Klaus Havelund, PhD

Senior Research Scientist and Principal

NASA Jet Propulsion Laboratory
Pasadena, California, USA

klaus.havelund@jpl.nasa.gov
www.havelund.com
+1-818-239-2178

Profile

Senior Research Scientist and Principal at NASA JPL with 30+ years in formal methods, runtime verification, and automated software analysis. 189 publications, 14,155 citations, h-index: 54.

Education

Ph.D. in Computer Science — University of Copenhagen, Denmark 1990–1994
PhD carried out in part at École Normale Supérieure, Paris
Master in Computer Science — University of Copenhagen, Denmark 1986

Professional Experience

NASA Jet Propulsion Laboratory (JPL), California 2006–Present

Senior Research Scientist (2009–Present) — Principal (2007–Present)

Kestrel Technology, California 2005–2006

Researcher

NASA Ames Research Center, California 1997–2005

Researcher

Earlier Research Positions

Aalborg University (1996–97), **Ecole Polytechnique and Paris 6 University** (1994–96), **DDC**[1]**/CRI, Denmark** (1984–90)

Student Programming Jobs

Skibsteknisk Laboratorium, **RC Datacenter**, **Christian Rovsing** (1977–1981)

Projects

Current JPL Projects

PyContract: Python library for runtime verification, used on Endurance rover for execution analysis and Europa Clipper session report analysis — **Daut:** Scala log analysis library (precursor to PyContract), used on Endurance — **Fuzz:** Python constraint-based tool for generating randomized command sequences with temporal logic constraints, applied to Europa Clipper FIT testing — **TaskSAT:** Python tool using constraint solving for verifying tasknets (MEXEC inputs), developed for Endurance project as part of PAAS strategic RTD — **JPL Coding Standards:** C++ (with Robert Bochino) and Python standards — **LLM Evaluation:** Evaluation of LLM-based programming tools (Windsurf, Claude Code), including F Prime application development — **Python Model Checker:** Developed with GenAI and University of Manchester as part of PAAS strategic RTD — **Other Research:** Combinatorial testing for Caleb Wagner's testing framework, Autonomica testing framework, STest implementation in Python, LLM-based log analysis

Past JPL Projects

Java Coding Standard (2014) — **DejaVu:** First-order past time temporal logic monitoring with BDDs — **nfer:** Logic and system for inferring interval abstractions — **K Language:** Constraint-based specification language for SysML class diagrams, developed with Bradley Clement — **TraceContract:** Scala log analysis API, used by NASA Ames during Lunar LADEE mission — **LogFire:** Scala implementation of RETE algorithm for runtime verification, used by Rajeev Joshi for Curiosity rover telemetry — **LogScope:** Python tool for checking log files against temporal logic specifications — **LogScope/C++:** Monitoring DSL with translation to C++ — **Armor:** CIL-based tool for instrumenting and monitoring C programs — **ScalaFM:** Exploring Scala for modeling (hierarchical state charts) — **Mcai 2.0:** Model Checking and Abstract Interpretation, NSF Expeditions project — **Ruler:** Runtime verification logic — **Survivable Software:** GCC-based framework for static analysis, instrumentation, and monitoring — **Lsccs:** Launch Site Command and Control System for space shuttle — **Map:** Translator from ASPEN planning language to Promela — **FPP State Machine DSL:** Contributions to F Prime sub-language for state machines

Past NASA Ames Projects

Java PathFinder (JPF): Pioneered and developed the first version of this model checker for Java programs

[1] https://en.wikipedia.org/wiki/Dansk_Datamatik_Center

Awards & Recognition

2021: Festschrift Workshop (Formal Methods in Outer Space) at ISoLA 2021 for 65th birthday — **2020**: SIGSOFT Impact Paper Award, JPL Magellan Award ($10K) — **2018**: RV Test of Time Award, Best Benchmark Award (DejaVu benchmark) — **2017**: JPL Voyager Award — **2016**: ASE Most Influential Paper — **2015**: CRV Competition Win (LogFire) — **2014**: ASE Most Influential Paper — **2011**: RV Best Paper, JPL Mariner Award — **2010**: JPL Ranger Award — **2009**: JPL Mariner Award, FLC Outstanding Technology Award — **2008–09**: Royal Academy Fellowship (UK) — **2008**: ACM Distinguished Paper, NASA Group Achievement — **2006**: NASA Tech Brief Award — **2003**: NASA TGIR Award — **2002**: EASST Best Paper

Invited Presentations

ASM'03 (Italy, 2003) — Joint CAV/ISSTA Special Event on Specification, Verification, and Testing of Concurrent Software (Boston, 2004) — Danish industry via CISS (Aalborg, 2004) — CASSIS'05 (Nice, 2005) — VVEIS'05 (Miami, 2005) — 8th JPL-GSFC Quality Mission Software Workshop (Santa Barbara, 2006) — PADTAD'06 (Portland, 2006) — ARTIST2 Summer School (Autrans, France, 2008) — Talk for 150 10-11 graders (Manchester, UK, 2009) — COMPASS'09 (York, UK, 2009) — 9th RV Workshop (Grenoble, 2009) — Software Reliability for Space Missions (Pasadena, 2009) — FMA'09 (Eindhoven, 2009) — EMSOFT'11 (Taipei, 2011) — SEFM'11 (Montevideo, 2011) — TORRENTS'11 (Toulouse, 2011) — HOWARD-60 (Manchester, 2011) — NFM'12 Panel (Norfolk, 2012) — SSS'12 (Minneapolis, 2012) — Marktoberdorf Summer School (2012) — LCCC workshop (Lund, 2012) — ISoLA'12 (Crete, 2012) — ICTSS'12 (Aalborg, 2012) — WODA'13 (Houston, 2013) — CERIST Autumn School (Algiers, 2013) — IFIP WG 1.9/2.15 Observer (Orlando, 2013; Vienna, 2014) — ISoLA'14 (Corfu, 2014) — TASE'14 (Changsha, 2014) — ISoLA'14 Tool Panel (Corfu, 2014) — FTSCS'14 (Luxembourg, 2014) — EITEC'15 (Seattle, 2015) — SyDe Summer School (Bremen, 2015) — IFIP WG 2.3 Observer (Pasadena, 2016) — CIT 2017 (Detroit, 2016) — RV 2016 (Madrid, 2016) — USC CS Colloquium (Los Angeles, 2016) — NFM'16 Panel (Minneapolis, 2016) — Software Correctness and Reliability Workshop (ETH Zurich, 2017) — SPIN 2018 with Doron Peled (Malaga, 2018) — ISoLA 2018 (Limassol, 2018) — ISoLA 2018 RV-TheToP Track with Doron Peled (Limassol, 2018) — RV 2018 with Grigore Rosu (Limassol, 2018) — ATVA 2020 with Doron Peled (Hanoi, virtual, 2020) — ESEC/FSE 2020 with Willem Visser (Sacramento, virtual, 2020) — ICTAC 2020 (Macau, virtual, 2020) — ISoLA 2022 (Rhodes, 2022) — JanFest 2023 invited paper (Bremen, 2023) — VORTEX 2024 (Vienna, 2024, presented by Doron Peled) — ICTAC 2025 (Marrakesh, 2025) — OSDF 2026 (JPL, 2026) — RE×AI 2026 (Los Angeles, 2026)

Academic Services

Conference Organization

Co-founder of Runtime Verification (RV) conference series (started 2001, became conference 2010) — RV 2001, 2002, 2004, 2006, 2010, 2011, 2014, 2016, 2017, 2021, RVCase'25 — SPIN 2000, 2008, 2017 — NFM 2011, 2015, 2022, 2026 (18th NFM at USC) — TACAS 2014 — VVPS workshops 2005, 2009, 2011 — Dagstuhl seminars on Runtime Verification 2007, 2010, 2014, 2017 — CPS School 2014 — ISoLA/AISoLA tracks: ISoLA 2016 (2 tracks), ISoLA 2018, ISoLA 2021, ISoLA 2022, AISoLA 2023, AISoLA 2024, AISoLA 2025, AISoLA 2026 — FM 2018 Industry Day

Program Committees

258 program committee memberships across 105 unique conferences/workshops: ABZ, AMDE, AOSD, ASARTI, ATVA, CAV, CONFENIS, DAW, DRHE, ECOOP, EITEC, Europar, FASE, FATES, FM, FMFun, FMI, FMICS, FMOODS & FORTE, FMPPTA, FMSPLE, FORMABS, FORTE, FTSCS, HAS, HSCC, HVC, I-Day@FM, ICAPS, ICFEM, ICI, ICSE, ICSEA, ICTAC, ICTSS, IEEE SMC-IT/SCC, INTELLI, ISSRE, ISoLA, InterAVT, Jan65, JPF, K, MOCHAP, MODELSWARD, MSVVEIS, MUSEPAT, MVI, MoChArt, Modularity, NFM, PADTAD, PATV, PLDI, PPMG, PrePost, REOS, RTT, RV, SAC-SVT, SATE, SAW, SEFM, SEW, SFEDL, SHM, SLE, SMC-IT, SPIN, SSV, SafePlan, Scala Days, ScottFest, SummerSim, SYNASC, TACAS, TAP, TASE, VMCAI, VORTEX, VSTTE, VVEIS, VVPS, WODA, iFM

Steering Committees

Runtime Verification (RV) conference (Chair 2010–2017, Member 2017–present) — NASA Formal Methods (NFM) symposium (Chair 2011–2017, Member 2017–present) — FME Advisory Committee — ETAPS (2014–2015) — SPIN (2009)

Editorial

FoMaC: Transactions on Foundations for Mastering Change

Other Service

Chair of the Formal Methods Europe (FME) Industry Committee — JPL's SRS (Senior Research Scientist) Council (served twice) — Member of undisclosed JPL review committees — Lecturer at annual F Prime workshops (2022–2023)

Memberships

ACM (Association for Computing Machinery), **IFIP WG 1.9/2.15** (Verified Software)

Publications

Complete List (189 publications — 14,155 citations — h-index: 54)

Books and Book Chapters (4)

1. The RAISE Language Group. *The RAISE Specification Language*. Prentice Hall, 1992. (Havelund wrote tutorial section, 250 pages)
2. Klaus Havelund. *RAISE in Perspective*. Chapter in *Logics of Specification Languages*, Springer, 2007.
3. Klaus Havelund, Giles Reger, Daniel Thoma, Eugen Zălinescu. *Monitoring Events that Carry Data*. Chapter in *Lectures on Runtime Verification*, LNCS 10457, Springer, 2018.
4. Ylies Falcone, Klaus Havelund, Giles Reger. *A Tutorial on Runtime Verification*. NATO Science for Peace and Security Series, IOS Press, 2013.

Journal Papers (38)

1. E. Ábrahám, K. Havelund. Some Recent Advances in Automated Analysis. STTT, 18(2):121-128, 2016.
2. R. Agarwal et al. Detection of Deadlock Potentials in Multithreaded Programs. IBM J. Research & Development, 54(5), 2010.
3. C. Artho et al. Combining Test-Case Generation and Runtime Verification. J. Theoretical Computer Science, 336(2-3), 2005.
4. C. Artho, K. Havelund, A. Biere. High-Level Data Races. Software Testing, Verification and Reliability, 13(4), 2004.
5. H. Barringer, A. Groce, K. Havelund, M. Smith. Formal Analysis of Log Files. J. Aerospace Computing, Info., and Comm., 7(11):365-390, 2010.
6. H. Barringer, D. Rydeheard, K. Havelund. Rule Systems for Run-Time Monitoring. J. Logic and Computation, 20(3), 2008.
7. E. Bartocci, Y. Falcone et al. First Intl. Competition on Runtime Verification. STTT, 2017.
8. M. Bennett, R. Borgen, K. Havelund, M. Ingham, D. Wagner. Prototyping a DSL for Monitor and Control Systems. J. Aerospace Computing, 7(11):338-364, 2010.
9. S. Bensalem, K. Havelund, A. Orlandini. Verification & Validation Meets Planning & Scheduling. STTT, 16(1), 2014.
10. D. Bjørner, A. Haxthausen, K. Havelund. Formal Model-oriented Software Development Methods. Future Generation Computer Systems, 7, 1992.
11. D. Bjørner, K. Havelund. A Half Century of Formal Methods. Submitted, under review, 2021.
12. E. Bodden, K. Havelund. Aspect-oriented Race Detection in Java. IEEE Trans. Software Engineering, 36(4):509-527, 2010.
13. G. Brat et al. Experimental Evaluation of V&V Tools on Martian Rover Software. Formal Methods in System Design, 25(2), 2004.
14. M. Broy et al. Does Every Computer Scientist Need to Know Formal Methods? Formal Aspects of Computing, 2024 (to appear).
15. J. Deshmukh, K. Havelund, I. Perez. Introduction to Selected Papers from NFM 2022. Innovations in Systems and Software Engineering, 2024.
16. H. Erdogmus, K. Havelund. Introduction to Selected Papers from SPIN 2017. STTT, 21, 2019.
17. Y. Eytani, K. Havelund, S. Stoller, S. Ur. Framework and Benchmark for Testing Tools for Multi-Threaded Programs. J. Concurrency and Computation, 2005.
18. C. George, K. Havelund, M. Nielsen, K.R. Wagner. The RAISE Language, Method and Tools. Formal Aspects of Computing, 1(1), 1989.
19. A. Groce, K. Havelund, G. Holzmann, R. Joshi, R-G. Xu. Establishing Flight Software Reliability. Annals of Math. and AI, accepted.
20. K. Havelund. Rule-based Runtime Verification Revisited. STTT, 17(2):143-170, 2015.
21. K. Havelund, G. Holzmann. Programming Event Monitors. STTT, 26:33-47, 2023.
22. K. Havelund, R. Kumar. Verified Change. FoMaC Transactions, submitted 2016.
23. K. Havelund, K.G. Larsen. The Fork Calculus. Nordic J. Computing, 1, 1994.
24. K. Havelund, M. Lowry, J. Penix. Formal Analysis of a Space Craft Controller using SPIN. IEEE Trans. Software Eng., 27(8):749-765, 2001.
25. K. Havelund, D. Peled. An Extension of LTL with Rules. STTT, 23, 2021.
26. K. Havelund, D. Peled, D. Ulus. First-Order Temporal Logic Monitoring with BDDs. FMSD, 56:1-21, 2020.
27. K. Havelund, D. Peled. On Monitoring Linear Temporal Properties. FMSD, 60:405-425, 2023.
28. K. Havelund, T. Pressburger. Model Checking Java Programs using Java PathFinder. STTT, 2(4):366-381, 2000.
29. K. Havelund, G. Roșu. Overview of Runtime Verification Tool Java PathExplorer. FMSD, 24(2), 2004.
30. K. Havelund, G. Roșu. Efficient Monitoring of Safety Properties. STTT, 6(2):158-173, 2004.
31. K. Havelund, W. Visser. Program Model Checking as a New Trend. STTT, 4(1), 2002.
32. S. Kauffman, K. Havelund, R. Joshi, S. Fischmeister. Inferring Event Stream Abstractions. FMSD, 2018.
33. S. Kauffman, K. Havelund, S. Fischmeister. What Can We Monitor Over Unreliable Channels? STTT, 23, 2021.
34. G. Roșu, K. Havelund. Rewriting-Based Techniques for Runtime Verification. Intl. J. Automated Software Eng., 12(2):151-197, 2005.
35. J. Seyster et al. InterAspect: Aspect-Oriented Instrumentation with GCC. FMSD, 41(3):295-320, 2012.
36. N. Shafiei, K. Havelund, P. Mehlitz. Concurrent Runtime Verification of Data Rich Events. STTT, 2023.
37. O. Sokolsky, K. Havelund, I. Lee. Introduction to Special Section on Runtime Verification. STTT, 14(3):243-247, 2012.
38. W. Visser, K. Havelund, G. Brat, S. Park, F. Lerda. Model Checking Programs. Intl. J. Automated Software Eng., 10(2), 2003.

Conference and Workshop Papers (147)

1. W. Ahrendt, B.K. Aichernig, K. Havelund. AI Assisted Programming (Track Intro). AISoLA 2024, LNCS.
2. W. Ahrendt, K. Havelund. AI Assisted Programming (Track Intro). ISoLA 2023, LNCS 14380.
3. W. Ahrendt, B. Aichernig, K. Havelund. AI Assisted Programming (Track Intro). AISoLA 2025, LNCS (to appear).
4. B. Aichernig, K. Havelund. AI-Assisted Programming with Test-based Refinement. ISoLA 2023, LNCS 14380.
5. B. Aichernig, K. Havelund. Correct-ish by Design. ISoLA/AISoLA 2024, LNCS.
6. J. Anton et al. Towards Industrial Scale Development of Custom Static Analyzers. Static Analysis Summit, NIST, 2006.
7. C. Artho et al. Experiments with Test Case Generation and Runtime Analysis. ASM'03, LNCS 2589, 2003.
8. C. Artho, K. Havelund, S. Honiden. Visualization of Concurrent Program Executions. SACT'07, 2007.
9. C. Artho, K. Havelund. Applying Jlint to Space Exploration Software. VMCAI'04, LNCS 2937, 2004.
10. C. Artho, K. Havelund, A. Biere. High-Level Data Races. VVEIS'03, 2003.
11. C. Artho, K. Havelund, A. Biere. Using Block-Local Atomicity to Detect Stale-Value Errors. ATVA'04, 2004.
12. C. Artho, K. Havelund, R. Kumar, Y. Yamagata. Domain-Specific Languages with Scala. ICFEM 2015, LNCS 9407, 2015.
13. H. Barringer, Y. Falcone, K. Havelund, G. Reger, D. Rydeheard. Quantified Event Automata. FM 2012, LNCS 7436.

14. H. Barringer, A. Goldberg, K. Havelund, K. Sen. Program Monitoring with LTL in Eagle. PADTAD'04, 2004.
15. H. Barringer, A. Goldberg, K. Havelund, K. Sen. Rule-Based Runtime Verification. VMCAI'04, LNCS 2937, 2004.
16. H. Barringer, A. Groce, K. Havelund, M. Smith. Entry Point for Formal Methods. FMA'09, 2009.
17. H. Barringer, K. Havelund. TraceContract: A Scala DSL for Trace Analysis. FM'11, LNCS 6664, 2011.
18. H. Barringer, K. Havelund. Internal vs External DSLs for Trace Analysis. RV 2011, 2011.
19. H. Barringer, K. Havelund, E. Kurklu, R. Morris. Checking Flight Rules with TraceContract. Scala Days 2011.
20. H. Barringer, K. Havelund, D. Rydeheard, A. Groce. Rule Systems for Runtime Verification: Tutorial. RV'09, LNCS 5779, 2009.
21. H. Barringer, D. Rydeheard, K. Havelund. Rule Systems for Run-Time Monitoring. RV'07, LNCS 4839, 2007.
22. M. Bennett, R. Borgen, K. Havelund, M. Ingham, D. Wagner. Development of Prototype DSL for M&C Systems. IEEE Aerospace, 2008.
23. S. Bensalem, J-C. Fernandez, K. Havelund, L. Mounier. Confirmation of Deadlock Potentials. PADTAD'06, 2006.
24. S. Bensalem, K. Havelund. Dynamic Deadlock Analysis of Multi-Threaded Programs. PADTAD'05, LNCS 3875, 2005.
25. D. Bjørner, K. Havelund. 40 Years of Formal Methods. FM 2014, LNCS 8442, 2014.
26. E. Bodden, K. Havelund. Racer: Effective Race Detection Using AspectJ. ISSTA'08, 2008. [**ACM Distinguished Paper**]
27. G. Brat et al. Experimental Evaluation of V&V Tools on Martian Rover Software. CMU/SEI Workshop, 2003.
28. M. Broy, K. Havelund, R. Kumar. Towards a Unified View of Modeling and Programming. ISoLA 2016, LNCS 9952, 2016.
29. M. Broy, K. Havelund, R. Kumar, B. Steffen. Unified View (Track Summary). ISoLA 2016, LNCS 9952, 2016.
30. M. Broy, K. Havelund, R. Kumar, B. Steffen. Unified View (Track Intro). ISoLA 2018, LNCS (TBD).
31. W. Buntine et al. Transformation Systems at NASA Ames. Software Transformation Systems, 1999.
32. I. Cohen, K. Havelund, M. Omer, D. Peled. Temporal Guardrails for LLM Conversations. Under review.
33. I. Cohen, K. Havelund, D. Peled, Y. Goldberg. Power of Reframing: Using LLMs in Synthesizing RV Monitors. RV 2025, LNCS 16087.
34. I. Cohen, K. Havelund, D. Peled, Y. Goldberg. Using LLMs in Synthesizing RV Monitors. Under review.
35. M. d'Amorim, K. Havelund. Event-Based Runtime Verification of Java Programs. WODA'05, 2005.
36. D. Dams, K. Havelund, S. Kauffman. A Python Library for Trace Analysis. RV 2022, LNCS 13498, 2022.
37. D. Dams, K. Havelund, S. Kauffman. Runtime Verification as Documentation. ISoLA 2022, LNCS 13702, 2022.
38. D. Drusinsky, K. Havelund. Execution-Based Model Checking of Interrupt-Based Systems. DSN'03 Workshop, 2003.
39. B. Duckett, K. Havelund, L. Stewart. Space Telemetry Analysis with PyContract. JanFest 2023, LNCS 14165, 2023.
40. M.S. Feather, S.L. Cornford, K. Havelund. Assurance of Model-Based Autonomy. RAMS 2022, 2022.
41. R.E. Filman, K. Havelund. Realizing Aspects by Transforming for Events. ASE'02, 2002.
42. R.E. Filman, K. Havelund. Source-Code Instrumentation and Quantification. Workshop on Foundations of AOP, 2002.
43. A. Goldberg, K. Havelund. Automated Runtime Verification with MEXICO. RV'05, ENTCS 144, 2005.
44. A. Goldberg, K. Havelund. Instrumentation of Java Bytecode for Runtime Analysis. FATES'05, 2005.
45. K. Havelund. Using Runtime Analysis to Guide Model Checking of Java Programs. SPIN'00, LNCS 1885, 2000.
46. K. Havelund. Monitoring with Data Automata. ISoLA 2014, LNCS 8803, 2014.
47. K. Havelund. Rule-Based Runtime Verification of Cyber-Physical Systems. ISoLA 2014, LNCS 8803, 2014.
48. K. Havelund. Closing the Gap between Specification and Programming. ISoLA 2016, LNCS 9952, 2016.
49. K. Havelund. Data Automata in Scala. ICTAC 2014, LNCS 8687, 2014.
50. K. Havelund. Rethinking Model Checking of Java Programs. FM 2018, LNCS 10951, 2018.
51. K. Havelund. Monitoring the Execution of Space Craft Flight Software. ISoLA 2018, LNCS 11245, 2018.
52. K. Havelund. The Many Roles of Runtime Verification. RV 2021, LNCS 12974, 2021.
53. K. Havelund, P. Alexander. If-Then-Else Theorems for Sets. Festschrift Klaus Grue, LNCS 7850, 2013.
54. K. Havelund, C. Artho, V. Jagannathan et al. What's Not in Your Benchmark? DaCapo Concurrency. PADTAD'10, 2010.
55. K. Havelund, S. Bensalem. Models of Behavior Composition (Track Intro). ISoLA 2010, LNCS 6415, 2010.
56. K. Havelund, I. Becerra, M.S. Feather. Three Mission Scenarios for Data Analysis Using Model Checking. NFM 2024, LNCS 14627, 2024.
57. K. Havelund, M. Broy, R. Kumar, B. Steffen. Revisiting Unified View 10 Years Later. ISoLA 2025, LNCS (to appear).
58. K. Havelund, M.S. Feather, I. Perez. Temporal Logic Anomaly Detection. VORTEX 2023, 2023.
59. K. Havelund, A. Goldberg. Verify Your Runs. VMCAI'05, LNCS 3385, 2005.
60. K. Havelund, G. Holzmann. Effectiveness of Randomization in Spin. SPIN'09, LNCS 5578, 2009.
61. K. Havelund, R. Joshi. Checking Flight Rules During Anomaly Resolution. FMA'11, 2011.
62. K. Havelund, R. Joshi. Using Aspect-Oriented Programming to Extend JPF. SPIN'08, LNCS 5156, 2008.
63. K. Havelund, R. Joshi, C. Artho. First-Order Runtime Verification of Multithreaded Java Programs. SPIN'09, LNCS 5578, 2009.
64. K. Havelund, B. Johnson. Algorithms for Garbage Collection With Multiple Generations. VDM-Europe'91, LNCS 551, 1991.
65. K. Havelund, R. Kumar. Outline of a Specification Language. ISoLA 2016, LNCS 9953, 2016.
66. K. Havelund, R. Kumar. The K Language. WOET'14, 2014.
67. K. Havelund, R. Kumar, D. Smith. Flight Rule Verification by Model Checking. MOCHAP 2015, EPTCS 187, 2015.
68. K. Havelund, K.G. Larsen. The Fork Calculus. ICALP'93, LNCS 700, 1993.
69. K. Havelund, G. Liu, C. Pasareanu. Automated Event Stream Analysis with Prefix Tree Acceptors. ISoLA 2018, LNCS 11247, 2018.
70. K. Havelund, G. Liu, C. Pasareanu. Model Inference of Event Streams. ISoLA 2020, LNCS 12476, 2020.
71. K. Havelund, D. Peled. Efficient Runtime Verification of First-Order Temporal Properties. SPIN'18, LNCS 10869, 2018.
72. K. Havelund, D. Peled. BDD-Based Runtime Verification. ISoLA 2018, LNCS 11247, 2018.
73. K. Havelund, D. Peled. Measuring First-Order Temporal Logic Monitors. ISoLA 2020, LNCS 12478, 2020.
74. K. Havelund, D. Peled. An Extension of LTL with Rules and its Application to RV. FMCAD 2020, 2020.
75. K. Havelund, D. Peled, D. Ulus. First-Order Temporal Logic Monitoring with BDDs. FMCAD 2017, 2017.
76. K. Havelund, D. Peled, D. Ulus. DejaVu Benchmark. RV 2018, LNCS 11237, 2018. [**Best Benchmark Award**]
77. K. Havelund, A. Pnueli. Programs, Protocols and Timing Diagrams. ECCS'97, LNCS 1414, 1997.
78. K. Havelund, T. Pressburger. Model Checking Java Programs using Java PathFinder. STTT Special Issue, 2000.
79. K. Havelund, G. Roşu. Monitoring Java Programs with Java PathExplorer. RV'01, ENTCS 55(2), 2001. [**RV 2018 Test of Time**]
80. K. Havelund, G. Roşu. Synthesizing Monitors for Safety Properties. TACAS'02, LNCS 2280, 2002. [**EASST Best Paper**]
81. K. Havelund, G. Roşu. Monitoring Programs using Rewriting. ASE'01, 2001. [**ASE 2016 Most Influential**]
82. K. Havelund, G. Roşu. Testing Linear Temporal Logic Formulae on Finite Execution Traces. PDMC'01, 2001.
83. K. Havelund, G. Roşu. Runtime Verification Workshop Series. RV'04, 2004.
84. K. Havelund, N. Shankar. Experiments in Theorem Proving and Model Checking for Protocol Verification. FME'96, LNCS 1051, 1996.
85. K. Havelund, M. Sirjani, F. Arbab, E.B. Johnsen. Monitoring Actors and Objects. ISoLA 2012, LNCS 7610, 2012.
86. K. Havelund, L. Stewart, N. Shafiei. Analyzing Space Flight Logs with TraceContract. SMC-IT/SCC 2023, 2023.
87. K. Havelund, W. Visser. Program Model Checking as a New Trend. STTT, 2002.
88. K. Havelund, W. Visser. Dynamic Partial Order Reduction for Model Checking Software. SPIN'00, LNCS 1885, 2000.
89. K. Havelund et al. Formal Analysis of the Remote Agent Before and After Flight. SPIN'00, LNCS 1885, 2000.
90. G. Holzmann, K. Havelund. Checking Code Generated by MDD Tools. ISoLA 2012, LNCS 7609, 2012.
91. M. Ingham, K. Havelund. Challenges in Verifying Autonomous Systems. AIAA Space 2007, 2007.
92. M. Ingham, K. Havelund, A. Barrett, R. Rasmussen. Issues in Runtime Assertion Checking. MOCHAP 2012, AAAI Technical Report, 2012.
93. R. Joshi, K. Havelund. Towards a Framework for Analyzing Fault Diagnostics. FMA'12, 2012.
94. R. Joshi, K. Havelund. Monitoring Fault-Tolerant System Recovery: A Tutorial. COMPASS'10, 2010.
95. S. Kauffman, K. Havelund. nfer: Inferring Event Stream Abstractions. RV 2016, LNCS 10012, 2016.
96. S. Kauffman, K. Havelund. Decentralized Traceability for Fault Diagnosis. MOCHAP 2016, ICAPS Workshop, 2016.
97. S. Kauffman, K. Havelund, R. Joshi. nfer - A Notation and System for Inferring Event Stream Abstractions. RV 2016, LNCS 10012, 2016.

98. S. Kauffman, K. Havelund, R. Joshi, S. Fischmeister. Semantics of Event Stream Processing. ISoLA 2016, LNCS 9952, 2016.
99. S. Kauffman, K. Havelund, S. Fischmeister. What Can We Monitor Over Unreliable Channels? RV 2019, LNCS 11757, 2019.
100. S. Kauffman, K. Havelund. Presentation Abstract: Runtime Verification Tools. RRRR 2025, 2025.
101. R. Kumar, K. Havelund. Event-Based Monitoring of UML/SysML State Machines. ISoLA 2014, LNCS 8802, 2014.
102. R. Kumar, K. Havelund. Comparing Different Mechanisms for Trace Checking. LPAR-21, EPiC 46, 2017.
103. R. Kumar, K. Havelund. Towards a Framework for Hierarchical SysML Models. ISoLA 2016, LNCS 9952, 2016.
104. C. Lourenço, K. Havelund, E. Uchitel. Automata-Based Monitoring in Presence of Predictions. RV 2024, LNCS 15228, 2024.
105. C. Lourenço, K. Havelund, Y. Falcone. Comparing Monpoly and DejaVu. RV 2021, LNCS 12974, 2021.
106. C. Lourenço, K. Havelund, Y. Falcone. Monitoring with Predictions and Explanations. ISoLA 2022, LNCS 13704, 2022.
107. C. Lourenço, K. Havelund, E. Uchitel. A Framework for Monitoring with Predictions and Explanations. ISoLA 2024, LNCS (to appear).
108. Y. Lu, D. Peled, K. Havelund, R. Cohen. Runtime Verification through Forward Chaining. ISoLA 2024, LNCS (to appear).
109. G. Mancini, F. Tiezzi, K. Havelund. Runtime Verification of DeFi Smart Contracts. ISoLA 2024, LNCS (to appear).
110. R. Meyer, K. Havelund, D. Peled. Synthesizing Optimal Collective Algorithms. VMCAI 2020, LNCS 11990, 2020.
111. C. Mitra, K. Havelund. Reachability Analysis Using Message Sequence Charts. SPIN'10, LNCS 6349, 2010.
112. F. Mora, M. Bersani, K. Havelund, L. Rossi. Towards Formal Methods for Aerial Swarms. ISoLA 2024, LNCS (to appear).
113. F. Mora, M. Bersani, K. Havelund, X. Qu. A First-Order Logic with Frames. TACAS 2024, LNCS 14571, 2024.
114. F. Mora, M. Bersani, K. Havelund, L. Rossi. QTL Over Finite Traces. TIME 2024, 2024.
115. F. Mora, P. Polgreen, K. Havelund. Hardware-in-the-Loop Runtime Verification. MEMOCODE 2023, 2023.
116. F. Mora, P. Polgreen, K. Havelund. PyAAL: Python Library for AMBA. MEMOCODE 2024, 2024.
117. F. Mora, P. Polgreen, K. Havelund. AMBA Property Specification Language. NFM 2023, LNCS 13903, 2023.
118. F. Mora, C. Presotto, M. Bersani, K. Havelund, S. Cheikh. A First-Order Logic with Frames as Data Type. SPIN 2024, LNCS 14624, 2024.
119. F. Mora, C. Presotto, K. Havelund, M. Bersani. Coinductive Stream Analysis with LTL and QTL. VORTEX 2023, 2023.
120. C. Muñiz, F. Pardo, K. Havelund. Flexible Monitoring of Temporal Specifications. RV 2022, LNCS 13498, 2022.
121. D. Peled, K. Havelund. Refining the Safety–Liveness Classification. FM 2018, LNCS 10951, 2018.
122. D. Peled, K. Havelund, G. Penczek. Model Checking Relaxed. ISoLA 2020, LNCS 12476, 2020.
123. D. Peled, K. Havelund, S. Tavolato. Monitoring Timed Properties (Tutorial). RV 2020, LNCS 12399, 2020.
124. D. Peled, K. Havelund. Runtime Verification: From Propositional to First-Order Temporal Logic. RV 2018, LNCS 11237, 2018.
125. I. Perez, K. Havelund. Debugging with Stack Traces. ISoLA 2024, LNCS (to appear).
126. A. Rademaker, K. Havelund, N. Flatt. LDDL: Logical Declarative Distributed Languages. ISoLA 2012, LNCS 7609, 2012.
127. G. Reger, K. Havelund. What is a Trace? A Runtime Verification Perspective. ISoLA 2016, LNCS 9953, 2016.
128. G. Reger, K. Havelund. Specification of Parametric Monitors. ISoLA 2018, LNCS 11245, 2018.
129. J. Rushby, K. Havelund. Integrating Model Checking and Theorem Proving. FME'96, LNCS 1051, 1996.
130. A. Salman, K. Havelund. RuleRunner: A Python Tool for Reactive Rule-Based Monitoring. VSTTE 2023, LNCS 14415, 2023.
131. K. Sen, G. Roşu, K. Havelund. Runtime Safety Analysis of Multithreaded Programs. ESEC/FSE'03, 2003.
132. J. Seyster et al. Runtime Verification with State Estimation. RV 2011, LNCS 7186, 2011. [**Best Paper Award**]
133. N. Shafiei, K. Havelund. Efficient Runtime Monitoring using Vector Clocks. ATVA 2020, LNCS 12302, 2020.
134. N. Shafiei, K. Havelund, P. Mehlitz. Fast and Precise Runtime Verification. ISoLA 2020, LNCS 12477, 2020.
135. N. Shafiei, K. Havelund, P. Mehlitz. First-Order Logic Runtime Monitoring with Data. RV 2020, LNCS 12399, 2020.
136. N. Shafiei, K. Havelund, P. Mehlitz. DejaVu: A Tool for Temporal Verification. ISoLA 2018, LNCS 11247, 2018.
137. V. D'Silva, K. Havelund et al. New Challenges for the Verification Community. FMCAD 2011, 2011.
138. M. Sirjani, K. Havelund, E.B. Johnsen. Towards Compositional Verification. FM 2020, LNCS 12233, 2020.
139. T. Sullivan, K. Havelund. The Path from Specification to Testing: Are We There Yet? ISoLA 2025, LNCS (to appear).
140. E. Timmons, M. Reeves, B. Ayton, B.C. Williams, M.D. Ingham, J. Castillo-Rogez, W. Seto, K. Havelund, A. Jasour, B. Donitz, D. Mages, A. Rahmani, P. Tavallali, S. Chung. Information-Driven and Risk-Bounded Autonomy for Scientist Avatars. ASCEND 2021, 2021.
141. W. Visser, K. Havelund, G. Brat, S. Park. Model Checking Programs. ASE 2000, 2000. [**2020 SIGSOFT Impact, 2014 ASE Most Influential**]
142. W. Visser, K. Havelund, G. Brat, S. Park, F. Lerda. Model Checking Programs. ASE'03, LNCS 2791, 2003.
143. W. Visser, K. Havelund, G. Brat, S. Park. Java PathFinder - Second Generation of a Java Model Checker. WAVE'00, 2000.
144. W. Visser, K. Havelund, J. Penix. Adding Active Objects to SPIN. SPIN Workshops, LNCS 1680, 1999.
145. R. Washington, K. Golden, J. Bresina, D.E. Smith, C. Anderson, T. Smith, K. Havelund. Model-Based Programming of Exploratory Robots. ISAIRAS'99, 1999.
146. Y. Watanabe, C. Artho, K. Havelund, Y. Tanabe. Fuzz-Based Test Case Generation. ICST 2012, 2012.
147. S. Zhioua, S. Riff, R. Boulifa, K. Havelund. Monitoring Event Frequencies. ISoLA 2020, LNCS 12477, 2020.

Teaching

Most of my career has been spent in research labs, so my teaching experience is limited. Appointed Lecturer in Computer Science at California Institute of Technology: taught classes on program monitoring and runtime verification (2008, 2009).