

Curriculum Vitae

Dr. Klaus Havelund

Ph.D in Computer Science
Principal Scientist at NASA's Jet Propulsion Laboratory
California Institute of Technology (Caltech)
Appointed Caltech Lecturer in Computer Science

Email: Klaus.Havelund@jpl.nasa.gov
Personal web: <http://www.havelund.com>
Cell Phone: (+1)650-814-0677

Citizenship: Denmark, US permanent resident (green card)
Born the 17th of October 1955 in Copenhagen, Denmark

Education

1991–94 Ph.D in Computer Science from the University of Copenhagen,
Prepared at **Ecole Normale Supérieure, Paris.**

“The Fork Calculus - Towards A Logic for Concurrent ML”.

Development of a process algebra in the CCS family with the objective to study a specification language for the Concurrent ML (CML) programming language (details p.2).

1986 Master Thesis in Computer Science from the University of Copenhagen.

“Stepwise Development of a Denotational Stack Semantics”.

Study of the relationship between abstract and operational descriptions of programming languages (details p.3).

Professional Experience

July 2006 – present NASA's Jet Propulsion Laboratory, Pasadena, Los Angeles, USA.
From July 2007 employed by California Institute of Technology (Caltech) as a *principal* computer scientist. From July 2006 - July 2007 contracting through Columbus Technologies (details p.3).

Mar 2005 – July 2006 Kestrel Technology, California, USA (details p.3).

Apr 1997 – February 2005 Researcher at NASA Ames Research Center, California, USA. Contracting through Recom Technologies 1997-2001 and through Kestrel Technology 2001-2005 (details p.4).

Sep 1996–Mar 1997 Researcher at the Department of Computer Science, Aalborg University, Denmark (details p.4).

Oct 1994–Jul 1996 Researcher on a HCM (Human Capital Mobility) grant financed by the European Community. Research Lab: LITP, Paris 6, France (details p.4).

Jan–Oct 1994 Post-Doc at Ecole Polytechnique, Paris, France (details p.5).

1988–91 Researcher at CRI (advanced Danish software company) within the RAISE formal methods project, which in 1988 was transferred from DDC, see below (details p.5).

1984–88 Researcher at the industrial research institute DDC (Danish Datamatics Center) and member of the European ESPRIT project RAISE, the purpose of which was to develop a formal specification language (details p.5).

1979–82 Software programmer in various companies (half time during my university studies).

1 University Studies

1991–94 Ph.D in Computer Science from DIKU, Denmark

The Ph.D work was carried out from February 1991 to December 1993, and was defended March 1994 at DIKU, University of Copenhagen, Denmark. During this period I spent more than two years at Ecole Normale Supérieure, Paris, France.

The supervisor was Klaus Grue (DIKU). The members of the jury were Klaus Grue, Kim Guldstrand Larsen (AUC, Denmark) and Mogens Nielsen (DAIMI, Denmark).

The main motivation for writing a Ph.D was to extend the work I had carried out during 6 years in the European ESPRIT project RAISE.

The thesis, written in English, has as title: *“The Fork Calculus—Towards a Logic for Concurrent ML”*. The Fork Calculus, FC, presents a theory for dynamic process creation where processes interact through hand-shake communication. This calculus differs from Robin Milner’s CCS in the way that processes are put in parallel. In CCS there exists a binary parallel operator $|$ with which two processes p and q may be put in parallel as $p|q$. In FC this binary operator has been replaced with a unary `fork`-operator, and a process p can be activated to execute in the background, in parallel with the remaining program, with the command `fork(p)`. Also, FC has sequential composition instead of the action prefixing of CCS.

After having defined the syntax for FC, I have constructed an operational semantics, and based on that, I have studied various bisimulations, including a complete axiomatization of one of these. Two extensions of this calculus are then studied, one of which deals with program refinement, and one of which deals with dynamic process configuration as found in Robin Milner’s π -calculus. For each of these three calculi I have defined a Hennessy-Milner like modal logic.

The three calculi shall be seen as approximations to defining a refinement logic for the programming language CML (Concurrent ML). CML is an extension of the programming language ML with concurrency primitives, amongst them a `fork`-operator. The thesis ends with an outline of such a logic for CML.

Part of the work has been carried out in collaboration with Kim Guldstrand Larsen, Aalborg University (AUC), Denmark. Hence, K. Larsen is co-author on published papers. The references from my thesis work are the following: [12, 50, 76]

1986 Master Thesis in Computer Science from DIKU, Denmark

The topic of the thesis [75] was denotational semantics of programming languages with professor Neil D. Jones as supervisor. I got the note : 9 out of 10 – corresponding to 11 on the Danish scale. The goal of the project was to bridge the gap between an abstract semantic definition of an Algol-like programming language, and a concrete operational definition of the same language. The bridge was created by a series of four intermediate semantics, getting more and more concrete. In particular the work revealed systematically the distinction between static (compile time) and dynamic (runtime) semantics.

During my studies I also wrote a syntax checker for the Meta-IV language, the specification language of the formal method VDM. This together with the above denotational semantics project was my real introduction to the area of formal methods that I have then stayed within for more than a decade now.

In general, the education at DIKU is planned as a five year study, with three years broad introduction to fundamental areas of computer science, followed by two years of more advanced topics, including the master thesis.

2 Professional Experience

July 2006 – present Jet Propulsion Laboratory, Los Angeles, USA

At JPL I work as a *principal* in the LaRS group (Laboratory for Reliable Software), employed by California Institute of Technology (Caltech) since 2007. From July 2006 - July 2007 I was contracting for JPL through Columbus Technologies. The group's purpose is to support JPL missions in producing reliable software. The group produces tool support for static as well as dynamic analysis. The group also tries to engage directly in the development of mission critical software, applying advanced techniques and principles to achieve high levels of reliability. My personal interest is dynamic analysis, also referred to as runtime verification. I am currently working on the tool ARMOR for monitoring C programs (the most used programming language at JPL for safety critical flight software). I have also as part of a collaboration between JPL and Kennedy Space Center Florida developed (in Python) a system for monitoring telemetry emitted from the space shuttle launch platform. Finally, I have applied model checking in various contexts, including modeling a complex flight algorithm and writing a translator from the ASPEN plan-model language to the modeling language of the SPIN model checker, the purpose being to use SPIN to verify plan-models.

Mar 2005– July 2006 Researcher at Kestrel Technology, California, USA

Worked on two NASA projects: Model-Centric Safety-Critical Java for Exploration (MXJ), the PI was Kestrel Institute, and Reliable Software Systems Development (RSSD), the PI

was JPL (Gerard Holzmann). My main activity was the development of a specification language as an extension of Java for MXJ, and development of a C runtime monitoring system for RSSD. I also worked on a static analysis specification language.

1997–2005 Researcher at NASA Ames Research Center, California, USA

My activities concentrated on program verification and testing, and the goal was to develop techniques for locating errors in parallel programs.

I performed one of the more successful applications of model checking: the analysis of the Remote Agent for the Deep-Space 1 space craft, where we found errors that although corrected, later were re-introduced in other parts of the software, causing deadlock in flight. I furthermore engaged in projects involved with verification of planning systems, including the organization of VVPS'05, the 1st International Workshop on Verification and Validation of Model-Based Planning and Scheduling Systems, Monterey, June 2005.

I conceptualized and started the the Java PathFinder project, which thanks to colleagues today is the major project in the Automated Software Engineering group and has won two awards. Although I am not developer of the current system I started the project, and made the first prototype, which translated a substantial subset of Java into Promela, the modeling language of the SPIN model checker. This is pioneering work that has motivated other researchers on the international scene.

I worked on runtime verification where program executions are monitored and checked against requirement specifications. I started a series of runtime verification workshops, as documented on <http://www.runtime-verification.org>.

Sep 1996–Mar 1997 Researcher at Aalborg University, Denmark

I here worked in Kim Guldstrand Larsen's formal methods group. This group is known for their expertise in model checking, and in particular real-time model checking. During this stay, I applied the real-time model checker UPPAAL (developed partly by this group) to a 10 year old real-life audio/video protocol from the Audio/Video company Bang & Olufsen. During this effort I spotted the source of a known error, which had been around throughout all those years without being identifiable by normal testing.

Oct 1994–Jul 1996 HCM Grant, Paris 6 University, France

I was here financed by a HCM (Human Capital Mobility) grant from the European Community to do research in concurrency verification. My work in Therese Hardin's group was focused on formal specification and verification of concurrent real-time systems, for example communication protocols. Special emphasis was put on combining theorem proving in classical typed higher order logic with theorem proving in temporal logic. This was done basically by embedding TLA ('Temporal Logic of Actions' developed by Leslie Lamport) into the general purpose theorem prover PVS ('Prototype Verification System') developed by Owre, Shankar and Rushby at SRI International, California. A branch of this work consisted of combining theorem proving and model checking.

I collaborated with the people at SRI, in particular with Natarajan Shankar. A result of

this collaboration is the paper [53] presented at the *Formal Methods Europe* conference at Oxford in March 1996. I have spent more than 5 months at SRI over the last 18 months.

As an additional result of this collaboration, I initiated a visit to Paris 6 by John Rushby, where he gave a one day PVS course for academics and industrial people. About 70 persons attended. My own contribution was to provide a “hands-on” practical exercise in using PVS on computers after the course of John Rushby. As a result of my stay at Paris 6, PVS is now used there in research (including a Ph.D student) and will soon be used in teaching. Also, the PVS system is now available via ftp from Paris 6, which then has become one of 3 European internet sites providing PVS in addition to SRI in California.

Jan–Oct 94 Post-Doc at Ecole Polytechnique, Paris

I worked as a post-doc in Radhia Cousot’s group, financed by Ecole Polytechnique. During this period I learned about the theorem prover PVS, and considered how it could be used to specify and verify parallel systems.

1988–91 Researcher at CRI — the RAISE Project, Denmark

In 1988 the Danish software house CRI took over parts of the activities of DDC, amongst these the European ESPRIT project RAISE, see below. Hence, as participant of this project, and at that time employed by DDC, I was transferred from DDC to CRI. In general CRI was involved in several European ESPRIT projects, and in the European Space Agency’s programs. Hence, an inspiring international environment.

1984–1988 Researcher at DDC — The RAISE Project, Denmark

DDC was an industrially oriented research institution, with main activities within ESPRIT projects (European research program for information technology). The initiator and scientific chief of the institute was Professor Dines Bjørner – DTU (Technical University of Denmark), currently director of the United Nations University for software technology in Macau.

I was working for 6 years as scientific staff in ESPRIT project 315: RAISE. RAISE stands for ‘Rigorous Approach to Industrial Software Engineering’. The purpose of the project was to produce a formally (mathematically) based method for producing software. The overall goal was to combine VDM (‘Vienna Development Method’ developed by Dines Bjørner and Cliff Jones) with CSP (‘Communicating Sequential Processes’ developed by Hoare). The major teams of the project were the Danish DDC team and the British STC (Standard Telephones and Cables) team. However, also ABB and ICL were involved in the project. The project covered approximately 100 man years, with the partition between Danish and British effort being around 50:50.

The project had a number of internationally known computer scientists associated as consultants, who continuously followed the project. These were Manfred Broy, Cliff Jones, Don Sannella and Andrzej Tarlecki.

My main responsibilities throughout the 6 years were language design and semantics of the resulting language. The language design was carried out by a group of 10 people, and

was based on case studies generated by the true industrial partners (STC, ABB, ICL). After the language design, I wrote the final language semantics together with Robert Milne during a period of approximately 8 months. The last I did in the project was to write the majority (85%) of the textbook (published by Prentice-Hall) which explains the RAISE specification language. That is, I wrote *all* of the 250 page tutorial, and half of the 100 page reference manual. See [1]. This book is used today as a general introduction to formal methods at the Technical University of Denmark. Other RAISE references are [9, 6].

I have written several reports during the project, often in collaboration with other members of the team, and some of these were so-called deliverables to the European Commission.

The RAISE product (a specification language and associated tools) has been tested in the 5 year follow up ESPRIT project 5383: LaCoS, which involved several companies in several European countries: CRI, BNR Europe, SYPRO, Bull, MATRA Transport, INISEL Espacio, SSI, Technisystems and Lloyd's Register of Shipping.

See information about RAISE on the World Wide Web on the address:

"<http://dream.dai.ed.ac.uk/raise>". I was not involved in LaCoS since I decided to do a Ph.D directly after the end of the RAISE project.

3 Invited Presentations

- ASM'03, International Workshop on Abstract State Machines, Italy, March 2003.
- Joint CAV/ISSTA Special Event on Specification, Verification, and Testing of Concurrent Software. Boston, USA, July 2004.
- Danish industry, arranged by CISS (Centre for Embedded Software Systems), Aalborg, Denmark, December 2004.
- CASSIS'05, Construction and Analysis of Safe, Secure and Interoperable Smart devices, Nice, France, March 2005.
- VVEIS'05, The 3rd International Workshop on Verification and Validation of Enterprise Information Systems, Miami, Florida, May 2005.
- 8th JPL-GSFC Quality Mission Software Workshop, Santa Barbara, California, May 2006.
- PADTAD'06 (Parallel and Distributed Systems: Testing and Debugging). July 17 2006, Portland, Maine, USA.
- ARTIST2 Summer School. September 8-12, 2008, Autrans (near Grenoble), France.

4 Organized Workshops and Events

- The 7th International SPIN workshop in year 2000 (SPIN'00) at Stanford University, California. The workshop lasted 3 days and was stand-alone (was not associated to a bigger conference).

- The 1st International Workshop on Verification and Validation of Model-Based Planning and Scheduling Systems: <http://planning.cis.strath.ac.uk/vvpsws>.
- Started together with Grigore Rosu a sequence of so far eight workshops (seven have occurred and one is planned) on Runtime Verification (RV – starting with RV’01 in 2001 and including RV’08 in 2008), as documented on <http://www.runtime-verification.org>. I belong to the steering committee. Of the eight workshops I co-organized and co-chaired the following four:
 1. RV’01 - held as a CAV’01 satellite event in Paris, France, July 2001.
 2. RV’02 - held as a CAV’02 satellite event in Copenhagen, Denmark, July 2002.
 3. RV’04 - held as an ETAPS’04 satellite event in Barcelona, Spain, April 2004.
 4. FATES/RV’06 - held as a FLoC’06 event in Seattle, USA, August 2006

The remaining four workshops (RV’03 with CAV in Colorado, RV’05 with CAV in Edinburgh, RV’07 with AOSD in Vancouver, and RV’08 with ETAPS in Budapest) were or are organized by others that the RV steering committee, of which I am member, appointed. I feel partly responsible for the regular top level management of this workshop series and all its events.

- Dagstuhl meeting on runtime verification, January 3-6, 2007. See <http://www.dagstuhl.de/07011>.
- The 15th International SPIN Workshop on Model Checking of Software in Los Angeles, California (stand-alone event), August 10-12, 2008. Program co-chair. See <http://compilers.cs.ucla.edu/spin08>.

5 Program Committee Memberships

As a Program Committee (PC) member for a particular event one must review and select papers for presentation at that event (workshop, conference).

- VMCAI 2009, The 10th Intl. Conference on Verification, Model Checking, and Abstract Interpretation, Savannah, GA, January 18-20, 2009.
- SEFM 2008, 6th IEEE International Conference on Software Engineering and Formal Methods. November 10-14, 2008, Cape Town, South Africa.
- HVC 2008, The IBM Verification Conference 2008, Haifa, Israel, October 28-30, 2008.
- ICSEA 2008, The Third International Conference on Software Engineering Advances Sliema, Malta, October 26-31, 2008.
- SLE 2008, The First International Conference on Software Language Engineering, Toulouse, France, September 29-30, 2008.
- MSVVEIS 2008, The Sixth International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Barcelona, Spain, 2008.

- PADTAD 2008, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging, Seattle, Washington, USA, July 20-21, 2008.
- SAW 2008, ACM SIGPLAN Static Analysis Workshop (Co-located with PLDI 2008). Tucson, Arizona, June 12, 2008.
- AOSD 2008, Seventh International Conference on Aspect-Oriented Software Development. Brussels, Belgium, March 31 - April 4, 2008.
- CONFENIS 2007, the IFIP International Conference on Research and Practical Issues of Enterprise Information Systems. October 14-16, 2007, Beijing, China.
- The IBM Verification Conference 2007. October 23 - 25, 2007. Organized by IBM Research Lab in Haifa, Israel.
- SEFM 2007, 5th IEEE International Conference on Software Engineering and Formal Methods. September 10-14, 2007, London, UK.
- ICSEA 2007, the Second International Conference on Software Engineering Advances August 25-31, 2007, Cap Esterel, French Riviera, France.
- PADTAD 2007, Parallel and Distributed Systems: Testing and Debugging. In conjunction with the International Symposium on Software Testing and Analysis (ISSTA), July 09, 2007, London, England.
- MSVVEIS 2007, the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, 12-13 June, 2007, Funchal, Madeira - Portugal.
- The IBM Verification Conference 2006. October 23 - 26, 2006. Organized by IBM Research Lab in Haifa, Israel.
- ICSEA 2006, International Conference on Software Engineering Advances.
- SEFM 2006, 4th IEEE International Conference on Software Engineering and Formal Methods.
- FMICS 2006, 11th International Workshop on Formal Methods for Industrial Critical Systems.
- PADTAD 2006, Parallel and Distributed Systems: Testing and Debugging.
- MSVVEIS 2006, The Fourth International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems.
- TACAS 2006, The 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems.
- RV 2005, The 5th International Workshop on Runtime Verification.
- VVEIS 2005, The 3rd International Workshop on Verification and Validation of Enterprise Information Systems.
- SEFM 2005, 3rd IEEE International Conference on Software Engineering and Formal Methods.

- FATES 2005, Formal Approaches to Testing of Software.
- ICI 2004, International Conference on Informatics.
- CAV 2004, The 16th Conference in Computer Aided Verification.
- DAW 2004, Dynamic Aspects Workshop.
- SFEDL 2004, Semantic Foundations of Engineering Design Languages.
- REOS 2003, Workshop on Requirements Engineering and Open Systems.
- ASARTI 2003, Workshop: Advancing the State-of-the-Art in Run-Time Inspection.
- FMPPTA 2003, The 8th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
- DSN'03, Workshop on Model Checking for Dependable Software-Intensive Systems.
- ESEC/FSE'03, 4th joint meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering.
- CAV 2002, The 14th Conference in Computer Aided Verification.
- RT-TOOLS, 2002 Workshop on Real-Time Tools.
- FMPPTA 2002, The 7th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
- SPIN 2002, The 9th International SPIN Workshop on Model Checking of Software.
- SPIN 2001, The 8th International SPIN Workshop on Model Checking of Software.
- FMPPTA 2001, 6th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
- MVI 2001, Model-based Validation of Intelligence.
- Lfm 2000, Fifth NASA Langley Formal Methods Workshop.
- PATV 2000, The First International Workshop on Automated Program Analysis, Testing and Verification.
- JFLA 2000, Les onzimes Journées Francophones des Langages Applicatifs.

References

Books and Book Chapters

- [1] The RAISE Language Group. *The RAISE Specification Language*. The BCS Practitioner Series. Prentice Hall, 1992. The book consists of two parts I: RSL Tutorial (pages 9-250), and II: RSL Reference Description (pages 251-369). Havelund wrote part I, while Havelund together with Anne Haxthausen wrote part II. The language presented in the book is designed by the *RAISE Language Group*: Chris George, Peter Haff, Klaus Havelund, Anne Haxthausen, Robert Milne, Claus Bendix Nielsen, Soeren Prehn and Kim Ritter Wagner.

- [2] Klaus Havelund. *RAISE in Perspective*. Invited chapter in *Logics of Specification Languages*. Edited by Dines Bjørner and Martin Henson. Monographs in Theoretical Computer Science. An EATCS Series, 2007, 624 p., Hardcover.

Journal Papers

- [3] Cyrille Artho, Howard Barringer, Allen Goldberg, Klaus Havelund, Sarfraz Khurshid, Mike Lowry, Corina Pasareanu, Grigore Rosu, Koushik Sen, Willem Visser, and Rich Washington. Combining Test-Case Generation and Runtime Verification. *Journal of Theoretical Computer Science*, 336(2-3), May 2005. Extended version of [19].
- [4] Cyrille Artho, Klaus Havelund, and Armin Biere. High-Level Data Races. *Software Testing, Verification and Reliability*, 13(4), 2004. Extended version of [22]. To appear.
- [5] H. Barringer, D. Rydeheard and K. Havelund. Rule Systems for Run-Time Monitoring: from Eagle to RuleR. Journal version of paper accepted at RV'07. Special issue of the Journal of Logic and Computation (JLC). To Appear.
- [6] D. Bjørner, A. Haxthausen, and K. Havelund. Formal, Model-oriented Software Development Methods: From VDM to ProCoS and from RAISE to LaCoS. *Future Generation Computer Systems*, 7, 1992.
- [7] Guillaume Brat, Doron Drusinsky, Dimitra Giannakopoulou, Allen Goldberg, Klaus Havelund, Mike Lowry, Corina Pasareanu, Willem Visser, and Rich Washington. Experimental Evaluation of Verification and Validation Tools on Martian Rover Software. *Formal Methods in System Design*, 25(2), 2004. Journal version of [31].
- [8] Yaniv Eytani, Klaus Havelund, Scott Stoller, and Shmuel Ur. Toward a Framework and Benchmark for Testing Tools for Multi-Threaded Programs. *Journal of Concurrency and Computation: Practice and Experience - to appear*, 2005. Wiley.
- [9] Chris George, Klaus Havelund, Mogens Nielsen, and Kim Ritter Wagner. The RAISE Language, Method and Tools. *Formal Aspects of Computing*, 1(1), January-March 1989. Journal version of [68].
- [10] K. Havelund and T. Pressburger. Model Checking Java Programs using Java PathFinder. *International Journal on Software Tools for Technology Transfer*, 2(4):366–381, April 2000. Special issue of STTT containing selected submissions to the 4th SPIN workshop, Paris, France, 1998.
- [11] K. Havelund and W. Visser. Program Model Checking as a New Trend. *International Journal on Software Tools for Technology Transfer*, 4(1), October 2002. Special issue of STTT containing selected submissions to the 7th SPIN workshop, Stanford, USA, 2000, organized by K. Havelund, J. Penix and W. Visser. The article is the introductory article to the special issue. Havelund and Visser are guest editors.
- [12] Klaus Havelund and Kim Guldstrand Larsen. The Fork Calculus. *Nordic Journal of Computing*, 1, 1994. Journal version of [58].
- [13] Klaus Havelund, Mike Lowry, and John Penix. Formal Analysis of a Space Craft Controller using SPIN. *IEEE Transactions on Software Engineering*, 27(8):749–765, August 2001. Journal version of [60].

- [14] Klaus Havelund and Grigore Roşu. An Overview of the Runtime Verification Tool Java PathExplorer. *Formal Methods in System Design*, 24(2), March 2004. Extended journal version of [62].
- [15] Klaus Havelund and Grigore Roşu. Efficient Monitoring of Safety Properties. *International Journal on Software Tools for Technology Transfer (STTT)*, 6(2):158–173, 2004. Extended journal version of [52].
- [16] Grigore Roşu and Klaus Havelund. Rewriting-Based Techniques for Runtime Verification. *International Journal on Automated Software Engineering*, 12(2):151–197, 2005. Substantially expanded version of [63].
- [17] Willem Visser, Klaus Havelund, Guillaume Brat, Seungjoon Park, and Flavio Lerda. Model Checking Programs. *International Journal on Automated Software Engineering*, 10(2), April 2003. Extended journal version of [73].

Conference and Workshop Papers

- [18] J. Anton, E. Bush, A. Goldberg, K. Havelund, D. Smith and A. Venet. Towards the Industrial Scale Development of Custom Static Analyzers. *Static Analysis Summit*, U.S. National Institute of Standards and Technology, Gaithersburg, MD, USA, June 29, 2006.
- [19] C. Artho, D. Drusinsky, A. Goldberg, K. Havelund, M. Lowry, C. Pasareanu, G. Roşu, and W. Visser. Experiments with Test Case Generation and Runtime Analysis. In E. Börger, A. Gargantini, and E. Riccobene, editors, *Abstract State Machines (ASM’03)*, volume 2589 of *LNCS*, pages 87–107. Springer, March 2003.
- [20] C. Artho, K. Havelund and S. Honiden. Visualization of Concurrent Program Executions. Workshop on Software Architectures and Component Technologies (SACT’07), Beijing, China, July 23, 2007.
- [21] C. Artho and K. Havelund. Applying Jlint to Space Exploration Software. In B. Steffen and G. Levi, editors, *Proceedings of Fifth International VMCAI conference (VMCAI’04)*, volume 2937 of *LNCS*. Springer, January 2004.
- [22] Cyrille Artho, Klaus Havelund, and Armin Biere. High-Level Data Races. In *VVEIS’03, The First International Workshop on Verification and Validation of Enterprise Information Systems*, April 2003. Angers, France.
- [23] Cyrille Artho, Klaus Havelund, and Armin Biere. Using Block-Local Atomicity to Detect Stale-Value Concurrency Errors. In *2nd International Symposium on Automated Technology for Verification and Analysis (ATVA’04), Taiwan*, October–November 2004. To appear.
- [24] H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Program Monitoring with LTL in Eagle. In *Proceedings of Workshop on Parallel and Distributed Systems: Testing and Debugging (PADTAD’04)*, April 2004.
- [25] H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Rule-Based Runtime Verification. In B. Steffen and G. Levi, editors, *Proceedings of Fifth International VMCAI conference (VMCAI’04)*, volume 2937 of *LNCS*. Springer, January 2004.

- [26] H. Barringer, D. Rydeheard and K. Havelund. Rule Systems for Run-Time Monitoring: from Eagle to RuleR. 7th International Workshop on Runtime Verification (RV'07), Vancouver, Canada, March 13, 2007. To appear in LNCS.
- [27] M. Bennett, R. Borgen, K. Havelund, M. Ingham and David Wagner. Development of a Prototype Domain-Specific Language for Monitor and Control Systems. IEEE Aerospace Conference, Big Sky, Montana, March 1-8, 2008.
- [28] Saddek Bensalem, Jean-Claude Fernandez, Klaus Havelund, and Laurent Mounier. Confirmation of Deadlock Potentials Detected by Runtime Analysis. In *Parallel and Distributed Systems: Testing and Debugging (PADTAD'06)*, July 2006. Portland, Maine, USA.
- [29] Saddek Bensalem and Klaus Havelund. Dynamic Deadlock Analysis of Multi-Threaded Programs. In *Parallel and Distributed Systems: Testing and Debugging (PADTAD'05)*, November 2005. Haifa, Israel.
- [30] Eric Bodden and Klaus Havelund. Racer: Effective Race Detection Using AspectJ. In *Proceedings of International Symposium on Software Testing and Analysis (ISSTA'08)*, Seattle, WA, July 20-24, 2008. Won ISSTA's ACM SIGSOFT Distinguished Paper Award.
- [31] Guillaume Brat, Dimitra Giannakopoulou, Allen Goldberg, Klaus Havelund, Mike Lowry, Corina Pasareanu, Arnaud Venet, and Willem Visser. Experimental Evaluation of Verification and Validation Tools on Martian Rover Software. In *CMU/SEI Software Model Checking Workshop*, March 2003. Pittsburg, USA.
- [32] W. Buntine, B. Fischer, K. Havelund, M. Lowry, T. Pressburger, S. Roach, P. Robinson, and J. V. Baalen. Transformation Systems at NASA Ames. In *Software Transformation Systems*, May 1999. Los Angeles, California, USA.
- [33] Marcelo d'Amorim and Klaus Havelund. Event-Based Runtime Verification of Java Programs. In *Workshop on Dynamic Analysis (WODA'05)*, May 2005. St. Louis, Missouri, USA.
- [34] Doron Drusinsky and Klaus Havelund. Execution-Based Model Checking of Interrupt-Based Systems. In *Workshop on Model Checking for Dependable Software-Intensive Systems*, June 2003. San Francisco, CA, USA. Affiliated with DSN'03, the International Conference on Dependable Systems and Networks.
- [35] R. E. Filman and K. Havelund. Realizing Aspects by Transforming for Events. In *Proceedings, International Conference on Automated Software Engineering (ASE'02)*. Institute of Electrical and Electronics Engineers, 2002. Edinburgh, Scotland, 23-27 September 2002.
- [36] R. E. Filman and K. Havelund. Source-Code Instrumentation and Quantification of Events. In Gary T. Leavens and Ron Cytron, editors, *Foundations Of Aspect-Oriented Languages, FOAL 2002, Enschede, The Netherlands*, April 2002. Proceedings available at <http://www.cs.wustl.edu/~cytron/FOAL>.
- [37] Robert Filman and Klaus Havelund. The Effect of AOP on Software Engineering, with Particular Attention to OIF and Event Quantification. In *Workshop on Software*

Engineering Properties of Languages for Aspect Technologies (SPLAT'03), March 2003. Boston, USA.

- [38] D. Giannakopoulou and K. Havelund. Automata-Based Verification of Temporal Properties on Running Programs. In *Proceedings, International Conference on Automated Software Engineering (ASE'01)*, pages 412–416. Institute of Electrical and Electronics Engineers, 2001. Coronado Island, California, 26-29 November 2001.
- [39] Allen Goldberg and Klaus Havelund. Instrumentation of Java Bytecode for Runtime Analysis. In *Fifth ECOOP Workshop on Formal Techniques for Java-like Programs (FTfJP'03)*, July 2003. Darmstadt, Germany.
- [40] Allen Goldberg and Klaus Havelund. Automated Runtime Verification with Eagle. In *Workshop on Verification and Validation of Enterprise Information Systems (VVEIS'05)*, May 2005. Miami, Florida, USA.
- [41] Allen Goldberg and Klaus Havelund and Conor McGann. Runtime Verification for Autonomous Space Craft Software. In *IEEE Aerospace Conference, Big Sky, Montana, USA, March 2005*, October 2004.
- [42] K. Havelund. Mechanical Verification of a Garbage Collector. In D. Méry and B. Sanders, editors, *FMPPTA'99: Fourth International Workshop on Formal Methods for Parallel Programming : Theory and Applications*, number 1586 in Lecture Notes in Computer Science, pages 1258–1283. Springer-Verlag, April 1999. San Juan, Puerto Rico, USA.
- [43] Klaus Havelund. Java PathFinder, A Translator from Java to Promela (Abstract). In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 6th Workshop).
- [44] Klaus Havelund. Using Runtime Analysis to Guide Model Checking of Java Programs. In Klaus Havelund, John Penix, and Willem Visser, editors, *SPIN Model Checking and Software Verification*, volume 1885 of *Lecture Notes in Computer Science*, pages 245–264. Springer, 2000.
- [45] Klaus Havelund, Alex Groce, Gerard Holzmann, Rajeev Joshi, and Margaret Smith. Automated Testing of Planning Models. In K. Suzuki, T. Higashino, A. Ulrich, T. Hasegawa, editors, *MOCHART'08: The Fifth International Workshop on Model Checking and Artificial Intelligence*, Lecture Notes in Computer Science, to appear, Springer-Verlag. Patras, Greece, July 2008.
- [46] K. Havelund. Runtime Verification of C Programs. Abstract. *1st Dagstuhl Seminar on Runtime Verification, Schloss Dagstuhl, Germany, January 3–6, 2007*.
- [47] K. Havelund. Runtime Verification of C Programs. TestCom/FATES conference (Testing of Communicating Systems and Formal Approaches to Testing of Software), volume 5047 of *Lecture Notes in Computer Science*, pages 7–22. Springer. Tokyo, Japan, June 10–13, 2008.

- [48] K. Havelund, A. Goldberg, R. E. Filman, and G. Roşu. Program Instrumentation and Trace Analysis. In *Monterey Workshop 2002, Radical Innovations of Software and Systems Engineering in the Future, Venice, Italy*, October 2002. Invited participation. One page abstract.
- [49] K. Havelund, S. Johnson, and G. Roşu. Specification and Error Pattern Based Program Monitoring. In *European Space Agency Workshop on On-Board Autonomy, Noordwijk, Holland.*, pages 323–330. European Space Agency, October 2001.
- [50] K. Havelund and K. G. Larsen. A Refinement Logic for the Fork Calculus. In S. T. Vuong and S. T. Chanson, editors, *Protocol Specification, Testing and Verification XIV*, IFIP WG 6.1 Symposium. Chapman and Hall, 1995. Vancouver, Canada.
- [51] K. Havelund, K. G. Larsen, and A. Skou. Formal Verification of an Audio/Video Power Controller using the Real-Time Model Checker UPPAAL. In *5th Int. AMAST Workshop on Real-Time and Probabilistic Systems*, number 1601 in Lecture Notes in Computer Science. Springer-Verlag, May 1999. Bamberg, Germany.
- [52] K. Havelund and G. Roşu. Synthesizing Monitors for Safety Properties. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2002, Grenoble, France*, Electronic Notes in Theoretical Computer Science, pages 342–356. Springer, April 2002. **Won EASST award for best software science paper presented at ETAPS’02.**
- [53] K. Havelund and N. Shankar. Experiments in Theorem Proving and Model Checking for Protocol Verification. In M-C. Gaudel and J. Woodcock, editors, *FME’96: Industrial Benefit and Advances in Formal Methods*, volume 1051 of *Lecture Notes in Computer Science*, pages 662–681. Springer-Verlag, 1996.
- [54] K. Havelund and J. Skakkebæk. Applying model checking in Java verification. In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 6th Workshop).
- [55] K. Havelund, A. Skou, K. G. Larsen, and K. Lund. Formal Modeling and Analysis of an Audio/Video Protocol: An Industrial Case Study Using UPPAAL. In *Proc. of the 18th IEEE Real-Time Systems Symposium*, pages 2–13, Dec 1997. San Francisco, California, USA.
- [56] K. Havelund, S. Stoller, and S. Ur. Benchmark and Framework for Encouraging Research on Multi-Threaded Testing Tools. In *PADTAD’03, Parallel and Distributed Systems: Testing and Debugging*, April 2003. Nice, France. Invited paper.
- [57] Klaus Havelund and Allen Goldberg. Verify Your Runs. In *Verified Software: Theories, Tools, Experiments (VSTTE’05)*, October 2005. Zurich, Switzerland.
- [58] Klaus Havelund and Kim Guldstrand Larsen. The Fork Calculus. In *20th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 700 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993. A preliminary version occurred in proceedings of 4th Nordic Workshop on Program Correctness, Report no 78, University of Bergen, 1993.

- [59] Klaus Havelund, Mike Lowry, Seungjoon Park, Charles Pecheur, John Penix, Willem Visser, and John L. White. Formal Analysis of the Remote Agent, Before and After Flight. In *Fifth NASA Langley Formal Methods Workshop, Virginia*, June 2000.
- [60] Klaus Havelund, Mike Lowry, and John Penix. Formal Analysis of a Space Craft Controller using SPIN. In Gerard Holzmann, Elie Najm, and Ahmed Serhrouchni, editors, *4th Workshop on Automata Theoretic Verification with the SPIN Model Checker (SPIN'98)*, pages 147–168. Ecole Nationale Supérieure des Télécommunications, November 1998. Paris, France.
- [61] Klaus Havelund and Grigore Roşu. Java pathexplorer - a runtime verification tool. In *The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS'01)*, 2001. Montreal, Canada.
- [62] Klaus Havelund and Grigore Roşu. Monitoring java programs with java pathexplorer. In Klaus Havelund and Grigore Roşu, editors, *Proceedings of the First Workshop on Runtime Verification (RV'01), Paris, France, July 2001*, volume 55 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science, 2001.
- [63] Klaus Havelund and Grigore Roşu. Monitoring programs using rewriting. In *Proceedings, International Conference on Automated Software Engineering (ASE'01)*, pages 135–143. Institute of Electrical and Electronics Engineers, 2001. Coronado Island, California, 26-29 November 2001.
- [64] Klaus Havelund and Eric Van Wyk. Aspect-Oriented Monitoring of C Programs. In *Proceedings of the Sixth IARP-IEEE/RAS-EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments*, Pasadena, CA, May 17-18, 2008.
- [65] L. Khatib, N. Muscettola, and K. Havelund. Verification of Plan Models using UP-PAAL. In *First Goddard Workshop on Formal Approaches to Agent-Based Systems*, March 2000. NASA's Goddard Space center, Maryland, March 2000.
- [66] Lina Khatib, Nicola Muscettola, and Klaus Havelund. Mapping Temporal Planning Constraints into Timed Automata. In *The Eighth International Symposium on Temporal Representation and Reasoning (Time'01)*, IEEE press, June 2001. Cividale Del Friuli, Italy.
- [67] M. Lowry, K. Havelund, and J. Penix. Verification and Validation of AI Systems that Control Deep-Space Spacecraft. In Z.W. Ras and A. Skowron, editors, *Tenth International Symposium on Methodologies for Intelligent Systems*, number 1325 in *Lecture Notes in Artificial Intelligence*. Springer-Verlag, October 1997. Charlotte, North Carolina.
- [68] Mogens Nielsen, Klaus Havelund, Kim Ritter Wagner, and Chris George. The RAISE Language, Method and Tools. In *VDM - The Way Ahead*, volume 328 of *LNCS*, pages 376–405. Springer-Verlag, 1988.
- [69] J. Penix, P. Alexander, and K. Havelund. Declarative Specification of Software Architectures. In *Proc. 12th IEEE International Conference on Automated Software Engineering*, November 1997. Incline Village.

- [70] J. Penix, C. Pecheur, and K. Havelund. Using Model Checking to Validate AI Planner Domain Models. In *Proceedings of the 23rd Annual Software Engineering Workshop*, December 1998. NASA Goddard.
- [71] Margaret Smith and Klaus Havelund. Requirements Capture with RCAT. In *Proceedings of the 16th IEEE International Requirements Engineering Conference*, Barcelona, Spain, September 8-12, 2008.
- [72] W. Visser, K. Havelund, G. Brat, and S. Park. Java PathFinder - Second Generation of a Java Model Checker. In *Workshop on Advances in Verification (WAVE'00)*, Chicago, USA. July 2000. Workshop associated with CAV'00.
- [73] W. Visser, K. Havelund, G. Brat, and S. Park. Model Checking Programs. In *Proceedings of ASE'2000: The 15th IEEE International Conference on Automated Software Engineering*. IEEE CS Press, September 2000.
- [74] W. Visser, K. Havelund, and J. Penix. Adding Active Objects to SPIN. In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 5th Workshop).

Theses

- [75] K. Havelund. Stepwise Development of a Denotational Stack Semantics. Master Thesis, DIKU, Department of Computer Science, University of Copenhagen, Denmark, 1984.
- [76] K. Havelund. *The Fork Calculus – Towards a Logic for Concurrent ML*. PhD thesis, Institute for Computer Science – University of Copenhagen (DIKU), March 1994. DIKU technical report 94/4.