRE verification methodology. NASA Technical Memorandum 100669, December 1988.

[13] Ricky W. Butler, (ed.). NASA formal methods workshop 1990. NASA Conference Publication 10052, November 1990.

[14] Victor A. Carreño. Interpretation of IEEE-854 floating-point standard and definition in the HOL system. September 1995.

[15] Victor A. Carreño and Paul S. Miner. Specification of the IEEE-854 floating-point standard in HOL and PVS. To be presented at the 1995 International Workshop on Higher Order Logic Theorem Proving and its Applications, September 11-14, Aspen Grove, Utah, USA, as a track B paper and included in supplemental proceedings, 1995.

[16] Computer Resource Management Inc. In *Digital Systems Validation Handbook – volume III*, number DOT/FAA/CT-88/10. FAA.

[17] Judith Crow and Ben L. Di Vito. Formalizing space shuttle software requirements. In *Workshop on Formal Methods in Software Practice (FMSP '96)*, pages 40–48, San Diego, California, January 1996.

[18] Judy Crow. Finite-State Analysis of Space Shuttle Contingency Guidance Requirements. Technical Report SRI-CSL-95-17, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1995. Also forthcoming as a NASA Contractor Report for Task NAS1-20334.

[19] Ben L. Di Vito. Formalizing new navigation requirements for nasa's space shuttle. In *Formal Methods Europe (FME '96)*, Oxford, England, March 1996.

[20] Ben L. Di Vito and Ricky W. Butler. Provable transient recovery for frame-based, fault-tolerant computing systems. In *Real-Time Systems Symposium*, Phoenix, Az, December 1992.

[21] Ben L. Di Vito and Ricky W. Butler. Formal techniques for synchronized fault-tolerant systems. In *Dependable Computing for Critical Applications 3*, Dependable Computing and Fault-Tolerant Systems, pages 279–306. Springer Verlag, Wien New York, 1993. Also presented at 3rd IFIP Working Conference on Dependable Computing for Critical Applications, Mondello, Sicily, Italy, Sept. 14–16, 1992.

[22] Ben L. Di Vito, Ricky W. Butler, and James L. Caldwell. High level design proof of a reliable computing platform. In *Dependable Computing for Critical Applications 2*, Dependable Computing and Fault-Tolerant Systems, pages 279–306. Springer Verlag, Wien New York, 1992.

[23] Ben L. Di Vito, Ricky W. Butler, and James L. Caldwell, II. Formal design and verification of a reliable computing platform for real-time control (Phase 1 results). NASA Technical Memorandum 102716, October 1990.

[24] Ben L. Di Vito and Larry Roberts. Using Formal Methods to Assist in the Requirements Analysis of the Space Shuttle GPS Change Request. Contractor report, NASA Langley Research Center, Hampton, VA, 1996. To appear.

[25] John Kelly et. al. *Formal Methods Demonstration Project for Space Applications – Phase I Case Study: Space Shuttle Orbit DAP Jet Select.* Multi-Center NASA Team from Jet Propulsion Laboratory, Johnson Space Center, and Langley Research Center, December 1993. NASA Code Q Final Report (Unnumbered).

[26] FAA. System Design and Analysis. Advisory Circular AC 25.1309-1A, U.S. Department of Transportation, June 1988.

[27] David A. Fura, Phillip J. Windley, and Gerald C. Cohen. Formal design specification of a processor interface unit. NASA Contractor Report 189698, November 1992.

[28] Jack Goldberg et al. Development and analysis of the software implemented fault-tolerance (SIFT) computer. NASA Contractor Report 172146, 1984.

[29] Li Gong, Patrick Lincoln, and John Rushby. Byzantine agreement with authentication: Observations and applications in tolerating hybrid and link faults. In *Dependable Computing for Critical Applications (DCCA-5)*, Champaign, IL, September 1995.

[30] David Guaspari. Formally specifying the logic of an automatic guidance controller. In *Ada-Europe Conference*, Athens, Greece, May 1991.

[31] C. Michael Holloway. Third NASA formal methods workshop 1995. NASA Conference Publication 10176, 1995.

[32] Doug Hoover and Zewei Chen. Tbell: A mathematical tool for analyzing decision tables. NASA Contractor Report 195027, November 1994. Note: Tbell is now known as TableWise.

[33] Sally C. Johnson, C. Michael Holloway, and Ricky W. Butler. Second NASA formal methods workshop 1992. NASA Conference Publication 10110, November 1992.

[34] Cliff B. Jones. *Systematic Software Development using VDM*. International Series in Computer Science. Prentice Hall, Hemel Hempstead, UK, 2nd edition, 1992.

[35] Sara Kalvala, Myla Archer, and Karl Levitt. A methodology for integrating hardware design and verification. In *ACM International Workshop on Formal Methods in VLSI Design*, Miami, FL, January 1991.

[36] Karl Levitt and et. al. Formal verification of a microcoded VIPER microprocessor using HOL. NASA Contractor Report 4489, February 1993.

[37] Patrick Lincoln and John Rushby. A formally verified algorithm for interactice consistency under a hybrid fault model. NASA Contractor Report 4527, July 1993.

[38] Patrick Lincoln and John Rushby. Formal verification of an interactive consistency algorithm for the draper ftp architecture under a hybrid fault model. In *1994 Computer Assurance (COMPASS) Conference*, June 1994.

[39] Steve 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 30–43, Boca Raton, Florida USA, April 1995.

[40] Paul S. Miner. An extension to schneider's general paradigm for fault-tolerant clock synchronization. NASA Technical Memorandum 107634, Langley Research Center, Hampton, VA, 1992.

[41] Paul S. Miner. Verification of fault-tolerant clock synchronization systems. NASA Technical Paper 3349, November 1993.

[42] Paul S. Miner. Defining the IEEE-854 floating-point standard in PVS. NASA Technical Memorandum 110167, NASA, Langley Research Center, Hampton, VA, June 1995.

[43] Paul S. Miner, Peter A. Padilla, and Wilfredo Torres. A provably correct design of a fault-tolerant clock synchronization circuit. In *11th Digital Avionics Systems Conference*, pages 341–346, Seattle, WA, October 1992.

[44] J Strother Moore. A formal model of asynchronous communication and its use in mechanically verifying a biphase mark protocol. NASA Contractor Report 4433, June 1992.

[45] J Strother Moore. Mechanically verified hardware implementing an 8-bit parallel io byzantine agreement processor. NASA Contractor Report 189588, April 1992.

[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] Jing Pan and Karl Levitt. Towards a formal specification of the IEEE floating-point standard with application to the verification of floating-point coprocessors. In *24th Asilomar Conference on Signals, Systems & Computers*, Monterrey, CA., November 1990.

[48] John Rushby. Formal specification and verification of a fault-masking and transient-recovery model for digital flight-control systems. NASA Contractor Report 4384, July 1991.

[49] John Rushby. Formal specification and verification of a fault-masking and transient-recovery model for digital flight-control systems. In *Second International Symposium on Formal Techniques in Real Time and Fault Tolerant Systems*, volume 571 of *Lecture Notes in Computer Science*, pages 237–258. Springer Verlag, Nijmegen, The Netherlands, January 1992.

[50] John Rushby. Formal verification of an oral messages algorithm for interactive consistency. NASA Contractor Report 189704, October 1992.

[51] John Rushby. Formal methods and digital systems validation for airborne systems. NASA Contractor Report 4551, December 1993.

[52] John Rushby. A formally verified algorithm clock sychronization under a hybrid fault model. In *ACM Principles of Distributed Computing '94*, August 1994.

[53] John Rushby. Improvements in the formally verified analysis of the interactive convergence clock synchronization algorithm and its extension to a hybrid fault model. NASA Contractor Report 194970, 1994.

[54] John Rushby. Formal methods and their role in digital systems validation for airborne systems. NASA Contractor Report 4673, August 1995.

[55] John Rushby and Friedrich von Henke. Formal verification of a fault-tolerant clock synchronization algorithm. NASA Contractor Report 4239, June 1989.

[56] Thomas Schubert and Karl Levitt. Verification of memory management units. In *Second IFIP Conference on Dependable Computing For Critical Applications*, pages 115–123, Tucson, Arizona, February 1991.

[57] Thomas Schubert, Karl Levitt, and Gerald C. Cohen. Towards composition of verified hardware devices. NASA Contractor Report 187504, November 1991.

[58] Thomas Schubert, Karl Levitt, and Gerald C. Cohen. Formal verification of a set of memory management units. NASA Contractor Report 189566, 1992.

[59] Natarajan Shankar. Mechanical verification of a schematic Byzantine clock synchronization algorithm. NASA Contractor Report 4386, July 1991.

[60] J.M. Spivey, editor. *The Z Notation: A Reference Manual.* Prentice Hall International Series in Computer Science. Prentice Hall, Hemel Hempstead, UK, 1989.

[61] Mandayam Srivas and Mark Bickford. Verification of the FtCayuga fault-tolerant microprocessor system (Volume 1: A case study in theorem prover-based verification). NASA Contractor Report 4381, July 1991.

[62] Mandayam Srivas and Mark Bickford. Moving formal methods into practice: Verifying the FTPP scoreboard: Phase 1 results. NASA Contractor Report 189607, May 1992.

[63] Mandayam Srivas and Mark Bickford. Verification of a fault-tolerant property of a multiprocessor system. In *Theorem Provers in Circuit Design: Theory, Practice and Experience*, Nijmegen, The Netherlands, June 1992.

[64] Mandayam Srivas and Steve Miller. Formal verification of an avionics microprocessor. NASA Contractor Report 4682, July 1995.

[65] Phillip J. Windley. Abstract hardware. In *ACM International Workshop on Formal Methods in VLSI Design*, Miami, FL, January 1991.

[66] Phillip J. Windley, Karl Levitt, and Gerald C. Cohen. Formal proof of the AVM-1 microprocessor using the concept of generic interpreters. NASA Contractor Report 187491, March 1991.

[67] Phillip J. Windley, Karl Levitt, and Gerald C. Cohen. The formal verification of generic interpreters. NASA Contractor Report 4403, October 1991.

[68] William D. Young. Verifying the interactive convergence clock synchronization algorithm using the boyer-moore theorem prover. NASA Contractor Report 189649, April 1992.