Curriculum Vitae

Dr. Klaus Havelund

Ph.D in Computer Science
Senior Research Scientist (SRS[1]) at NASA's Jet Propulsion Laboratory
California Institute of Technology (Caltech)
Email: Klaus.Havelund@jpl.nasa.gov
Personal website: http://www.havelund.com
JPL website: http://lars-lab.jpl.nasa.gov/people/havelund.html
Cell phone: (+1)818-239-2178

JPL is a NASA field center, and a Federally Funded Research and Development Center (FFRDC), managed by California Institute of Technology (Caltech).

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 February 2009: *Senior Research Scientist.* From July 2007 - January 2009: *principal* computer scientist. From July 2006 - June 2007: contracting through Columbus Technologies. (details p.3).

Mar 2005 – July 2006 Kestrel Technology, California, USA (details p.5).

[1]The SRS nomination is given to individuals that have demonstrated achievement comparable to that required for an appointment as a professor at a leading university. See: `http://scienceandtechnology.jpl.nasa.gov/community/srslist`.

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.5).

Sep 1996–Mar 1997 Researcher at the Department of Computer Science, Aalborg University, Denmark (details p.5).

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.5).

Jan–Oct 1994 Post-Doc at Ecole Polytechnique, Paris, France (details p.6).

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.6).

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.6).

1979–82 Software programmer in various Danish companies (part time during my university studies). These include Skibsteknisk Laboratorium in Lyngby (a Maritime ship simulation institute, where I developed an accounting system and a sensitive map for entering coordinates, both in FORTRAN); RC Datacenter in Copenhagen (a large data service center, where I worked on the member database for the Danish academics union, as well as an insurance company database, both in Algol); and Christian Rovsing in Herlev (an advanced data service center, where I developed a dental clinic service application in Pascal).

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: [24, 93, 128]

1986 Master Thesis in Computer Science from DIKU, Denmark

The topic of the thesis [127] 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 was 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. However, the average study time was 9 years (for a master!) ... for those who made it. At some point over 400 started each year and 12 came out in the other end.

2 Professional Experience

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

I am a *Senior Research Scientist* in the LaRS group (Laboratory for Reliable Software), employed by California Institute of Technology (Caltech). This position corresponds to Caltech tenure. From July 2006 - July 2007 I was contracting for JPL through Columbus Technologies and from 2007-2009 as a principal (Caltech employee). 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.

Current projects:

- ScalaFM: exploring the use of the Scala programming language for Modeling, e.g. hierarchical state charts.
- DejaVu: a first-order past time temporal logic and system using BDDs for monitoring event traces.
- nfer: a logic and system for inferring interval abstractions from event traces.

Past projects at JPL:

- The **K** wide-spectrum specification language. A textual language for SysML class diagrams with constraints. Specifications are translated to SMT for verification (See: `www.theklanguage.com`).
- TraceContract: a log analysis API in Scala.
- LogFire: an implementation of the rule-based RETE algorithm for runtime verification in Scala.
- Mcai 2.0: Model Checking and Abstract Interpretation. Also referred to as : Computational Modeling and Analysis for Complex Systems. NSF Expeditions project. My focus is inference of specifications from execution traces.
  See: `http://cmacs.cs.cmu.edu`.
- *Java Coding Standard*: Development of a JPL institutional coding standard for Java. A set of coding rules that Java programmers should follow.
- Ruler: in search for a satisfactory runtime verification logic – convenient to use, expressive, covering state machines a well as logic, efficient to monitor, and easy to implement.
- *Survivable Software*: A GCC-based framework for static analysis, program instrumentation, program monitoring and recovery (currently for the C programming language).
- LogScope: a tool for checking log files against temporal logic specifications. Implemented in Python.
- Armor: a CIL-based tool for instrumenting and monitoring C programs (the programming language used at JPL for flight software).
- Lsccs: Launch Site Command and Control System. A Python application for commanding and monitoring the space shuttle launch platform at Kennedy Space Center. The purpose was to explore the advantages and disadantages of a domain specific language versus an extension of Python for this task.
- Map: a translator from the ASPEN planning language (`http://aspen.jpl.nasa.gov`) to Promela, the modeling language of the SPIN model checker. The purpose was 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 [98] 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 [18, 14].

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 Shiping.

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 Memberships

- Member of the Association for Computing Machinery (ACM).
- Member of IFIP working group WG 1.9/2.15.
- On the editorial board for FoMaC: Transactions on Foundations for Mastering Change.
- Member of the External Expert Group for COEMS - Continuous Observation of Embedded Multicore Systems.
- On JPL's board for selection of JPL principles.
- Served on JPL's SRS (Senior Research Scientist) Council. A 10 (approximately) person council representing researchers/scientists at JPL.
- Served on the evaluation committee for the INRIA Research Theme: Programs, Verification and Proofs. Report finished May 2015. With Valeria de Paiva, Kathleen Fisher (Chair), Andrew Kennedy, Gerwin Klein, Rustan Leino, Claire Loiseaux, Alan Mycroft, and Luke Ong.

4 Awards

- The JPL Voyager award in recognition of research contributions, including publication record, tool development, and conference organization. August 2017.

- The paper "Monitoring Programs using Rewriting", authors: Klaus Havelund and Grigore Rosu, published at ASE 2001, won the *ASE 2016 Most Influential Paper award* (`http://ase-conferences.org/Mip.html`). This paper was one of our first papers on runtime verification.

- The LogFire tool won the offline track of CRV-2015, the 2nd International Competition on Runtime Verification (`https://www.cost-arvi.eu/?page_id=664`). Held in connection with RV'15, The 15th International Conference on Runtime Verification, September 22 - September 25, 2015 Vienna, Austria.

- The paper "Model Checking Programs", authors: Willem Visser, Klaus Havelund, Guillaume Brat, and SeungJoon Park, published at ASE 2000, won the *ASE 2014 Most Influential Paper award* (`http://ase-conferences.org/Mip.html`). This paper represents the Java PathFinder project at the time, and the prospects of model checking concrete programs rather than abstract models.

- The *JPL Mariner award* in recognition of significant and sustained efforts to establish a new tool-based checking capability for a broad range of coding standards (C, C++, and Java) at JPL. August 2011.

- *Best paper award* for the paper: *"Runtime Verification with State Estimation"*, Scott D. Stoller, Ezio Bartocci, Justin Seyster, Radu Grosu, Klaus Havelund, Scott A. Smolka, and Erez Zadok. Presented at The 2nd International Conference on Runtime Verification (RV 2011), San Francisco, California, USA, October 27-30, 2011.

- The *JPL Ranger award* for the development of a Java Coding Standard and its implementation as an automated code checker. July 2010.

- The *JPL Mariner award* for the successful delivery of the LogScope tool for MSL (Mars Science Laboratory). LogScope checks output log files against a formal specification and reports violations. The tool was delivered to the FIT [testing] team to support flight software testing. July 2009.

- *Outstanding Technology Development Award* for Java PathFinder (JPF), Federal Laboratory Consortium (FLC, see: `http://www.federallabs.org`) Far West Region Awards. July 2009.

- *Royal Academy of Engineering Distinguished Visiting Fellowship* at the University of Manchester during December 2008-April 2009.

- *ACM Distinguished Paper Award* for the paper: *"Racer: Effective Race Detection Using AspectJ"*, Eric Bodden and Klaus Havelund. Presented at the International Symposium on Software Testing and Analysis (ISSTA'08), Seattle, WA, July 2008.

- *NASA's Group Achievement Award* to the Launch Control System Proof-of-Concept team. For successful demonstration of KSC's Launch Control System Proof-of-Concept Architecture for the Constellation Program's Command, Control and Communication Project. Signed by NASA's Administrator, Michael D. Griffin, Washington DC, *"this eighth day of May Two Thousand Eight"*.

- *Award for contribution to a NASA Tech Brief article* : *"Automated Testing using Symbolic Execution and Temporal Monitoring"*, that highlights a NASA Ames innovation. September 2006.

- NASA Office of Aerospace Technology *Turning Goals Into Reality (TGIR) Engineering Innovation Award* for the Java PathFinder (JPF). June 2003.
  See `http://ti.arc.nasa.gov/story.php?id=76`.
- *EASST award for best software science paper* presented at ETAPS'02: *"Synthesizing Monitors for Safety Properties"*, Klaus Havelund and Grigore Rosu. Presented at the International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'02), Grenoble, France, April 2002.

5 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. Portland, Maine, USA, July 17 2006.
- ARTIST2 Summer School. Autrans (near Grenoble), France, September 8-12, 2008.
- Talk for 150 10-11 graders (Space : The Final Frontier) from schools around Manchester, UK. April 1, 2009.
- COMPASS'09, Correctness, Modeling, and Performance of Aerospace Systems. York, UK, March 28 2009. Satellite workshop of ETAPS'09.
- 9th International Workshop on Runtime Verification. June 26-28, 2009, Grenoble, France. Satellite workshop of CAV'09.
- Software Reliability for Space Missions. July 20, 2009, Pasadena, California, USA. Satellite workshop of SMC-IT'09.
- FMA'09, Formal Methods for Aerospace. Eindhoven, the Netherlands, November, 2009. Satellite workshop of FM'09.
- EMSOFT'11, International Conference on Embedded Software. Taipei, Taiwan, October 9-14, 2011.
- SEFM'11, Software Engineering and Formal Methods. November 7-18, 2011, Montevideo, Uruguay. Invited tutorial together with Martin Leucker.

- TORRENTS'11, 2nd TORRENTS Workshop. Toulouse, France, December 12, 2011.
- HOWARD-60 Higher-Order Workshop on Automated Runtime Verification and Debugging. December 20, 2011, Manchester, UK.
- NFM'12, 4th NASA Formal Methods Symposium. Panel member/speaker on formal methods at NASA. April 3-5, 2012, Norfolk, Virginia, USA.
- SSS'12, Summer Software Symposium. July 20, 2012, Minneapolis, MN, USA.
- The 33rd Marktoberdorf Summer School, Aug 2012 (Engineering Dependable Software Systems). Lecture topic: Verifying Execution Traces.
- LCCC workshop: System Design meets Equation-based Languages. Invited participant (speaker). September 19-21, 2012, Lund, Sweden.
- ISoLA'12, Leveraging Applications of Formal Methods, Verification and Validation. Special track on Runtime Verification: the application perspective. October 15-18, 2012, Heraklion, Crete.
- ICTSS'12, The 24th IFIP Int. Conference on Testing Software and Systems. November 19-21, 2012, Aalborg, Denmark.
- WODA'13, The 11th International Workshop on Dynamic Analysis. March 16, 2013, Houston, Texas, USA.
- The CERIST Autumn School on Cyber-Physical Systems, Algiers, Algeria, September 30-October 3, 2013.
- Observer at IFIP Working Group 1.9/2.15 (Verified Software). December 9-12, 2013. University of Central Florida, Orlando, Florida, USA.
- Observer at IFIP Working Group 1.9/2.15 (Verified Software). July 14-16, 2014. Vienna, Austria.
- ISoLA'14, Special track on Statistical Model Checking, Past Present and Future. October 8-11, 2014, Corfu, Greece.
- TASE'14, The 8th International Symposium on Theoretical Aspects of Software Engineering. September 1-3, 2014, Changsha, China.
- Participant on "Tool panel" at: ISoLA'14, 6th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. October 8-11, 2014, Corfu, Greece. Panel directed by Dirk Beyer.
- FTSCS'14, Third International Workshop on Formal Techniques for Safety-Critical Systems. November 6-7, 2014, Luxembourg.
- EITEC'15, 2nd International IFIP Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems. April 13, 2015, Seattle, Washington, USA.
- SyDe Summer School 2015 - Post Graduate Program in System Design (Invited lecture). 9-11 September, 2015, Bremen, Germany.

- Observer at IFIP Working Group 2.3 (Programming Methodology). January 11-15, 2016. Pasadena, California, USA.
- CIT 2017, The 5th International CTI Conference. June 6-8, 2016. Detroit, MI, USA.
- Invited lecture for: RV 2016, The 16th International Conference on Runtime Verification. September 23-30, 2016. Madrid, Spain.
- Invited talk at: USC (University of Southern California) in the CS Colloquium Lecture series. November 1, 2016. Los Angeles, California, USA. Invited by Chao Wang.
- NFM'16, 8th NASA Formal Methods Symposium. Member on panel on formal methods at NASA. June 7-9, 2016, Minneapolis, MN, USA.
- Invited speaker at: Workshop on Software Correctness and Reliability. Oct 13-14, 2017, ETH Zurich, Switzerland.

6 Organization of Workshops and Conferences

A steering committee member is responsible for the continued execution of a meeting (workshop, conference, etc.). An organizer is responsible for a particular instance of a meeting.

6.1 Steering Committee Positions

6.1.1 Current

- Steering committee member for the RV (Runtime Verification) conference.
- Steering committee member for the NFM (NASA Formal Methods) conference.
- Steering committee member for the FME (Formal Methods Europe) conference.
- On the advisory board for ICACDS (International Conference on Advances in Computing and Data Sciences).

6.1.2 Past

- Chair of the steering committee for the RV (Runtime Verification) conference during 2010-2017. I co-started this as a workshop with Grigore Rosu in 2001. It became a conference in 2010.
- Chair of the steering committee for the NFM (NASA Formal methods) conference during 2011-2017.
- Steering committee member for the ETAPS conference series during 2014-2015 (due to having been PC chair for TACAS in 2014).
- Steering committee member for the SPIN symposium during 2009 (due to having been PC chair for SPIN in 2008).

6.2 Organization of Meetings

- SPIN workshops:
  - Co-organized 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).
  - PC chair of the 15th International SPIN Workshop on Model Checking of Software in Los Angeles, California (stand-alone event), August 10-12, 2008. See `http://compilers.cs.ucla.edu/spin08`.
  - PC Chair of the 24th International SPIN Symposium on Model Checking of Software. Santa Barbara, California, USA, April, 2017.
- VVPS workshops:
  - Started and co-organized the 1st International Workshop on Verification and Validation of Model-Based Planning and Scheduling Systems, Monterey, California, USA, June 2005: `http://planning.cis.strath.ac.uk/vvpsws`.
  - Co-organized the 2nd International Workshop on Verification and Validation of Planning and Scheduling Systems, Thessaloniki, Greece, September 2009: `http://www-vvps09.imag.fr`.
  - Co-organized the 3rd International Workshop on Verification and Validation for Planning and Scheduling Systems, Freiburg, Germany, June 11, 2011: `http://icaps11.icaps-conference.org/workshops/vvps.html`.
- RV workshops and conferences:

  I started in 2001, together with Grigore Rosu, the RV (Runtime Verification) workshop series. It became a conference in 2010. See `http://www.runtime-verification.org`.

  I co-organized myself the following RV events:
  - RV'01 - held as a CAV'01 satellite event in Paris, France, July 2001.
  - RV'02 - held as a CAV'02 satellite event in Copenhagen, Denmark, July 2002.
  - RV'04 - held as an ETAPS'04 satellite event in Barcelona, Spain, April 2004.
  - FATES/RV'06 - held as a FLoC'06 event in Seattle, USA, August 2006
  - RV'10 - The first International Conference on Runtime Verification, Malta, November, 2010. General chair.
  - RV'11 - The Second International Conference on Runtime Verification, San Francisco, September, 2011. I provided financial guarantee to conference hotel.
  - RV'14 - The Fifth International Conference on Runtime Verification, Waterloo, Canada, September, 2014. Publicity committee member.
  - RV'16 - The 16th International Conference on Runtime Verification. Madrid, Spain, September 23-30, 2016. Tool chair.
  - RV'17 - The 17th International Conference on Runtime Verification. Seattle, Washington, September 12-15, 2017. General chair.

  - RV-CuBES'17 - An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardization for Runtime Verification Tools. Seattle, Washington State, USA, September 15, 2017. (my role: PC chair).

- Dagstuhl seminars:

  - Co-organized a Dagstuhl meeting on runtime verification, January 3-6, 2007. See `http://www.dagstuhl.de/07011`.
  - Co-organized a Dagstuhl meeting on Runtime Verification, Diagnosis, Planning and Control for Autonomous Systems, November 7-12, 2010. See `http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=10451`.
  - Co-organizing a Dagstuhl meeting on Next Generation Static Software Analysis Tools, August 25-29, 2014.

- NFM symposiums:

  - NFM 2011 symposium: co-PC chair for the 3rd NASA Formal Methods Symposium, Pasadena, USA, April 18-20, 2011: `http://lars-lab.jpl.nasa.gov/nfm2011`.
  - NFM 2015 symposium: co-PC chair for the 7th NASA Formal Methods Symposium, Pasadena, USA, April 27-29, 2015: `http://www.nasaformalmethods.org`.

- TACAS 2014 conference: co-PC chair for TACAS 2014, the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Grenoble, France, April 6-13, 2014. `http://www.etaps.org/index.php/2014/tacas`.

- CPS SCHOOL 2014: Co-organizing Summer School on Cyber-Physical Systems - 2014 Edition. Organized by PERSYVAL-Lab and NASA-JPL. Grenoble, France, July 7-11, 2014.

- ISoLA 2016: 6th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Corfu, Greece, October 6-14, 2016.

  - Track titled: Towards a Unified view of Modeling and Programming. (Co-organizer together with Manfred Broy, Rahul Kumar, and Bernhard Steffen).
  - Track titled: Static and Runtime Verification: Competitors or Friends? (Co-organizer together with Dilian Gurov, Marieke Huisman, and Rosemary Monahan).

- FM 2018 Industry Day (I-Day), at 22nd International Symposium on Formal Methods (FM 2018), Oxford, UK, July 15-17, 2018. (Co-chair together with Ralf Pinger).

7 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).

1. SAC SVT 2018 Software Verification and Testing at SAC 2018, Pau, France, April 9 - 13, 2018.

2. TACAS 2018 The 24st International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Thessaloniki, Greece, April 14-21, 2018.

3. MODELSWARD 2018 6th International Conference on Model-Driven Engineering and Software Development Funchal, Madeira, Portugal, January 22-24, 2018.

4. SEFM 2017 15th International Conference on Software Engineering and Formal Methods Trento, Italy, September 4-8, 2017.

5. FMi 2017 The 5th IEEE International Workshop on Formal Methods Integration San Diego, USA, August 4-6, 2017.

6. VORTEX 2017 2nd Workshop on Runtime Verification for Object-Oriented Languages and Systems Barcelona, Spain, June 20, 2017.

7. ECOOP 2017 European Conference on Object-Oriented Programming Barcelona, Spain, June 18-23, 2017.

8. NFM 2017 9th NASA Formal Methods Symposium. NASA Ames Research Center, Moffett Field, CA, USA, May 16-18, 2017.

9. TACAS 2017, The 23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Uppsala, Sweden, April 23-29, 2017.

10. SAC-SVT 2017 Software Verification and Testing Marrakech, Morocco, April 3 - 7, 2017.

11. MODELSWARD 2017 5th International Conference on Model-Driven Engineering and Software Development. Porto, Portugal, February 19-21, 2017.

12. FTSCS 2016 Fifth International Workshop on Formal Techniques for Safety-Critical Systems. Tokyo, Japan, November 14, 2016.

13. ISSRE 2016 27th IEEE International Symposium on Software Reliability Engineering. Ottawa, Canada, October 23-27, 2016.

14. ICTSS 2016 IFIP 28th International Conference on Testing Software and Systems. Graz, Austria, October 17-19, 2016.

15. ATVA 2016, 14th International Symposium on Automated Technology for Verification and Analysis. Shanghai, China, October 12-15, 2016.

16. ISoLA 2016 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation Corfu, Crete, October 5-14, 2016.

17. RV 2016, 7th International Conference on Runtime Verification. Madrid, Spain, September 23-30, 2016.

18. SEW-36 - 2016, 37th Annual IEEE Software Engineering Workshop. Gdansk, Poland, September 11 - 14, 2016.

19. FORMABS 2016 Formal Methods for Analysis of Business Systems Singapore, September 3, 2016.

20. FMi 2016 4th IEEE International Workshop on Formal Methods Integration. Pittsburgh, Pennsylvania, USA, July 28-30, 2016.

21. VORTEX 2016 1st Workshop on Runtime Verification for Object-Oriented Languages and Systems Rome, Italy, July 18, 2016.

22. SEFM 2016, 14th International Conference on Software Engineering and Formal Methods. Vienna, Austria, July 4-8, 2016.

23. SafePlan 2016 Planning, Scheduling and Dependability in Safe Human-Robot Interactions London, UK, June 13, 2016.

24. NFM 2016, 8th NASA Formal Methods Symposium. Minneapolis, MN, USA, June 7-9, 2016.

25. PrePost 2016, 1st Workshop onPre- and Post-Deployment Verification Techniques. Reykjavik, Iceland, June 1-3, 2016.

26. ABZ 2016, 5th International ABZ Conference on ASM, Alloy, B, TLA, VDM, Z. Linz, Austria, May 23-27, 2016.

27. EITEC 2016, 4rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems. Vienna, Austria, April 11, 2016.

28. SAC-SVT 2016, Software Verification and Testing. Pisa, Italy, April 3-8, 2016.

29. SAC-MUSEPAT 2016, Multicore Software Engineering, Performance, Applications, and Tools. Pisa, Italy, April 3-8, 2016.

30. MODELSWARD 2016, 4th International Conference on Model-Driven Engineering and Software Development. Rome, Italy, February 19-21, 2016.

31. FTSCS 2015, Fourth International Workshop on Formal Techniques for Safety-Critical Systems. Paris, France, November 6-7, 2015.

32. RV 2015, 6th International Conference on Runtime Verification. Vienna, Austria, September 22-25, 2015.

33. SYNASC 2015, 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Timisoara, Romania, September 21-24, 2015.

34. SEFM 2015, 13th International Conference on Software Engineering and Formal Methods. York, UK, September 7-11, 2015.

35. SPIN 2015, 22nd International SPIN Symposium on Model Checking of Software. Stellenbosch, South Africa, August 24-26, 2015.

36. FMi 2015, 3rd IEEE International Workshop on Formal Methods Integration. San Francisco, California, USA, August 13-15, 2015.

37. SummersSim 2015, Summer Simulation Multi-Conference. Chicago, IL, USA, July 26-29, 2015.

38. FM 2015, The 20th International Symposium on Formal Methods. Oslo, Norway, June 22-26, 2015.

39. MOCHAP 2015, Workshop on Model Checking and Automated Planning. Jerusalem, Israel, June 7, 2015.

40. ICSE Student Competition 2015, 37th ACM/IEEE International Conference on Software Engineering - ACM Student Research Competition. Firenze, Italy, May 16-24, 2015.

41. EITEC 2015, 2nd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems. Seattle, WA, USA, April 13-17, 2015.

42. TACAS 2015, The 21th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. London, UK, April 11-19, 2015.

43. PPMG 2015, Second Congress on Multicore and GPU Programming. Caceres, Spain, March 5-6, 2015.

44. MODELSWARD 2015, 3rd International Conference on Model-Driven Engineering and Software Development. Angers, France, February 9-11, 2015.

45. MUSEPAT 2014, 3rd International Conference on Multicore Software Engineering, Performance, and Tools. Hong Kong, November 16-17, 2014.

46. FTSCS 2014, Third International Workshop on Formal Techniques for Safety-Critical Systems. Luxembourg, November 6-7, 2014.

47. ISSRE 2014, 25th IEEE International Symposium on Software Reliability Engineering. Naples, Italy, November 3-6, 2014.

48. ACM Student Research Competition, MODELS 2014, ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems. Valencia, Spain, September 28-October 3, 2014.

49. ICTSS 2014, IFIP 26th International Conference on Testing Software and Systems. Madrid, Spain, September 23-25, 2014.

50. SEFM 2014, 12th edition of the International Conference on Software Engineering and Formal Methods. Grenoble, France, September 1-5, 2014.

51. RV 2014, 5th International Conference on Runtime Verification. On PC as well as on tool demonstration committee. Waterloo, Canada, September, 2014.

52. SEW 2014, 36th Annual IEEE Software Engineering Workshop. NASA Ames Research Center, California, USA, August 19-20 2014.

53. FMi 2014, 2nd IEEE International Workshop on Formal Methods Integration. San Francisco, California, USA, August 13-15, 2014.

54. VSTTE 2014, 6th Working Conference on Verified Software: Theories, Tools, and Experiments. Vienna, Austria, July 17-18, 2014.

55. SummerSim'14, Summer Simulation Multi-Conference. Monterey, CA, USA, July 6 - 10, 2014.

56. MOCHAP 2014, Workshop on Model Checking and Automated Planning. Portsmouth, USA, June 21-26, 2014.

57. FORTE 2014, 34th IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems. Berlin, Germany, June 3-6, 2014.

58. ABZ 2014, 4th International ABZ Conference on ASM, Alloy, B, TLA, VDM, Z. Toulouse, France, June 2-6, 2014.

59. NFM 2014, 6th NASA Formal Methods Symposium. NASA Johnson Space Center, Houston, Texas, USA, April 29 - May 1, 2014.

60. Modularity 2014, 13th International Conference on Modularity. Lugano, Switzerland, April 22-26, 2014.

61. HSCC 2014, The 17th International Conference on Hybrid Systems: Computation and Control. Berlin, Germany, April 15th-17th, 2014.

62. TACAS 2014, The 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Grenoble, France, April 6-13, 2014. In addition to being PC chair.

63. HAS 2014, 4th Workshop on Hybrid Autonomous Systems. Grenoble, France, April 5-13, 2014. In conjunction with the The European Joint Conferences on Theory and Practice of Software (ETAPS 2014).

64. MODELSWARD 2014, 2nd International Conference on Model-Driven Engineering and Software Development. Lisabon, Portugal, January 7-9, 2014.

65. ISSRE 2013, 24th IEEE International Symposium on Software Reliability Engineering. Pasadena, CA, USA, November 4-7, 2013.

66. ICTSS 2013, IFIP 25th International Conference on Testing Software and Systems. Istanbul, Turkey, November 13-15, 2013.

67. FTSCS 2013, Second International Workshop on Formal Techniques for Safety-Critical Systems. Queenstown, New Zealand, October 28, 2013.

68. SEFM 2013, 11th IEEE International Conference on Software Engineering and Formal Methods. Madrid, Spain, September 25-27, 2013.

69. FMI 2013, IEEE International Workshop on Formal Methods Integration. San Francisco, California, USA, August 14-16, 2013

70. NFM 2013, Fifth NASA Formal Methods Symposium. Moffett Field, California, USA, May 14 - 16, 2013.

71. RV 2013, 4rd International Conference on Runtime Verification. Rennes, France, September 24-27, 2013.

72. MUSEPAT 2013, International Conference on Multicore Software Engineering, Performance, and Tools. Saint Petersburg, Russia, August 21-23, 2013.

73. SPIN 2013, International SPIN Symposium on Model Checking of Software. Stony Brook, NY, USA, July 8-9, 2013.

74. HSCC 2013, The 16th International Conference on Hybrid Systems: Computation and Control. Philadelphia, USA, April 8-11, 2013.

75. AOSD 2013, 12th International Conference on Aspect-Oriented Software Development. Fukuoka, Japan, March 24-29, 2013.

76. TACAS 2013, The 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Rome, Italy, March 16-24, 2013.

77. WODA 2013, 11th International Workshop on Dynamic Analysis. Houston, Texas, March 16, 2013.

78. VMCAI 2013, The 14th Intl. Conference on Verification, Model Checking, and Abstract Interpretation. Rome, Italy, January 20-22, 2013, with POPL.

79. ISSRE 2012, 23rd IEEE International Symposium on Software Reliability Engineering. Dallas, TX, USA, November 27-30, 2012.

80. ICTSS 2012, IFIP 23rd International Conference on Testing Software and Systems. Aalborg, Denmark, November 19-21, 2012.

81. ICFEM 2012, The 14th Intl. Conference on Formal Engineering Methods. Kyoto, Japan, November 12-16, 2012.

82. FTSCS 2012, First International Workshop on Formal Techniques for Safety-Critical Systems. Kyoto, Japan, November 12, 2012.

83. JPF 2012, The Java Pathfinder Workshop 2012, affiliated with FSE 2012. Cary North Carolina, November 11-16, 2012.

84. HVC 2012, Haifa Verification Conference 2012. Haifa, Israel, November 6-8, 2012.

85. SEW 2012, 35th Annual IEEE Software Engineering Workshop. Heraclion, Crete, Greece, 12-13 October 2012.

86. SEFM 2012, 10th IEEE International Conference on Software Engineering and Formal Methods. Thessaloniki, Greece, October 1-5, 2012.

87. RV 2012, 3rd International Conference on Runtime Verification. Istanbul, Turkey, September 25-28, 2012.

88. RTT 2012, 3rd International Workshop on Requirements@run.time. Chicago, USA, September 24, 2012.

89. FMSPLE 2012, Third Workshop on Formal Methods and Analysis in Software Product Line Engineering. Salvador, Brazil, September 2, 2012.

90. FM 2012, The 18th International Symposium on Formal Methods. Paris, France, August 27-31, 2012.

91. PADTAD 2012, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging. Minneapolis, MN, USA, July 16, 2012.

92. MSVVEIS 2012, The 10th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems. Wroclaw, Poland, June 28-July 1, 2012.

93. ICAPS 2012, The 22nd International Conference on Automated Planning and Scheduling. Special track on continuous planning. Atibaia, Sao Paulo, Brazil, June 24-28, 2012.

94. ABZ 2012, International Conference of Alloy, ASM, B, VDM, and Z Users. Pisa, Italy, June 18-22, 2012.

95. FMOODS & FORTE 2012, IFIP International Conference on Formal Techniques for Distributed Systems joint international conference, 14th Formal Methods for Open Object-Based Distributed Systems, 32nd Formal Techniques for Networked and Distributed Systems. Stockholm, Sweden, 13-16 June 2012.

96. INTELLI 2012, The First International Conference on Intelligent Systems and Applications. Chamonix / Mont Blanc, France, April 29 - May 4, 2012.

97. HSCC 2012, The 15th International Conference on Hybrid Systems: Computation and Control. Beijing, April 17-19, 2012.

98. NFM 2012, Fourth NASA Formal Methods Symposium. Norfolk, Virginia, USA, April 3 - 5, 2012.

99. AOSD 2012, 11th International Conference on Aspect-Oriented Software Development. Potsdam, Germany, March 25-30th, 2012.

100. HVC 2011 Award committee, Haifa Verification Conference 2011. Haifa, Israel, December 5-8, 2011.

101. HVC 2011, Haifa Verification Conference 2011. Haifa, Israel, December 5-8, 2011.

102. JPF 2011, The Java Pathfinder Workshop 2011, affiliated with ASE 2011. Oread, Lawrence, Kansas, November 12, 2011.

103. RV 2011, 2nd International Conference on Runtime Verification. Berkeley, California, September 27-30, 2011.

104. FMSPLE 2011, 2nd International Workshop on Formal Methods and Analysis in Software Product Line Engineering. Munich, Germany, August 26, 2011.

105. K 2011, 2nd International K Workshop. Cheile Gradistei, Romania, 8-12 August, 2011.

106. SHM 2011, 2nd International Workshop on Software Health Management. Palo Alto, California, USA, August 2, 2011.

107. PADTAD 2011, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging. Toronto, Canada, July 17, 2011. Co-located with ISSTA 2011.

108. SPIN 2011, 18th International SPIN Workshop on Model Checking Software. Snowbird, Utah, USA, July 13-14, 2011. Co-located with CAV 2011.

109. FM 2011, The 17th International Symposium on Formal Methods. Limerick, Ireland, June 20-24, 2011.

110. SEW 2011, 34th Annual IEEE Software Engineering Workshop. Limerick, Ireland, June 20-21, 2011. Co-located with FM 2011.

111. MSVVEIS 2011, The 9th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems. Bejing, China, June 8-9, 2011.

112. Scala Days 2011, The Second Scala Workshop. Stanford University, USA, June 2, 2011.

113. HSCC 2011, The 14th International Conference on Hybrid Systems: Computation and Control. Chicago, April 12-14, 2011.

114. FASE 2011, The 14th International Conference on Fundamental Approaches to Software Engineering. Saarbrucken, Germany, 26 March - 4 April, 2011. Co-located with the The European Joint Conferences on Theory and Practice of Software (ETAPS 2011).

115. HAS 2011, Workshop on Hybrid Autonomous Systems. Saarbruecken, Germany, April 2-3, 2011. Co-located with the The European Joint Conferences on Theory and Practice of Software (ETAPS 2011).

116. ICTSS 2010, IFIP 22nd International Conference on Testing Software and Systems. Natal, Brazil, November 8-10, 2010.

117. HVC 2010, The IBM Verification Conference 2010. Haifa, Israel, October 5-7, 2010.

118. SSV 2010, Systems Software Verification. Vancouver, BC, Canada, October 4-6, 2010. Co-located with the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010).

119. SEFM 2010, 8th IEEE International Conference on Software Engineering and Formal Methods. Pisa, Italy, September 13-17, 2010.

120. MoChArt 2010, Sixth Workshop on Model Checking and Artificial Intelligence. Atlanta, Georgia, USA, July 11-12, 2010. Co-located with the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010).

121. PADTAD 2010, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging. Trento, Italy, July 12-13, 2010. Co-located with ISSTA 2010.

122. DRHE 2010, The Seventh IARP Workshop on Technical Challenges for Dependable Robots in Human Environments. Toulouse, France, June 16-17, 2010.

123. MSVVEIS 2010, The 8th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems. Funchal, Madeira, Portugal, June 8-12, 2010.

124. PLDI 2010, Conference on Programming Language Design and Implementation. External Review Committee (ERC) member. Toronto, Canada, June 5-10, 2010.

125. NFM 2010, Second NASA Formal Methods Symposium. Washington D.C, USA, April 13 - 15, 2010.

126. SATE 2009, Static Analysis Tool Exposition Workshop. Arlington, Virginia, USA, November 6, 2009.

127. VVPS 2009, 2nd International Workshop on Verification and Validation of Planning and Scheduling Systems, Thessaloniki, Greece, September 2009.

128. PADTAD 2009, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging. Chicago, Illinois, USA, July 19-20, 2009. Co-located with ISSTA 2009.

129. CAV 2009, The 21'st International Conference on Computer Aided verification. Grenoble, France, June 26-July 2, 2009.

130. RV 2009, The 9th International Workshop on Runtime Verification. Grenoble, France, June 26-June 28, 2009.

131. SPIN 2009, 16th International SPIN Workshop on Model Checking of Software. Grenoble, France, June 26-28, 2009, Co-located with CAV 2009.

132. MSVVEIS 2009, The Seventh International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Milano, Italy, 2009.

133. NFM 2009, The 1'st NASA Formal Methods Symposium. Mountain View (Silicon Valley), California, USA, April 6-8, 2009.

134. VMCAI 2009, The 10th Intl. Conference on Verification, Model Checking, and Abstract Interpretation, Savannah, GA, January 18-20, 2009.

135. SEFM 2008, 6th IEEE International Conference on Software Engineering and Formal Methods. November 10-14, 2008, Cape Town, South Africa.

136. HVC 2008, The IBM Verification Conference 2008, Haifa, Israel, October 28-30, 2008.

137. ICSEA 2008, The Third International Conference on Software Engineering Advances Sliema, Malta, October 26-31, 2008.

138. SLE 2008, The First International Conference on Software Language Engineering, Toulouse, France, September 29-30, 2008.

139. MSVVEIS 2008, The Sixth International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Barcelona, Spain, 2008.

140. PADTAD 2008, Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging, Seattle, Washington, USA, July 20-21, 2008.

141. SAW 2008, ACM SIGPLAN Static Analysis Workshop (Co-located with PLDI 2008). Tucson, Arizona, June 12, 2008.

142. AOSD 2008, Seventh International Conference on Aspect-Oriented Software Development. Brussels, Belgium, March 31 - April 4, 2008.

143. CONFENIS 2007, the IFIP International Conference on Research and Practical Issues of Enterprise Information Systems. October 14-16, 2007, Beijing, China.

144. The IBM Verification Conference 2007. October 23 - 25, 2007. Organized by IBM Research Lab in Haifa, Israel.

145. SEFM 2007, 5th IEEE International Conference on Software Engineering and Formal Methods. September 10-14, 2007, London, UK.

146. ICSEA 2007, the Second International Conference on Software Engineering Advances August 25-31, 2007, Cap Esterel, French Riviera, France.

147. 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.

148. MSVVEIS 2007, the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, 12-13 June, 2007, Funchal, Madeira - Portugal.

149. The IBM Verification Conference 2006. October 23 - 26, 2006. Organized by IBM Research Lab in Haifa, Israel.

150. ICSEA 2006, International Conference on Software Engineering Advances.

151. SEFM 2006, 4th IEEE International Conference on Software Engineering and Formal Methods.

152. FMICS 2006, 11th International Workshop on Formal Methods for Industrial Critical Systems.

153. PADTAD 2006, Parallel and Distributed Systems: Testing and Debugging.

154. MSVVEIS 2006, The Fourth International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems.

155. TACAS 2006, The 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems.

156. RV 2005, The 5th International Workshop on Runtime Verification.

157. VVEIS 2005, The 3rd International Workshop on Verification and Validation of Enterprise Information Systems.

158. SEFM 2005, 3rd IEEE International Conference on Software Engineering and Formal Methods.

159. FATES 2005, Formal Approaches to Testing of Software.

160. ICI 2004, International Conference on Informatics.

161. CAV 2004, The 16th Conference in Computer Aided Verification.

162. DAW 2004, Dynamic Aspects Workshop.

163. SFEDL 2004, Semantic Foundations of Engineering Design Languages.

164. REOS 2003, Workshop on Requirements Engineering and Open Systems.

165. ASARTI 2003, Workshop: Advancing the State-of-the-Art in Run-Time Inspection.

166. FMPPTA 2003, The 8th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.

167. DSN'03, Workshop on Model Checking for Dependable Software-Intensive Systems.

168. ESEC/FSE'03, 4th joint meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering.

169. CAV 2002, The 14th Conference in Computer Aided Verification.

170. RT-TOOLS, 2002 Workshop on Real-Time Tools.

171. FMPPTA 2002, The 7th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.

172. SPIN 2002, The 9th International SPIN Workshop on Model Checking of Software.

173. SPIN 2001, The 8th International SPIN Workshop on Model Checking of Software.

174. FMPPTA 2001, 6th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.

175. MVI 2001, Model-based Validation of Intelligence.

176. Lfm 2000, Fifth NASA Langley Formal Methods Workshop.

177. PATV 2000, The First International Workshop on Automated Program Analysis, Testing and Verification.

178. JFLA 2000, Les onzimes Journes Francophones des Langages Applicatifs.

8 Teaching

Most of my carrier has been spent in research labs, so my teaching experience is limited. I have given the following classes as appointed Lecturer in Computer Science at California Institute of Technology (`http://www.caltech.edu`):

- A class on *program monitoring* techniques during the period April 28 - May 29, 2008, as the second half of a class on *Reliable Software: Testing and Monitoring*.
  See course website: `http://www.runtime-verification.org/course`.
- A modified version of this course was given during May 2009.
  See course website: `http://www.runtime-verification.org/course09`.

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.

[3] Klaus Havelund, Giles Reger, Daniel Thoma, and Eugen Zălinescu. Monitoring Specifications with Data. Book chapter, Springer. 2017. Book editors: Ezio Bartocci and Ylies Falcone. To appear.

[4] Ylies Falcone, Klaus Havelund, and Giles Reger. *A Tutorial on Runtime Verification*. Edited by Manfred Broy, Doron Peled, and Georg Kalus. IOS Press online book. NATO Science for Peace and Security Series - D: Information and Communication Security. Volume 34. 2013. Lecture given by Klaus Havelund at Summer School Marktoberdorf 2012 - Engineering Dependable Software Systems. July 31 to August 12, 2012.

JPL Java Coding Standard

JPL Institutional Coding Standard for the Java Programming Language. Version 1.0, March 31, 2014. Created by semmle.com, Klaus Havelund (JPL), and Al Niessner (JPL).

Journal Papers

[5] Erika Ábrahám and Klaus Havelund. Some Recent Advances in Automated Analysis. International Journal on Software Tools for Technology Transfer (STTT), 18(2), pages 121-128, 2016. Introduction article to special issue containing selected submissions from TACAS 2014.

[6] Rahul Agarwal, Saddek Bensalem, Eitan Farchi, Klaus Havelund, Yarden Nir-Buchbinder, Scott D. Stoller, Shmuel Ur, and Liqiang Wang. Detection of Deadlock Potentials in Multithreaded Programs. IBM Journal of Research & Development, Volume: 54 Issue:5. September 2010. Digital Object Identifier: 10.1147/JRD.2010.2060276.

[7] 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 [34].

[8] Cyrille Artho, Klaus Havelund, and Armin Biere. High-Level Data Races. *Software Testing, Verification and Reliability*, 13(4), 2004. Extended version of [37]. To appear.

[9] Howard Barringer, Alex Groce, Klaus Havelund and Margaret Smith. Formal Analysis of Log Files. Journal of Aerospace Computing, Information, and Communication.

Volume 7 - Issue 11, 2010. Pages 365-390. Invited talk given at the *SMC-IT'09 workshop: Software Reliability for Space Missions*. July 20, 2009, Pasadena, California, USA.

[10] Howard Barringer, David Rydeheard and Klaus Havelund. Rule Systems for Run-Time Monitoring: from Eagle to RuleR. Journal version of [48]. Journal of Logic and Computation (JLC), Vol. 20 No. 3. Oxford University Press. Published online 21 November 2008 doi:10.1093/logcom/exn076.

[11] Ezio Bartocci, Ylies Falcone, Borzoo Bonakdarpour, Christian Colombo, Normann Decker, Klaus Havelund, Yogi Joshi, Felix Klaedtke, Reed Milewicz, Giles Reger, Grigore Rosu, Julien Signoles, Daniel Thoma, Eugen Zalinescu, and Yi Zhang. First International Competition on Runtime Verification – Rules, Benchmarks, Tools, and Final Results of CRV 2014. *International Journal on Software Tools for Technology Transfer (STTT)*, volume TBD, issue TBD, pages TBD, 2017. Springer-Verlag.

[12] Matthew Bennett, Richard Borgen, Klaus Havelund, Michel Ingham and David Wagner. Prototyping a Domain-Specific Language for Monitor and Control Systems. Journal of Aerospace Computing, Information, and Communication. Volume 7 - Issue 11, 2010. Pages 338-364. Journal version of [49].

[13] Saddek Bensalem, Klaus Havelund, and Andrea Orlandini. Verification & Validation Meets Planning & Scheduling. *International Journal on Software Tools for Technology Transfer (STTT), 16(1) 2014*. Introduction to special issue containing selected submissions for the 3rd ICAPS workshop on Verification & Validation of Planning & Scheduling Systems (VVPS 2011). (Also presented at MOCHAP 2014 - Workshop on Model Checking and Automated Planning. Portsmouth, USA, June 21-26, 2014).

[14] 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.

[15] Eric Bodden and Klaus Havelund. Aspect-oriented Race Detection in Java. *IEEE Transactions on Software Engineering (TSE), Volume 36 no. 4, July/August 2010.* Pages 509-527. Journal version of [53].

[16] 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 [54].

[17] 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.

[18] 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 [117].

[19] Alex Groce, Klaus Havelund, Gerard Holzmann, Rajeev Joshi, and Ru-Gang Xu. Establishing Flight Software Reliability: Testing, Model Checking, Constraint-Solving,

and Monitoring. *Annals of Mathematics and Artificial Intelligence*. Submitted first time 2009, accepted for publication. To appear.

[20] Klaus Havelund. Rule-based Runtime Verification Revisited. *International Journal on Software Tools for Technology Transfer (STTT)*, volume 17, issue 2, page 143-170. doi:10.1007/s10009-014-0309-2. Special STTT issue containing selected submissions from the 5th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2012); Track: Runtime Verification, the Application Perspective. Published online April 2013. Paper version March 2015.

[21] Klaus Havelund and Rahul Kumar. Verified Change. To appear in FoMaC: Transactions on Foundations for Mastering Change (edited by Bernhard Steffen). First issue containing submissions from the editorial board. Submitted in final version August 2016.

[22] 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.

[23] 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, organzied by K. Havelund, J. Penix and W. Visser. The article is the introductory article to the special issue. Havelund and Visser are guest editors.

[24] Klaus Havelund and Kim Guldstrand Larsen. The Fork Calculus. *Nordic Journal of Computing*, 1, 1994. Journal version of [103].

[25] 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 [105].

[26] 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 [109].

[27] 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 [97].

[28] Sean Kauffman, Klaus Havelund, Rajeev Joshi, and Sebastian Fischmeister. Inferring Event Stream Abstractions. Submitted to: Formal Methods in System Design (FMSD). Springer. 2017. Journal version of RV 2016 paper with the title: nfer - A Notation and System for Inferring Event Stream Abstractions. Under review.

[29] 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 [110].

[30] Justin Seyster, Ketan Dixit, Xiaowan Huang, Radu Grosu, Klaus Havelund, Scott A. Smolka, Scott D. Stoller, and Erez Zadok. InterAspect: Aspect-Oriented Instrumentation with GCC. Formal Methods in System Design, 41(3), 295-320, December 2012. Extended version of [121].

[31] Oleg Sokolsky, Klaus Havelund and Insup Lee. Introduction to the Special Section on Runtime Verification. *International Journal on Software Tools for Technology Transfer, STTT*, 14(3):243–247, June 2012. Introduction article to special issue containing selected submissions on runtime verification.

[32] 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 [125].

Conference and Workshop Papers

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] Cyrille Artho, Klaus Havelund, Rahul Kumar, and Yoriyuki Yamagata. Domain-Specific Languages with Scala. In Michael Butler, Sylvain Conchon, and Fatiha Zaidi editors, 17th International Conference on Formal Engineering Methods (ICFEM 2015). 3-6 November 2015 - Paris. LNCS, Volume 9407, pages 1-16, Springer-Verlag.

[40] Howard Barringer, Ylies Falcone, Klaus Havelund, Giles Reger, and David Rydeheard. Quantified Event Automata - Towards Expressive and Efficient Runtime Monitors. FM 2012, Paris, France, August 27-31, 2012. Volume 7436 of Lecture Notes in Computer Science.

[41] 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.

[42] 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.

[43] Howard Barringer, Alex Groce, Klaus Havelund, and Margareth Smith. An Entry Point for Formal Methods: Specification and Analysis of Event Logs - Extended Abstract. Invited talk given at *the 1'st Workshop on Formal Methods in Aerospace (FMA'09)*. Affiliated with FM'09. To appear in Electronic Proceedings of Theoretical Computer Science (EPTCS). Eindhoven, Holland, November 3, 2009. The work received JPL's internal Mariner Award in August 2009. Preliminary 2 page abstract for handout-workshop proceedings was published with the title "Runtime Verification of Log Files, a Trojan Horse for Formal Methods" (Howard Barringer, Alex Groce, Klaus Havelund, David Rydeheard, and Margareth Smith).

[44] Howard Barringer and Klaus Havelund. TraceContract: A Scala DSL for Trace Analysis. 17th International Symposium on Formal Methods (FM'11), Limerick, Ireland, June 20-24, 2011. Volume 6664 of Lecture Notes in Computer Science.

[45] H. Barringer and K. Havelund. Internal versus External DSLs for Trace Analysis. Extended abstract for presented tutorial. In *Proceedings of the 2nd International Conference on Runtime Verification (RV 2011)*, San Francisco, California, USA, October 27-30, 2011. Volume TBD of *LNCS*.

[46] Howard Barringer, Klaus Havelund, Elif Kurklu, and Robert Morris. Checking Flight Rules with TraceContract: Application of a Scala DSL for Trace Analysis. Scala Days 2011, Stanford University, USA, June 2, 2011.

[47] Howard Barringer, Klaus Havelund, David Rydeheard, and Alex Groce. Rule Systems for Runtime Verification: A Short Tutorial. *9th international workshop on Runtime Verification (RV'09), volume 5779 of LNCS*. July, 2009.

[48] 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. Volume 4839 of *LNCS*.

[49] M. Bennett, R. Borgen, K. Havelund, M. Ingham and D. Wagner. Development of a Prototype Domain-Specific Language for Monitor and Control Systems. IEEE Aerospace Conference, Big Sky, Montana, March 1-8, 2008.

[50] 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.

[51] Saddek Bensalem and Klaus Havelund. Dynamic Deadlock Analysis of Multi-Threaded Programs. In *Parallel and Distributed Systems: Testing and Debugging (PADTAD'05)*, November 2005. Volume 3875 of Lecture Notes in Computer Science. Haifa, Israel.

[52] Dines Bjørner and Klaus Havelund. 40 Years of Formal Methods - Some Obstacles and Some Possibilities?. FM 2014, Singapore, May 12-16, 2014. Volume 8442 of Lecture Notes in Computer Science.

[53] 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**.

[54] 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.

[55] Manfred Broy, Klaus Havelund, and Rahul Kumar. Towards a Unified View of Modeling and Programming. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2016: 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation – Part II. Track: Towards a Unified View of Modeling and Programming*, organized by Manfred Broy, Klaus Havelund, Rahul Kumar, and Bernhard Steffen. Corfu, Greece, October 10-14, 2016. Volume 9952 of *LNCS*, pages 238-260, Springer-Verlag.

[56] Manfred Broy, Klaus Havelund, Rahul Kumar, and Bernhard Steffen. Towards a Unified View of Modeling and Programming (Track Summary). In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2016: 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation – Part II. Track: Towards a Unified View of Modeling and Programming*, organized by Manfred Broy, Klaus Havelund, Rahul Kumar, and Bernhard Steffen. Corfu, Greece, October 10-14, 2016. Volume 9952 of *LNCS*, pages 3-10, Springer-Verlag.

[57] 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.

[58] Marcelo d'Amorim and Klaus Havelund. Event-Based Runtime Verification of Java Programs. In *Proceedings of the third international workshop on dynamic analysis (WODA'05)*, May 2005. St. Louis, Missouri, USA. SIGSOFT Softw. Eng. Notes 30(4), ACM, New York, NY, USA. Page 1-7. doi: http://doi.acm.org/10.1145/1082983.1083249.

[59] 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.

[60] 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, Scottland, 23-27 September 2002.

[61] 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`.

[62] Robert Filman and Klaus Havelund. The Effect of AOP on Softare 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.

[63] 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.

[64] 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.

[65] 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.

[66] Allen Goldberg and Klaus Havelund Conor McGann. Runtime Verification for Autonomous Space Craft Software. In *IEEE Aerospace Conference, Big Sky, Montana, USA, March 2005*, October 2004.

[67] Alex Groce, Klaus Havelund, and Margareth Smith. From Scripts to Specifications: the Evolution of a Flight Software Testing Effort. In *ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, - Volume 2, 2010, Cape Town, South Africa, May 2-8, 2010*. doi 10.1145/1810295.1810314. The work received JPL's internal Mariner Award in August 2009.

[68] Dilian Gurov, Klaus Havelund, Marieke Huisman, and Rosemary Monahan. Static and Runtime Verification, Competitors or Friends? (Track Summary). In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2016: 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation – Part I. Track: Static and Runtime Verification, Competitors or Friends?*, organized by Dilian Gurov, Klaus Havelund, Marieke Huisman, and Rosemary Monahan. Corfu, Greece, October 10-14, 2016. Volume 9952 of *LNCS*, pages 397-401, Springer-Verlag.

[69] 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.

[70] 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).

[71] 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.

[72] K. Havelund. Runtime Verification of C Programs. Abstract. *1st Dagstuhl Seminar on Runtime Verification, Schloss Dagstuhl, Germany, January 3–6, 2007.*

[73] 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.

[74] Klaus Havelund. Implementing Runtime Monitors. Extended abstract. 2nd TORRENTS Workshop (Time ORiented Reliable Embedded NeTworked Systems), Toulouse, France, December 12, 2011, Invited presentation.

[75] Klaus Havelund. Closing the Gap Between Specification and Programming: VDM^{++} and Scala. HOWARD-60: Higher Order Workshop on Automated Runtime verification and Debugging. Festschrift celebrating Howard Barringer's 60th Birthday, hosted by CICADA and CS, the University of Manchester. Manchester, UK, December 20, 2011. EasyChair Proceedings Volume 1 (first edition). Editors: Andrei Voronkov and Margarita Korovina. Invited presentation. Extended abstract appeared under the title: *Where Specification and Programming Meet.*

[76] Klaus Havelund. Howard Barringer - The Man who Invented the Past. HOWARD-60: Higher Order Workshop on Automated Runtime verification and Debugging. Festschrift celebrating Howard Barringer's 60th Birthday, hosted by CICADA and CS, the University of Manchester. Manchester, UK, December 20, 2011. EasyChair Proceedings Volume 1 (first edition). Editors: Andrei Voronkov and Margarita Korovina.

[77] Klaus Havelund. What does AI have to do with RV? (extended abstract). In *5th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Track: Runtime Verification - the Application Perspective. (ISoLA 2012)*, Heraclion, Crete, October 15-18, 2012. Volume 7610 of *LNCS*.

[78] Klaus Havelund. Requirements-Driven Log Analysis - Extended Abstract. In B. Nielsen and C. Weise, editors, *24th IFIP WG 6.1 Int. Conference on Testing Software and Systems (ICTSS'12)*. 19-21 November 2012 - Aalborg, Denmark. LNCS 7641, Springer-Verlag.

[79] Klaus Havelund. A Scala DSL for Rete-based Runtime Verification (short paper). In *Proceedings of the 4th International Conference on Runtime Verification (RV 2013)*, Rennes, France, September 24-27, 2013. Volume 8174 of *LNCS*.

[80] Klaus Havelund. Mastering Change @ Runtime. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2014: 6th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Response to a call for opinion statements from members of the editorial board of the upcoming journal: LNCS Transactions on Foundations for Mastering Change (FoMaC)*, Corfu, Greece, October 8-11, 2014. Volume 8802 of *LNCS*, pages 533-534, Springer-Verlag.

[81] Klaus Havelund. Monitoring with Data Automata. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2014: 6th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Track: Statistical Model Checking, Past Present and Future*, organized by Kim Larsen and Axel Legay. Corfu, Greece, October 8-11, 2014. Volume 8803 of *LNCS*, pages 254-273, Springer-Verlag.

[82] Klaus Havelund. Data Automata in Scala. In Martin Leucker and Ji Wang, editors, *The 8th International Symposium on Theoretical Aspects of Software Engineering (TASE 2014)*, Changsha, China, September 1-3, 2014. IEEE Computer Society Press, pages 1-9. Invited presentation.

[83] 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.

[84] Klaus Havelund, Alex Groce, Gerard Holzmann, Rajeev Joshi, and Margaret Smith. Automated Testing of Planning Models. In D. Peled and M. Wooldridge, editors, *MOCHART'08: The Fifth International Workshop on Model Checking and Artificial Intelligence*, Lecture Notes in Artificial Intelligencemo, volume 5348, Springer-Verlag. Patras, Greece, July 2008.

[85] Klaus Havelund and Gerard Holzmann. Software Certification: Coding, Code, and Coders. *EMSOFT 2011: International Conference on Embedded Software. October 9-14, 2011, Taipei, Taiwan*, Invited paper.

[86] Klaus Havelund, Michel Ingham, and David Wagner. A Case Study in DSL Development: An Experiment with Python and Scala. In *Scala Days 2010, Lausanne, Switzerland*, April 2010.

[87] 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.

[88] Klaus Havelund and Rajeev Joshi. Comprehension of Spacecraft Telemetry using Hierarchical Specifications of Behavior. In Stephan Merz and Jun Pang, editors, *16th International Conference on Formal Engineering Methods (ICFEM 2014)*, Luxembourg, November 3-7, 2014. Volume 8829 of *LNCS*, pages 187-202, Springer-Verlag.

[89] Klaus Havelund and Rajeev Joshi. Experience with Rule-Based Analysis of Spacecraft Logs. In Cyrille Artho and Peter Olveczky, editors, *Third International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2014)*, Luxembourg, November 6-7, 2014. Volume 476 of *Communications in Computer and Information Science (CCIS)*, Springer-Verlag, pages 1-16. An ICFEM 2014 Satellite Event.

[90] Klaus Havelund and Rajeev Joshi. Modeling and Monitoring of Hierarchical State Machines in Scala. 9th International Workshop on Software Engineering for Resilient Systems (SERENE 2017), September 4-5, 2017, Geneva, Switzerland. LNCS Volume TDB.

[91] Klaus Havelund and Rajeev Joshi. Modeling Rover Communication using Hierarchical State Machines with Scala. 2nd International Workshop on the Timing Performance

in Safety Engineering (TIPS 2017), 12 September, 2017, Trento, Italy. LNCS Volume TDB.

[92] Klaus Havelund, Rahul Kumar, Chris Delp and Bradley Clement. K: A Wide Spectrum Language for Modeling, Programming, and Analysis. 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016). 19-21 February 2016 - Rome, Italy. SciTePress. Pages 111-122. Editors: Slimane Hammoudi, Luís Ferreira Pires, Bran Selic, and Philippe Desfray.

[93] 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.

[94] 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.

[95] Klaus Havelund and Giles Reger. Runtime Verification Logics - A Language Design Perspective. KIMfest 2017, a conference in honour of Kim G. Larsen on the occasion of his 60th birthday, 19-20 August 2017, Aalborg University, Denmark. LNCS TBD.

[96] Klaus Havelund, Giles Reger, and Grigore Rosu. Runtime Verification - Past Experiences and Future Projections. Submitted to LNCS 10000, special issue in celebration of issue number 10,000 of Lecture Notes in Computer Science. Under review.

[97] 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**.

[98] 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.

[99] 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).

[100] 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 18*th *IEEE Real-Time Systems Symposium*, pages 2–13, Dec 1997. San Francisco, California, USA.

[101] 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.

[102] Klaus Havelund and Allen Goldberg. Verify Your Runs. In *Verified Software: Theories, Tools, Experiments (VSTTE'05)*, October 2005. Zurich, Switzerland.

[103] 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.

[104] 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.

[105] 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.

[106] Klaus Havelund, Doron Peled, and Dogan Ulus. First Order Temporal Logic Monitoring with BDDs. 17th Conference on Formal Methods in Computer-Aided Design (FMCAD 2017), 2-6 October, 2017, Vienna, Austria. LNCS Volume TDB.

[107] Klaus Havelund and Giles Reger. Specification of Parametric Monitors - Quantified Event Automata versus Rule Systems. SyDe Summer School 2015. In Rolf Drechsler and Ulrich Kuhne, editors, Formal Modeling and Verification of Cyber-Physical Systems (book), pages 151-189. 1st International Summer School on Methods and Tools for the Design of Digital Systems, Bremen, Germany, September 2015.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

[112] Sean Kauffman, Klaus Havelund, and Rajeev Joshi. `nfer` - A Notation and System for Inferring Event Stream Abstractions. In *Proceedings of the 16th International Conference on Runtime Verification*. Volume 10012 of Lecture Notes in Computer Science, pages 235-250, Springer-Verlag. Madrid, Spain, September 23-30, 2016.

[113] Sean Kauffman, Rajeev Joshi, and Klaus Havelund. Towards a Logic for Inferring Properties of Event Streams. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2016: 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation – Part II. Track: RVE - Runtime Verification and Enforcement, the (Industrial) Application Perspective*, organized by Ezio Bartocci and Ylies Falcone. Corfu, Greece, October 10-14, 2016. Volume 9952 of *LNCS*, pages 333-338, Springer-Verlag.

[114] L. Khatib, N. Muscettola, and K. Havelund. Verification of Plan Models using UPPAAL. In *First Goddard Workshop on Formal Approaches to Agent-Based Systems*, March 2000. NASA's Goddard Space center, Maryland, March 2000.

[115] 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.

[116] 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.

[117] 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.

[118] 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.

[119] 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.

[120] Giles Reger and Klaus Havelund. What is a Trace? A Runtime Verification Perspective. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA 2016: 7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation – Part II. Track: RVE - Runtime Verification and Enforcement, the (Industrial) Application Perspective*, organized by Ezio Bartocci and Ylies Falcone. Corfu, Greece, October 10-14, 2016. Volume 9952 of *LNCS*, pages 339-355, Springer-Verlag.

[121] Justin Seyster, Ketan Dixit, Xiaowan Huang, Radu Grosu, Klaus Havelund, Scott A. Smolka, Scott D. Stoller, and Erez Zadok. Aspect-Oriented Instrumentation with GCC. In *Proceedings of the 1st International Conference on Runtime Verification (RV 2010)*, St. Julians, Malta, November 1-4, 2010. Volume 6418 of *LNCS*.

[122] 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.

[123] Scott D. Stoller, Ezio Bartocci, Justin Seyster, Radu Grosu, Klaus Havelund, Scott A. Smolka, and Erez Zadok. Runtime Verification with State Estimation. In *Proceedings of the 2nd International Conference on Runtime Verification (RV 2011)*, San Francisco, California, USA, October 27-30, 2011. Volume 7186 of *LNCS*. **Won best paper award**.

[124] 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.

[125] 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.

[126] 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

[127] K. Havelund. Stepwise Development of a Denotational Stack Semantics. Master Thesis, DIKU, Department of Computer Science, University of Copenhagen, Denmark, 1984.

[128] 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.

NASA Technology Briefings

[129] Klaus Havelund, Rahul Kumar, Bradley Clement, and Chris Delp. The K Development Language (half a page). NASA Tech Briefs, July 2016. `http://viewer.zmags.com/publication/561bcaac#/561bcaac/60`.