Alex David Groce

Education

Ph.D., **Computer Science**, Carnegie Mellon University, March 2005
Thesis Title: **Error Explanation and Fault Localization with Distance Metrics**.
Thesis Committee: Edmund Clarke (chair), David Garlan, Reid Simmons, and Willem Visser

B.S., **Computer Science**, North Carolina State University, May 1999 (*summa cum laude*)

B.S., **Multidisciplinary Studies**, North Carolina State University, May 1999 (*summa cum laude*)

Minors: **English Literature** and **Science, Technology, and Society**

Experience

Summary: I have over fifteen years of experience in research, development, and analysis of complex software systems. My work focuses on testing, developing, specifying, and understanding critical distributed, embedded, aerospace, systems, and security software and software/hardware systems, in order to increase the reliability, security, and efficiency of these systems. I contributed significantly to the implementation and design of verification and analysis tools used internationally for research, industrial application, and teaching, including NASA's Java PathFinder 2, CBMC, MAGIC, SyMP, JPL's LogScope, the Concurrency Workbench (NC), DeepState, Slither, and TSTL. Recently I have been developing languages and systems tools to make software testing and development of secure code easier for real-world users without a research background. I have been PI or co-PI on externally funded grants totalling over $9.1M ($1.2M my share), and have authored or co-authored more than 100 publications, primarily in ACM, IEEE, and top verification and formal methods venues, with over 4,500 citations, an h-index of 36, and an i10-index of 76, according to Google Scholar. I served on the program committees for ICSE, ASE, and ISSTA multiple times in recent years, write a regular column for ACM SIGSOFT Software Engineering Notes, and am an associate editor for IEEE Software and ACM Transactions on Software Engineering and Methodology. I make 500-1,500 GitHub commits in a typical year, and have reported over 100 bugs *that have been fixed as a result*, in widely used open source software systems as part of my research.

8/2019–present · **Consulting Security Engineer, Trail of Bits:** Continued exploring novel smart contract analysis methods and tools, surveyed sources of smart contract flaws in high value contracts, and potential for various automated vulnerability discovery approaches.

2/2019–8/2019 · **Security Engineer, Trail of Bits (Sabbatical):** Explored novel approaches to analyzing smart contracts and blockchain implementations, combining static and dynamic methods. Contributed to development of Echidna, Slither, and other tools for smart contract analysis.

1/2017–present · **Associate Professor with Tenure, School of Informatics, Computing, and Cyber Systems, Northern Arizona University:** Explored methods for bringing advanced automated testing techniques into development practice, especially for Python and other languages used widely in bioinformatics and other scientific applications. Continued to advance state-of-the-art in software testing.

9/2015–12/2016 · **Associate Professor with Tenure, School of Electrical Engineering and Computer Science, Oregon State University:** Continued investigations of core problems in software and software/hardware systems understanding and correctness, including lightweight tools for automated testing and understanding by non-experts. Began investigations of self-repairing and evolving systems for software longevity. Co-chaired undergraduate curriculum committee.

6/2009–9/2015 · **Assistant Professor, School of Electrical Engineering and Computer Science, Oregon State University:** Initiated investigation of methods and metrics for end-user testing of machine learning systems. Continued work on unified approaches to testing, with new techniques applied to finding flaws in widely used production-quality C compilers, JavaScript engines, and embedded file systems. Mentored graduate students, taught graduate and undergraduate classes on software engineering and verification, and served on graduate and undergraduate curriculum committees and the hiring committee.

2/2011–5/2015 · **Consultant, Aries Design Automation, LLC:** Provided expertise on verification, testing, and analysis for SBIR and other projects and proposals. Main consultant on NIST SBIR project "Using Automated Abstractions to Classify System States for Software Health Monitoring," aimed at improving systems engineering of health monitoring systems via code analysis and machine learning.

4/2008–6/2008, 4/2009-6/2009 · **Lecturer in Computer Science, part-time, California Institute of Technology:** Taught CS 119, Reliable Software: Testing and Monitoring. Used research and JPL flight system experiences to introduce state-of-the-art techniques and practical methods for testing and verification to students, with a focus on automated approaches to executing software in order to find faults.

4/2005–6/2009 · **Laboratory for Reliable Software, Jet Propulsion Laboratory:** Led test automation development and design, Mars Science Laboratory (Curiosity rover) Flight Software Internal Test Team. Introduced new techniques for exploiting traces in static analysis of programs, integrated model checking and dynamic analysis, developed a successful random testing approach for mission file systems, and contributed to modeling and verification of Dawn mission launch sequence and fault protection. Led file system acceptance testing for a NASA Discovery class mission; led model checking and random testing efforts for Mars Science Laboratory (Curiosity) file storage modules; contributed to design for file systems used to store images, science products, and telemetry during spaceflight missions. Served on design and code review panels for flight software systems and hardware drivers. Worked with systems engineers to develop methods for specifying, generating, and understanding logs of complex spacecraft software and hardware activity.

8/1999–3/2005 · **Doctoral student, Carnegie Mellon University:** Invented methods for error explanation and fault isolation using distance metrics, applied to aerospace, security, and micro-kernel code. Invented and implemented novel approaches for counterexample guided abstraction refinement and heuristic search guided model checking. Implemented Athena proof system for security protocols, devised language and type system for encoding security protocols in the SyMP tool. Enriched teaching by incorporating research ideas into instruction, assignments, and evaluation in undergraduate classes.

5/2002–8/2002 · **Summer Student Research Program, RIACS (Research Institute for Advanced Computer Science)/NASA Ames Research Center, Robust Software Engineering group:** Invented and implemented methods for error explanation, using model checking counterexamples to provide automatic feedback about the causes and location of errors in complex systems.

5/2001–8/2001 · **Summer Student Research Program, RIACS/NASA Ames Research Center, Robust Software Engineering group:** Invented and experimented with novel (and successful) heuristics for model checking Java programs; implemented heuristic search in the Java PathFinder 2 model checker.

5/2000–8/2000 · **Research intern, Bell Laboratories (Murray Hill):** Implemented black box checking algorithm (model checking for an unknown model using finite-state machine learning algorithms) and investigated theoretical aspects and applications of the algorithm to software model checking.

5/1999–8/1999 · **Summer research assistant, SUNY Stony Brook:** Continued work from the previous summer.

5/1998–8/1998 · **Summer research assistant, North Carolina State University:** Implemented a model checker based on Alternating Büchi Tableau Automata and developed logical optimizations for ABTAs.

Research Interests

Designing, *specifying*, *coding*, *testing*, *verifying*, *understanding*, and *debugging* complex computer systems. My research has combined testing, static analysis, formal methods, programming languages, and machine learning approaches as required. My current focus is on techniques for testing critical systems including compilers, web browsers, file systems, libraries and infrastructure, and more generally, all layers of modern computer system environments. In particular, I am interested in bridging the gap between "security" approaches to automated testing (e.g. fuzzers like AFL and libFuzzer, binary analysis tools) and those more rooted in the software testing community; since "bugs are bugs" there is no reason for a deep divide in aggressive testing practice or theory. In the spirit of Henry Petroski's proposal that progress in engineering arises from understanding failures, I believe that a deeper understanding of **bugs** is essential to better *software and systems engineering* and better engineering education.

Topics

Software testing: I am exploring the effectiveness and relative ease of (randomized) testing, and the relationship between testing, runtime verification or dynamic analysis, and model checking using unsound abstractions — including shared models and frameworks for testing and model checking and strategies based on constraint-solving and machine learning. I am aiding core developers of the Linux kernel to use mutation analysis to improve kernel systems testing methods, and to verify critical algorithms.

Software model checking: I continue to investigate the use of SAT and SMT solvers for bounded model checking (CBMC), predicate abstraction-based approaches (MAGIC, SATABS), and explicit-state exploration with SPIN and Java PathFinder.

Understanding complex program executions: I am working with scientists at the United States Forest Service to analyze complex models used to predict climate change impacts, with techniques that should also apply to analyzing model checking and test system executions.

Educational use of analysis tools: I am interested in incorporating mature, robust "research" tools for system design, debugging, and verification (model checking, random testing, static analysis, delta-debugging, etc.) into engineering education: I believe such tools not only make for better engineering practice, but make the learning experience more rewarding and interesting to students.

Honors

ACM 2017 **Senior Member**

ACM/IEEE International Conference on Software Engineering 2016 **Distinguished Poster Award**

IEEE International Conference on Software Testing, Verification and Validation (ICST) 2014 **Best Paper Award**

NASA Software of the Year Award for Mars Science Laboratory Flight Software, 2013 (MSL Flight Software team)

NASA Space Act Award for LogScope Software, 2011

National Science Foundation Faculty **Early Career Development (CAREER) Program Award**, 2011

JPL Mariner Award for LogScope Testing Software, 2009

JPL Spot Award (for Multi Mission System Architecture Platform (MSAP) File System Testing), 2006

ACM/IEEE International Conference on Software Engineering 2003 **ACM SIGSOFT Distinguished Paper Award**

NASA "Engineering Innovation" Turning Goals Into Reality (TGIR) Award 2003 (Java PathFinder team)

National Science Foundation Graduate Fellowship

NCSU Class of 1999 College of Humanities and Social Sciences Scholar (valedictorian for CHASS)

Phi Beta Kappa

Funding

"RcppDeepState, a simple way to fuzz test compiled code in R packages", PI: Toby Hocking, Co-PI: Alex Groce, R Consortium Infrastructure Steering Committee Grants, $34,000, January 2020-December 2020.

"Interfaces, Models, and Monitoring for Resource-aware Transformations that Augment the Lifecycle of Systems (IMMoRTALS)", PIs: Matt Gillen (BBN), Doug Schmidt (Vanderbilt), Eric Walkingshaw (Oregon State University), Heng Yin (Syracuse), Co-PIs: Alex Groce, Jules White (Vanderbilt), Jacob Staples (Securboration), DARPA BRASS (Building Resource Adaptive Software Systems), BAA-15-36, $464,625, $1.6M total Oregon State University budget, (total project budget $7.7M), October 2015-September 2019.

"Advanced Tools for Effective Automated Test Generation", PIs: Miroslav Velev (Aries Design Automation, LLC), Alex Groce, *NASA Small Business Technology Transfer Phase I* T11.01-9878, $52,037 (total budget $125,000), July 2015-2016.

"Explorations of Testing in the Cloud", PI: Alex Groce, Amazon Web Services in Education Grants, $10,000, January 2015-2017.

"II-EN: Software Tools for Monte-Carlo Optimization", PI: Alan Fern, Co-PIs: Alex Groce, Sinisa Todorovic, Prasad Tadepalli, Thomas Dietterich, *National Science Foundation* CNS-1406049, $442,366, October 2014-September 2017 (infrastructure development for cloud-based optimization tools with ML, graphics, and testing applications; includes $12,000 dedicated compute time for software testing research).

"Diversity and Feedback in Random Testing for Systems Software", PIs: Alex Groce, John Regehr (University of Utah), *National Science Foundation* CCF-1217824, $242,244 (total budget $491,280), September 2012-2015.

"CAREER: Integrating Automated Software Testing Methods", PI: Alex Groce, *National Science Foundation* CCF-1054876, $400,000, September 2011-2016.

Books, Edited Volumes

Alex Groce and Stefan Leue (eds). Proceedings of the 2nd International Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies. Electronic Proceedings in Theoretical Computer Science, Volume 259, October 2017.

Alex Groce and Madanlal Musuvathi (eds). Model Checking Software: Proceedings of the 18th International SPIN Workshop. Springer-Verlag, LNCS 6823, 2011.

Refereed Journal Publications

Josie Holmes, Iftekhar Ahmed, Caius Brindescu, Rahul Gopinath, He Zhang, and **Alex Groce**. Using Relative Lines of Code to Guide Automated Test Generation for Python. *ACM Transactions on Software Engineering and Methodology*, 29(4): 28-1:28-38, 2020.

Josie Holmes and **Alex Groce**. Using Mutants to Help Developers Distinguish and Debug (Compiler) Faults. *Journal of Software Testing, Verification, and Reliability*, 30(2), 2020 (online first).

Alex Groce, Iftekhar Ahmed, Carlos Jensen, Paul E. McKenney, and Josie Holmes. How Verified (or Tested) is My Code? Falsification-Driven Verification and Testing. *Automated Software Engineering Journal*, 25(4): 917-960, December 2018.

Josie Holmes, **Alex Groce**, Jervis Pinto, Pranjal Mittal, Pooria Azimi, Kevin Kellar, and James O'Brien. TSTL: the Template Scripting Testing Language. *International Journal on Software Tools for Technology Transfer*, 20(1):57-78, February 2018.

Rahul Gopinath, Iftekhar Ahmed, Mohammad Amin Alipour, Carlos Jensen, and **Alex Groce**. Mutation Reduction Strategies Considered Harmful. *IEEE Transactions on Reliability*, 66(3): 854-874, September 2017.

Rahul Gopinath, Iftekhar Ahmed, Mohammad Amin Alipour, Carlos Jensen, and **Alex Groce**. Does Choice of Mutation Tool Matter? *Software Quality Journal*, 25(3):871-920, September 2017.

Alex Groce, Mohammad Amin Alipour, Chaoqiang Zhang, Yang Chen, and John Regehr. Cause Reduction: Delta Debugging, Even Without Bugs. *Journal of Software Testing, Verification and Reliability*, 26(1):40-68, January 2016.

Milos Gligoric, **Alex Groce**, Chaoqiang Zhang, Rohan Sharma, Mohammad Amin Alipour, and Darko Marinov. Guidelines for Coverage-Based Comparisons of Non-Adequate Test Suites. *ACM Transactions on Software Engineering and Methodology*, 24(4):4-37, August 2015.

Alex Groce, Klaus Havelund, Gerard Holzmann, Rajeev Joshi, and Ru-Gang Xu. Establishing Flight Software Reliability: Testing, Model Checking, Constraint-Solving, Monitoring and Learning. *Annals of Mathematics and Artificial Intelligence*, 70(4):315-348, April 2014.

Alex Groce, Todd Kulesza, Chaoqiang Zhang, Shalini Shamasunder, Margaret Burnett, Weng-Keen Wong, Simone Stumpf, Shubhomoy Das, Amber Shinsel, Forrest Bice, and Kevin McIntosh. You Are the Only Possible Oracle: Effective Test Selection for End Users of Interactive Machine Learning Systems. *IEEE Transactions on Software Engineering*, 40(3):307-323, March 2014.

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. Swarm Verification Techniques. *IEEE Transactions on Software Engineering*, 37(6):845-857, November 2011.

Howard Barringer, **Alex Groce**, Klaus Havelund, and Margaret Smith. Formal Analysis of Log Files. *Journal of Aerospace Computing, Information, and Communication*, 7(11):365-390, December 2010.

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. Model Driven Code Checking. *Automated Software Engineering Journal*, 15(3-4):283-297, December 2008.

Alex Groce and Rajeev Joshi. Exploiting Traces in Static Program Analysis: Better Model Checking through *printf*s. *International Journal on Software Tools for Technology Transfer*, 10(2):131-144, March 2008.

Alex Groce, Doron Peled, and Mihalis Yannakakis. Adaptive Model Checking. *Logic Journal of the IGPL*, 14(5):729-744, October 2006.

Alex Groce, Sagar Chaki, Daniel Kroening, and Ofer Strichman. Error Explanation with Distance Metrics. *International Journal on Software Tools for Technology Transfer*, 8(3):229-247, June 2006.

Alex Groce and Willem Visser. Heuristics for Model Checking Java Programs. *International Journal on Software Tools for Technology Transfer*, 6(4):260-276, August 2004.

Sagar Chaki, Edmund Clarke, **Alex Groce**, Joel Ouaknine, Ofer Strichman, and Karen Yorav. Efficient Verification of Sequential and Concurrent C Programs. *Formal Methods in System Design, Special Issue on Software Model Checking*, 25(2-3):129-166, September-November 2004.

Sagar Chaki, Edmund Clarke, **Alex Groce**, Somesh Jha, and Helmut Veith. Modular Verification of Software Components in C. *IEEE Transactions on Software Engineering*, 30(6):388-402, June 2004.

Refereed Conference and Workshop Publications

Alex Groce and Josie Holmes. Practical Automatic Lightweight Nondeterminism and Flaky Test Detection and Debugging for Python. *IEEE International Conference on Software Quality, Reliability, and Security*, pages 188-195, Macau, China, December 2020.

Miroslav Gavrilov, Kyle Dewey, **Alex Groce**, Davina Zamanzadeh, and Ben Hardekopf. A Practical, Principled Measure of Fuzzer Appeal: A Preliminary Study. *IEEE International Conference on Software Quality, Reliability, and Security*, pages 510-517, Macau, China, December 2020.

Gustavo Grieco, Will Song, Artur Cygan, Josselin Feist, and **Alex Groce**. Echidna: Effective, Usable, and Fast Fuzzing for Smart Contracts. *International Symposium on Software Testing and Analysis*, pages 557-560, Los Angeles, California, July 2020 (tool demonstration track, acceptance rate 31%).

John-Paul Smith, **Alex Groce**, Gustavo Grieco, and Josselin Feist. Echidna: A Practical Smart Contract Fuzzer. *International Conference on Financial Cryptography and Data Security*, Kota Kinabalu, Sabah, Malaysia, February 2020 (poster track).

Alex Groce, Josselin Feist, Gustavo Grieco, and Michael Colburn. What are the Actual Flaws in Important Smart Contracts (and How Can We Find Them)? *International Conference on Financial Cryptography and Data Security*, pages 634-653, Kota Kinabalu, Sabah, Malaysia, February 2020 (22% acceptance rate).

Aprit Christi, **Alex Groce**, and Austin Wellman. Building Resource Adaptations via Test-Based Software Minimization: Application, Challenges, and Opportunities. *IEEE International Symposium on Software Reliability Engineering*, pages 73-78, Berlin, Germany, October 2019 (industry track), **Nominated for Best Disruptive Paper - Industry Track**.

Mark Mossberg, Felipe Manzano, Eric Hennenfent, **Alex Groce**, Gustavo Grieco, Josselin Feist, Trent Brunson, and Artem Dinaburg. Manticore: A User-Friendly Symbolic Execution Framework for Binaries and Smart Contracts. *IEEE/ACM International Conference on Automated Software Engineering*, pages 1186-1189, San Diego, California, November 2019 (demonstrations track, acceptance rate 54%).

Arpit Christi, **Alex Groce**, and Rahul Gopinath. Evaluating Fault Localization for Resource Adaptation via Test-based Software Modification. *IEEE International Conference on Software Quality, Reliability, and Security*, pages 26-33, Sofia, Bulgaria, July 2019.

Josselin Feist, Gustavo Grieco, and **Alex Groce**. Slither: A Static Analysis Framework for Smart Contracts. *International Workshop on Emerging Trends in Software Engineering for Blockchain*, pages 8-15, Montreal, Canada, May 2019.

Josie Holmes and **Alex Groce**. Causal Distance-Metric-Based Assistance for Debugging After Compiler Fuzzing. *IEEE International Symposium on Software Reliability Engineering*, pages 166-177, Memphis, Tennessee, October 2018 (acceptance rate 24%), **invited for journal submission to STVR**.

Arpit Christi, Matthew Olson, Mohammad Amin Alipour, and **Alex Groce**. Reduce Before You Localize: Delta-Debugging and Spectrum-Based Fault Localization. *IEEE International Workshop on Debugging and Repair*, pages 184-191, Memphis, Tennessee, October 2018.

Peter Goodman, Gustavo Grieco, and **Alex Groce**. Tutorial: DeepState: Bringing Vulnerability Detection Tools into the Development Cycle. *IEEE Cybersecurity Development Conference (SecDev)*, pages 130-131, Boston, Massachusetts, September-October 2018 (tutorials track, acceptance rate 55%).

Arpit Christi and **Alex Groce**. Target Selection for Test-Based Resource Adaptation. *IEEE International Conference on Software Quality, Reliability, and Security*, pages 458-469, Lisbon, Portugal, July 2018 (acceptance rate 19%).

Alex Groce, Josie Holmes, Darko Marinov, August Shi, and Lingming Zhang. An Extensible, Regular-Expression-Based Tool for Multi-Language Mutant Generation. *ACM/IEEE International Conference on Software Engineering*, pages 25-28, Gothenburg, Sweden, May-June 2018 (demonstrations track).

Peter Goodman and **Alex Groce**. DeepState: Symbolic Unit Testing for C and C++. NDSS Workshop on Binary Analysis Research, San Diego, California, February 2018.

Alex Groce and Josie Holmes. Provenance and Pseudo-Provenance for Seeded Learning-Based Automated Test Generation. *NIPS 2017 Interpretable ML Symposium*, Long Beach, California, December 2017.

Arpit Christi, **Alex Groce**, and Rahul Gopinath. Resource Adaptation via Test-Based Software Minimization. *IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 61-70, Tucson, Arizona, September 2017 (acceptance rate 21%).

Alex Groce, Josie Holmes, and Kevin Kellar. One Test to Rule Them All. *ACM International Symposium on Software Testing and Analysis*, pages 1-11, Santa Barbara, California, July 2017 (acceptance rate 26%).

Alex Groce, Paul Flikkema, and Josie Holmes. Towards Automated Composition of Heterogeneous Tests for Cyber-Physical Systems. *Workshop on Testing Embedded and Cyber-Physical Systems*, pages 12-15, Santa Barbara, California, July 2017.

Josie Holmes and **Alex Groce**. A Suite of Tools for Making Effective Use of Automatically Generated Tests. *ACM International Symposium on Software Testing and Analysis*, pages 356-359, Santa Barbara, California, July 2017 (Tools and Demonstrations track).

Rahul Gopinath, Carlos Jensen, and **Alex Groce**. The Theory of Composite Faults. *IEEE International Conference on Software Testing, Verification and Validation*, pages 47-57, Tokyo, Japan, March 2017 (acceptance rate 27%).

Iftekhar Ahmed, Carlos Jensen, **Alex Groce**, and Paul McKenney. Applying Mutation Analysis on Kernel Test Suites: An Experience Report. *International Workshop on Mutation Analysis*, pages 110-115, Tokyo, Japan, March 2017.

Iftekhar Ahmed, Rahul Gopinath, Caius Brindescu, **Alex Groce**, and Carlos Jensen. Can Testedness be Effectively Measured? *ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, pages 547-558, Seattle, Washington, November 2016 (acceptance rate 27%).

Josie Holmes, **Alex Groce**, and Mohammad Amin Alipour. Mitigating (and Exploiting) Test Reduction Slippage. *Workshop on Automated Software Testing*, pages 66-69, Seattle, Washington, November 2016.

Ali Aburas and **Alex Groce**. A Method Dependence Relations Guided Genetic Algorithm. *International Symposium on Search-Based Software Engineering*, pages 267-273, Raleigh, North Carolina, October 2016 (short paper track).

Mohammad Amin Alipour, August Shi, Rahul Gopinath, Darko Marinov, and **Alex Groce**. Evaluating Non-Adequate Test-Case Reduction. *IEEE/ACM International Conference on Automated Software Engineering*, pages 16-26, Singapore, Singapore, September 2016 (acceptance rate 20%).

Mohammad Amin Alipour, **Alex Groce**, Rahul Gopinath, and Arpit Christi. Generating Focused Random Tests Using Directed Swarm Testing. *ACM International Symposium on Software Testing and Analysis*, pages 70-81, Saarbrucken, Germany, July 2016 (acceptance rate 25%).

Pranjal Mittal and **Alex Groce**. Poster: TSTL: A Little Language for Automated Testing Written in Python. *PyCon 2016*, Portland, Oregon, May-June 2016.

Rahul Gopinath, Amin Alipour, Iftekhar Ahmed, Carlos Jensen, and **Alex Groce**. On the Limits of Mutation Reduction Strategies. *ACM/IEEE International Conference on Software Engineering*, pages 511-522, Austin, Texas, May 2016 (acceptance rate 19%).

Rahul Gopinath, Carlos Jensen, and **Alex Groce**. Poster: Topsy-Turvy: A Smarter and Faster Parallelization of Mutation Analysis. *ACM/IEEE International Conference on Software Engineering*, pages 740-743, Austin, Texas, May 2016 (poster track, acceptance rate 58%), **Distinguished Poster Award**.

Rahul Gopinath, Mohammad Amin Alipour, Iftekhar Ahmed, Carlos Jensen, and **Alex Groce**. Measuring Effectiveness of Mutant Sets. *International Workshop on Mutation Analysis*, pages 132-141, Chicago, Illinois, April 2016.

Alex Groce, Iftekhar Ahmed, Carlos Jensen, and Paul E. McKenney. How Verified is My Code? Falsification-Driven Verification. *IEEE/ACM International Conference on Automated Software Engineering*, pages 737-748, Lincoln, Nebraska, November 2015 (acceptance rate 21%), **invited for journal submission to ASE**.

Rahul Gopinath, Mohammad Amin Alipour, Iftekhar Ahmed, Carlos Jensen, and **Alex Groce**. How Hard Does Mutation Analysis Have to Be, Anyway? *IEEE International Symposium on Software Reliability Engineering*, pages 216-227, Gaithersburg, Maryland, November 2015 (acceptance rate 32%).

Alex Groce, Jervis Pinto, Pooria Azimi, and Pranjal Mittal. TSTL: A Language and Tool for Testing (Demo). *ACM International Symposium on Software Testing and Analysis*, pages 414-417, Baltimore, Maryland, July 2015 (Tools and Demonstrations track).

Alex Groce and Jervis Pinto. A Little Language for Testing. *NASA Formal Methods Symposium*, pages 204-218, Pasadena, California, April 2015 (acceptance rate 31%).

Yuanli Pei, Arpit Christi, Xiaoli Fern, **Alex Groce**, and Weng-Keen Wong. Taming a Fuzzer Using Delta Debugging Trails. *International Workshop on Software Mining*, Shenzhen, China, December 2014.

Rahul Gopinath, Carlos Jensen, and **Alex Groce**. Mutations: How Close are they to Real Faults? *IEEE International Symposium on Software Reliability Engineering*, pages 189-200, Naples, Italy, November 2014 (acceptance rate 25%).

Alex Groce, Mohammad Amin Alipour, and Rahul Gopinath. Coverage and Its Discontents. *ACM Symposium on New Ideas in Programming and Reflections on Software, Onward! Essays, part of SPLASH (ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity)*, pages 255-268, Portland, Oregon, October 2014.

Ali Aburas and **Alex Groce**. An Improved Memetic Algorithm with Method Dependence Relations (MAMDR). *International Conference on Quality Software*, pages 11-20, Dallas, Texas, October 2014 (acceptance rate 26%).

Chaoqiang Zhang, **Alex Groce**, and Mohammad Amin Alipour. Using Test Case Reduction and Prioritization to Improve Symbolic Execution. *ACM International Symposium on Software Testing and Analysis*, pages 60-70, San Jose, California, July 2014 (acceptance rate 28%).

Duc Le, Mohammad Amin Alipour, Rahul Gopinath, and **Alex Groce**. MuCheck: an Extensible Tool for Mutation Testing of Haskell Programs. *ACM International Symposium on Software Testing and Analysis*, pages 429-432, San Jose, California, July 2014 (Tools and Demonstration track).

Rahul Gopinath, Carlos Jensen, and **Alex Groce**. Code Coverage for Suite Evaluation by Developers. *ACM/IEEE International Conference on Software Engineering*, pages 72-82, Hyderabad, India, May-June 2014 (acceptance rate 20%).

Alex Groce, Mohammad Amin Alipour, Chaoqiang Zhang, Yang Chen, and John Regehr. Cause Reduction for Quick Testing. *IEEE International Conference on Software Testing, Verification and Validation*, pages 243-252, Cleveland, Ohio, March-April 2014 (acceptance rate 28%), **Best Paper Award, invited for journal submission to STVR**.

Amin Alipour, **Alex Groce**, Chaoqiang Zhang, Anahita Sanadaji, and Gokul Caushik. Finding Model-Checkable Needles in Large Source Code Haystacks: Modular Bug-Finding via Static Analysis and Dynamic Invariant Discovery. *International Workshop on Constraints in Formal Verification*, San Jose, California, November 2013.

Alex Groce, Chaoqiang Zhang, Mohammad Amin Alipour, Eric Eide, Yang Chen, and John Regehr. Help, Help, I'm Being Suppressed! The Significance of Suppressors in Software Testing. *IEEE International Symposium on Software Reliability Engineering*, pages 390-399, Pasadena, California, November 2013 (acceptance rate 35%).

Milos Gligoric, **Alex Groce**, Chaoqiang Zhang, Rohan Sharma, Amin Alipour, and Darko Marinov. Comparing Non-adequate Test Suites using Coverage Criteria. *ACM International Symposium on Software Testing and Analysis*, pages 302-313, Lugano, Switzerland, July 2013 (acceptance rate 26%), **invited for journal submission to ACM TOSEM**.

Yang Chen, **Alex Groce**, Chaoqiang Zhang, Weng-Keen Wong, Xiaoli Fern, Eric Eide, and John Regehr. Taming Compiler Fuzzers. *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 197-208, Seattle, Washington, June 2013 (acceptance rate 17%).

Alex Groce, Alan Fern, Jervis Pinto, Tim Bauer, Mohammad Amin Alipour, Martin Erwig, and Camden Lopez. Lightweight Automated Testing with Adaptation-Based Programming. *IEEE International Symposium on Software Reliability Engineering*, pages 161-170, Dallas, Texas, November 2012 (acceptance rate 30%).

Mohammad Amin Alipour and **Alex Groce**. Extended Program Invariants: Applications in Testing and Fault Localization. *International Workshop on Dynamic Analysis*, pages 7-11, Minneapolis, Minnesota, July 2012.

Alex Groce and Martin Erwig. Finding Common Ground: Choose, Assert, and Assume. *International Workshop on Dynamic Analysis*, pages 12-17, Minneapolis, Minnesota, July 2012.

Alex Groce, Chaoqiang Zhang, Eric Eide, Yang Chen, and John Regehr. Swarm Testing. *ACM International Symposium on Software Testing and Analysis*, pages 78-88, Minneapolis, Minnesota, July 2012 (acceptance rate 29%).

Alex Groce. Coverage Rewarded: Test Input Generation via Adaptation-Based Programming. *IEEE/ACM International Conference on Automated Software Engineering*, pages 380-383, Lawrence, Kansas, November 2011 (short paper, acceptance rate 35%).

Mohammad Amin Alipour and **Alex Groce**. Bounded Model Checking and Feature Omission Diversity. *International Workshop on Constraints in Formal Verification*, San Jose, California, November 2011.

Amber Shinsel, Todd Kulesza, Margaret Burnett, William Curran, **Alex Groce**, Simone Stumpf, and Weng-Keen Wong. Mini-Crowdsourcing End-User Assessment of Intelligent Assistants: A Cost-Benefit Study. *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 47-54, Pittsburgh, Pennsylvania, September 2011 (acceptance rate 35%).

Todd Kulesza, Margaret Burnett, Simone Stumpf, Weng-Keen Wong, Shubhomoy Das, **Alex Groce**, Amber Shinsel, Forrest Bice and Kevin McIntosh. Where Are My Intelligent Assistant's Mistakes? A Systematic Testing Approach. *International Symposium on End-User Development*, pages 171-186, Brindisi, Italy, June 2011 (acceptance rate 40%).

Alex Groce, Klaus Havelund, and Margaret Smith. From Scripts to Specifications: the Evolution of a Flight Software Testing Effort. *ACM/IEEE International Conference on Software Engineering*, pages 129-138, Cape Town, South Africa, May 2010 (Software Engineering in Practice, acceptance rate 23%).

Alex Groce. (Quickly) Testing the Tester via Path Coverage. *International Workshop on Dynamic Analysis*, Chicago, Illinois, July 2009.

James Andrews, **Alex Groce**, Melissa Weston, and Ru-Gang Xu. Random Test Run Length and Effectiveness. *IEEE/ACM International Conference on Automated Software Engineering*, pages 19-28, L'Aquila, Italy, September 2008 (acceptance rate 12%).

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. Tackling Large Verification Problems with the Swarm Tool. *SPIN Workshop on Model Checking of Software*, pages 134-143, Los Angeles, California, August 2008.

Klaus Havelund, **Alex Groce**, Gerard Holzmann, Rajeev Joshi, and Margaret Smith. Automated Testing of Planning Models. *Workshop on Model Checking and Artificial Intelligence*, pages 90-105, Patras, Greece, July 2008.

Alex Groce and Rajeev Joshi. Random Testing and Model Checking: Building a Common Framework for Nondeterministic Exploration. *International Workshop on Dynamic Analysis*, pages 22-28, Seattle, Washington, July 2008.

Alex Groce and Rajeev Joshi. Extending Model Checking with Dynamic Analysis. *Conference on Verification, Model Checking and Abstract Interpretation*, pages 142-156, San Francisco, California, January 2008 (acceptance rate 34%).

Nicolas Blanc, **Alex Groce**, and Daniel Kroening. Verifying C++ with STL Containers via Predicate Abstraction. *IEEE/ACM International Conference on Automated Software Engineering*, pages 521-524, Atlanta, Georgia, November 2007 (short paper, acceptance rate 25%).

Alex Groce, Gerard Holzmann, and Rajeev Joshi. Randomized Differential Testing as a Prelude to Formal Verification. *ACM/IEEE International Conference on Software Engineering*, pages 621-631, Minneapolis, Minnesota, May 2007 (Software Engineering in Practice, acceptance rate 27%).

Alex Groce and Rajeev Joshi. Exploiting Traces in Program Analysis. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 379-393, Vienna, Austria, March-April 2006 (acceptance rate 25%), **invited for journal submission to STTT**.

Daniel Kroening, **Alex Groce**, and Edmund Clarke. Counterexample Guided Abstraction Refinement via Program Execution. *International Conference on Formal Engineering Methods*, pages 224-238, Seattle, Washington, November 2004 (acceptance rate 27%).

Sagar Chaki, **Alex Groce**, and Ofer Strichman. Explaining Abstract Counterexamples. *ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, pages 73-82, Newport Beach, California, October-November 2004 (acceptance rate 15%).

Alex Groce, Daniel Kroening, and Flavio Lerda. Understanding Counterexamples with `explain`. *International Conference on Computer Aided Verification*, pages 453-456, Boston, Massachusetts, July 2004 (tool paper).

Alex Groce and Daniel Kroening. Making the Most of BMC Counterexamples. *Workshop on Bounded Model Checking*, pages 71-84, Boston, Massachusetts, July 2004.

Alex Groce. Error Explanation with Distance Metrics. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 108-122, Barcelona, Spain, March-April 2004 (acceptance rate 26%), **invited for journal submission to STTT**.

Sagar Chaki, Edmund Clarke, **Alex Groce**, and Ofer Strichman. Predicate Abstraction with Minimum Predicates. *Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, pages 19-34, L'Aquila, Italy, October 2003 (acceptance rate 37%).

Edjard Mota, Edmund Clarke, W. Oliveira, **Alex Groce**, J. Kanda, and M. Falcao. VeriAgent: an Approach to Integrating UML and Formal Verification Tools. *Brazilian Workshop on Formal Methods*, pages 111-129, Universidade Federal de Campina Grande, Brazil, October 2003.

Sagar Chaki, Edmund Clarke, **Alex Groce**, Somesh Jha, and Helmut Veith. Modular Verification of Software Components in C. *ACM/IEEE International Conference on Software Engineering*, pages 385-395, Portland, Oregon, May 2003 (acceptance rate 13%), **ICSE SIGSOFT Distinguished Paper Award, invited for journal submission to IEEE TSE**.

Alex Groce and Willem Visser. What Went Wrong: Explaining Counterexamples. *SPIN Workshop on Model Checking of Software*, pages 121-135, Portland, Oregon, May 2003.

Alex Groce and Willem Visser. Model Checking Java Programs using Structural Heuristics. *ACM International Symposium on Software Testing and Analysis*, pages 12-21, Rome, Italy, July 2002 (acceptance rate 19%).

Alex Groce, Doron Peled, and Mihalis Yannakakis. AMC: An Adaptive Model Checker. *International Conference on Computer Aided Verification*, pages 521-525, Copenhagen, Denmark, July 2002 (tool paper).

Alex Groce and Willem Visser. Heuristic Model Checking for Java Programs. *SPIN Workshop on Model Checking of Software*, pages 242-245, Grenoble, France, April 2002 (tool paper).

Alex Groce, Doron Peled, and Mihalis Yannakakis. Adaptive Model Checking. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 357-370, Grenoble, France, April 2002 (acceptance rate 31%).

Girish Bhat, Rance Cleaveland, and **Alex Groce**. Efficient Model Checking Via Büchi Tableau Automata. *International Conference on Computer Aided Verification*, pages 38-52, Paris, France, July 2001 (acceptance rate 29%).

Invited Papers

Miroslav Velev, Chaoqiang Zhang, Ping Gao, and **Alex Groce**. Exploiting Abstraction, Learning from Random Simulation, and SVM Classification for Efficient Dynamic Prediction of Software Health Problems. *International Symposium on Quality Electronic Design*, Santa Clara, CA, March 2015.

Alex Groce, Alan Fern, Martin Erwig, Jervis Pinto, Tim Bauer, and Mohammad Amin Alipour. Learning-Based Test Programming for Programmers. *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, pages 572-586, Heraclion, Crete, October 2012.

Howard Barringer, **Alex Groce**, Klaus Havelund, and Margaret Smith. Formal Analysis of Log Files. *SMC-IT Workshop on Software Reliability for Space Missions*, Pasadena CA, July 2009.

Howard Barringer, **Alex Groce**, Klaus Havelund, and Margaret Smith. An Entry Point for Formal Methods: Specification and Analysis of Event Logs. *1st Workshop on Formal Methods in Aerospace*, Electronic Proceedings of Theoretical Computer Science (EPTCS), Eindhoven, Holland, November 2009.

Howard Barringer, Klaus Havelund, David Rydeheard, and **Alex Groce**. Rule Systems for Runtime Verification: A Short Tutorial. *International Workshop on Runtime Verification*, pages 1-24, Grenoble, France, June 2009.

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. Swarm Verification. *IEEE/ACM International Conference on Automated Software Engineering*, pages 1-6, L'Aquila, Italy, September 2008.

Alex Groce, Gerard Holzmann, Rajeev Joshi, and Ru-Gang Xu. Putting Flight Software Through the Paces with Testing, Model Checking, and Constraint-Solving. *International Workshop on Constraints in Formal Verification*, pages 1-15, Sydney, Australia, August 2008.

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. New Challenges in Model Checking. *25 Years of Model Checking*, pages 65-76, Seattle, Washington, August 2006.

Columns, Book Reviews, and Magazine Articles

Alex Groce. Henry Petroski's *The Pencil*. Passages column, *ACM SIGSOFT Software Engineering Notes*, accepted for publication.

Alex Groce. Robert K. Merton's *On the Shoulders of Giants: A Shandean Postscript: The Post-Italianate Edition*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 45(4), 8-10, October 2020.

Alex Groce. Federico Biancuzzi and Shane Warden's *Masterminds of Programming: Conversations with the Creators of Major Programming Languages*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 45(3), 4-5, July 2020.

Alex Groce. John McPhee's *Basin and Range*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 45(2): 3-4, April 2020.

Alex Groce. Martin Gardner's *The Last Recreations: Hydras, Eggs, and Other Mathematical Mystifications*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 45(1): 4-5, January 2020.

Alex Groce. Gerald M. Weinberg's *An Introduction to General Systems Thinking*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 44(3): 10-11, July 2019.

Alex Groce. Gerald M. Weinberg's *The Psychology of Computer Programming: Silver Anniversary Edition*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 44(2): 3-4, April 2019.

Alex Groce. Simson Garfinkel, Daniel Weise, and Steven Strassman's *The UNIX-Haters Handbook*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 44(1): 7, January 2019.

Alex Groce. Tom DeMarco and Timothy Lister's *Peopleware: Productive Projects and Teams (Third Edition)*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 43(4): 6-7, October 2018.

Alex Groce. Confucius' *Analects*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 43(3): 5, July 2018.

Alex Groce. Bruce Sterling's *Ascendancies: The Best of Bruce Sterling*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 43(2): 6-7, April 2018.

Alex Groce. Karl Popper's *The Logic of Scientific Discovery*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 43(1), 4-5, January 2018.

Alex Groce. Brian Kernighan and P. J. Plauger's *The Elements of Programming Style (Second Edition)*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 42(4): 5, October 2017.

Alex Groce. Charles Petzold's *Code: The Hidden Language of Computer Hardware and Software*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 42(3): 9, July 2017.

Alex Groce. Herbert A. Simon's *The Sciences of the Artificial (Third Edition)*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 42(2): 5-6, April 2017.

Alex Groce. Daniel P. Friedman and Matthias Felleisen's *The Little Schemer - 4th edition*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(6): 5-6, November 2016.

Alex Groce. Jon Bentley's *More Programming Pearls: Confessions of a Coder*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(5): 6, September 2016.

Alex Groce. Sherry Turkle's *The Second Self: Computers and the Human Spirit*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(4): 6-7, July 2016

Alex Groce. Samuel C. Florman's *The Existential Pleasures of Engineering*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(3): 4-5, May 2016.

Alex Groce. Edward R. Tufte's *The Visual Display of Quantitative Information*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(2): 5, March 2016.

Alex Groce. David Agans' *Debugging: the 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 41(1): 5, January 2016.

Alex Groce. Donald E. Knuth's *Selected Papers on Computer Science*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 40(3): 4-5, May 2015.

Alex Groce. George Polya's *How to Solve It: A New Aspect of Mathematical Method*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 40(2): 5-6, March 2015.

Alex Groce. Hugh Kenner's *The Mechanic Muse*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 40(1): 8-9, January 2015.

Alex Groce. Andrew Hunt and David Thomas' *The Pragmatic Programmer: from journeyman to master*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(6): 6-7, November 2014.

Alex Groce. Vernor Vinge's *A Deepness in the Sky*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(5): 5, September 2014.

Alex Groce. Tom DeMarco and Timothy Lister's *Waltzing with Bears: Managing Risk on Software Projects*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(4): 8-9, July 2014.

Alex Groce. Henry Petroski's *To Engineer is Human: the Role of Failure in Successful Design*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(3): 6-7, May 2014.

Alex Groce. Jon Bentley's *Programming Pearls*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(2): 4-5, March 2014.

Alex Groce. Tracy Kidder's *The Soul of a New Machine*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 39(1): 6-7, January 2014.

Alex Groce. Frederick P. Brooks, Jr.'s *The Mythical Man-Month: Essays on Software Engineering*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 38(6): 6-7, November 2013.

Alex Groce. Charles Babbage's *Passages from the Life of a Philosopher*. Passages column, *ACM SIGSOFT Software Engineering Notes*, 38(5): 17-18, September 2013.

Technical Reports

Duc Le, Mohammad Amin Alipour, Rahul Gopinath, and **Alex Groce**. Mutation Testing of Functional Programming Languages. Technical Report, School of Computer Science and Electrical Engineering, Oregon State University, 2014.

Jamie Andrews, Yihao Ross Zhang, and **Alex Groce**. Comparing Automated Unit Testing Strategies. Technical Report 736, Department of Computer Science, University of Western Ontario, December 2010.

Brian Kernighan, Dennis Ritchie, Doug McIlroy, Eddie Benowitz, Scott Burleigh, Tim Canham, Benjamin Cichy, Ken Clark, Micah Clark, Len Day, Robert Denise, Will Duquette, Dan Dvorak, Dan Eldred, Ed Gamble, Peter Gluck, Kim Gostelow, Chris Grasso, **Alex Groce**, Dave Hecox, Gerard Holzmann, Joe Hutcherson, Rajeev Joshi, Roger Klemm, Frank Kuykendall, Mary Lam, Steve Larson, Todd Litwin, Tom Lockhart, Lloyd Manglapus, Kenny Meyer, Alex Murray, Al Niessner, Bob Rasmussen, Len Reder, Glenn Reeves, Kirk Reinholtz, Mike Roche, Nicolas Rouquette, Steve Scandore, Marcel Schoppers, Dave Smyth, Ken Starr, Igor Uchenik, Dave Wagner, Garth Watney, Steve Watson, Matt Wette, and Jesse Wright. JPL Institutional Coding Standard for the C Programming Language. Jet Propulsion Laboratory, web publication, March 3, 2009 (`http://lars-lab.jpl.nasa.gov/JPL\_Coding\_Standard\_C.pdf`).

Nicolas Blanc, Daniel Kroening, and **Alex Groce**. Verifying C++ with STL Containers via Predicate Abstraction. Technical Report 506, ETH Zürich, January 2006.

Alex Groce. Error Explanation and Fault Localization with Distance Metrics. (Ph.D. Thesis) Technical Report CMU-CS-05-121, Carnegie Mellon University, March 2005.

Alex Groce, Doron Peled, and Mihalis Yannakakis. AMC: An Adaptive Model Checker. ALR-2002-008, Avaya Labs Research, February 2002.

Alex Groce and Willem Visser. What Went Wrong: Explaining Counterexamples. Technical Report 02-08, RIACS, USRA, February 2002.

Alex Groce, Doron Peled, and Mihalis Yannakakis. Adaptive Model Checking. ALR-2002-002, Avaya Labs Research, January 2002.

Sergey Berezin and **Alex Groce**. SyMP: The Hacker's Manual. Carnegie Mellon University, web publication, May 12, 2001 (`http://www.cs.cmu.edu/~modelcheck/symp.html`).

Sergey Berezin and **Alex Groce**. SyMP: The User's Guide. Carnegie Mellon University, web publication, December 27, 2000 (`http://www.cs.cmu.edu/~modelcheck/symp.html`).

Professional Activities and Service

Associate Editor, ACM TOSEM (Transactions on Software Engineering and Methodology), August 2020-present.

Associate Editor, IET Software, January 2019-present.

ACM TOSEM (Transactions on Software Engineering and Methodology) Board of Distinguished Reviewers, February 2019-present.

Associate Editor for Software Testing, IEEE Software, September 2018-present.

Reviewer for *IEEE Transactions on Software Engineering* (TSE), *ACM Transactions on Software Engineering and Methodology* (TOSEM), *Journal of the ACM* (JACM), *Software Tools for Technology Transfer* (STTT), *Formal Methods in System Design* (FMSD), *IEEE Transactions on Parallel and Distributed Systems* (TPDS), *IEEE Transactions on Reliability*, *IEEE Transactions on Embedded Computing Systems* (TECS), *Information Processing Letters* (IPL), *IEEE Transactions on Computers* (TC), *IEEE Transactions on Industrial Informatics*, *Empirical Software Engineering*, *Acta Informatica*, *ACM Computing Surveys*, *Algorithmica*, *Computers & Security*, *Microprocessors and Microsystems: Embedded Hardware Design*, *Journal on Satisfiability, Boolean Modeling and Computation* (JSAT), *Journal of Logic and Computation*, *Journal of Parallel and Distributed Computing* (JPDC), *Artificial Intelligence*, *Journal of Computer and System Sciences* (JCSS), *Software Testing, Verification and Reliability* (STVR), *Automated Software Engineering Journal*, *Annals of Mathematics and Artificial Intelligence* (AMAI), *Science of Computer Programming*, *ACM Computing Surveys*, *Journal of Applied Logic* (JAL), *Journal of Computer Science and Technology* (JCST), *Journal of Systems and Software* (JSS), *PLOS ONE*, *Information and Software Technology* (IST), *The Computer Science Journal*, *Journal of Software: Evolution and Process*,*International Conference on Computer Aided Verification* (CAV), *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (TACAS), *ACM SIGSOFT Symposium on the Foundations of Software Engineering* (FSE), *IEEE/ACM International Conference on Automated Software Engineering* (ASE), *ACM/IEEE International Conference on Software Engineering* (ICSE), *ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages* (POPL), *ACM SIGPLAN Conference on Programming Language Design and Implementation* (PLDI), *ACM CHI Conference on Human Factors in Computing Systems* (CHI), *Verification, Model Checking, and Abstract Interpretation* (VMCAI), *International Conference on Software Testing, Verification and Validation* (ICST), *Symposium on Automated and Analysis-driven Debugging* (AADEBUG), *Fundamental Approaches to Software Engineering* (FASE), *NASA Formal Methods Symposium* (NFM), *ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (EICS), *Formal Techniques for Networked and Distributed Systems* (FORTE), *Asia and South Pacific Design Automation Conference* (ASP-DAC), *Logic for Programming, Artificial Intelligence, and Reasoning* (LPAR), *Formal Methods in Computer-Aided Design* (FMCAD), *Australasian Computer Science Conference* (ACSC), *SPIN Workshop on Model Checking of Software*, *Workshop on Model Checking and Artificial Intelligence* (MoChArt), *Specification and Verification of Component-Based Systems* (SAVCBS), *Workshop on Verification and Debugging* (V&D), *International Workshop on Constraints in Formal Verification* (CFV), *Workshop on Software Model Checking* (SoftMC), *Workshop on Verification and Validation for Planning and Scheduling Systems* (VVPS), *IEEE Software Engineering Workshop* (SEW), *Java Pathfinder Workshop* (JPF), *International Workshop on Dynamic Analysis* (WODA), and *Workshop on Formal Methods for Industrial Critical Systems* (FMICS).

Senior member of ACM (Association for Computing Machinery).

ACM, Special Interest Group on Software Engineering (SIGSOFT), Special Interest Group on Programming Languages (SIGPLAN), Special Interest Group on Embedded Systems (SIGBED) member.

ACM SIGSOFT "Passages" classic book review columnist for SIGSOFT Software Engineering Notes.

IEEE, IEEE Computer Society member.

External Reviewer for Natural Sciences and Engineering Research Council of Canada.

Reviewer for NASA Small Business Innovation Research (SBIR) program.

External Reviewer for Israel Science Foundation.

External Reviewer for South African National Research Foundation.

Panel and Committee Service

Program committee · 45th ACM/IEEE International Conference on Software Engineering 2022 (ICSE'22), Pittsburgh, Pennsylvania, May 2022.

Program committee · 36th ACM/IEEE International Conference on Automated Software Engineering (ASE'21), Melbourne, Australia, November 2021.

Program committee · 15th International Conference on Tests and Proofs (TAP'21), Bergen, Norway, June 2021.

Program committee · 14th IEEE International Conference on Software Testing, Validation, and Verification (ICST'21), Porto de Galinhas, Brazil, April 2021.

Program committee · 16th International Workshop on Mutation Analysis (MUTATION 2021), Porto de Galinhas, Brazil, April 2021.

Program committee · 9th International Conference on Model-Driven Engineering and Software Development (MODELSWARD'21), Vienna, Austria, February 2021.

Program committee · 35th ACM/IEEE International Conference on Automated Software Engineering (ASE'20), Melbourne, Australia, September 2020.

Co-chair · First International Workshop on Smart Contract Analysis (WoSCA), Los Angeles, California, July 2020; co-located with International Symposium on Software Testing and Analysis (ISSTA).

Program committee · 27th International SPIN Symposium on Model Checking of Software (SPIN'20), Chicago, Illinois, July 2020.

Program committee · 20th IEEE International Conference on Software Quality, Reliability, and Security (QRS'20), Vilnius, Lithuania, July 2020.

Program committee · 42nd ACM/IEEE International Conference on Software Engineering 2020 Software Engineering In Practice Track, Seoul, Korea, May 2020.

Program committee · 42nd ACM/IEEE International Conference on Software Engineering 2020 New Ideas and Emerging Results Track, Seoul, Korea, May 2020.

Program committee · 13th IEEE International Conference on Software Testing, Validation, and Verification (ICST'20), Porto, Portugal, March 2020.

Program committee · 34th ACM/IEEE International Conference on Automated Software Engineering (ASE'19) Journal First Presentations Track, San Diego, California, November 2019.

Program committee · 7th International Symposium on Research in Grey-Hat Hacking (GreHack'19), Grenoble, France, November 2019.

Workshops co-chair · ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'19), Beijing, China, July 2019.

Program committee · 41st ACM/IEEE International Conference on Software Engineering (ICSE'19), Montreal, Canada, May 2019.

Program committee · 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE'18), Montpellier, France, September 2018.

Program committee · 40th ACM/IEEE International Conference on Software Engineering (ICSE'18), Gothenburg, Sweden, May 2018.

Program committee · 17th International Conference on Runtime Verification (RV'17), Seattle, Washington, September 2017.

Program committee · ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'17), Santa Barbara, California, July 2017.

Program committee · Doctoral Symposium, ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'17), Santa Barbara, California, July 2017.

Program committee · 2017 Summer Computer Simulation Conference (SCSC'17), Bellevue, Washington, July 2017.

Co-chair · 2nd Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies (CREST'17), Uppsala, Sweden, April 2017; co-located with European Joint Conferences on Theory & Practice of Software (ETAPS).

Steering committee · International Workshop on Dynamic Analysis, 2013-2016.

Program committee · 23rd International SPIN Symposium on Model Checking of Software (SPIN'16), Eindhoven, the Netherlands, April 2016.

Program chair · 9th International Workshop on Constraints in Formal Verification, Austin, Texas, November 2015; co-located with IEEE/ACM International Conference on Computer-Aided Design.

Program committee · ACM SIGDA Student Research Competition, 34th IEEE/ACM International Conference on Computer-Aided Design, November 2015.

Program committee · 22nd International SPIN Symposium on Model Checking of Software (SPIN'15), Stellenbosch, South Africa, August 2015.

Chair · Doctoral symposium at ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'15), Baltimore, Maryland, July 2015.

Workshops selection committee · 37th ACM/IEEE International Conference on Software Engineering (ICSE'15), Firenze, Italy, May 2015.

Reviewing committee · 37th ACM/IEEE International Conference on Software Engineering (ICSE'15), Firenze, Italy, May 2015.

Program committee · 7th NASA Formal Methods Symposium (NFM'15), Pasadena, California, April 2015.

Publicity committee · 8th IEEE International Conference on Software Testing, Verification and Validation (ICST'15), Graz, Austria, April 2015.

Program committee · Research Tool Demonstrations for 22nd ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-DEMO'14), Hong Kong, November 2014.

Program committee · ACM SIGDA Student Research Competition, 33rd IEEE/ACM International Conference on Computer-Aided Design, November 2014.

Program committee · 21st International SPIN Symposium on Model Checking of Software (SPIN'14), San Jose, California, July 2014.

Program committee · ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'14), San Jose, California, July 2014.

Program committee · 28th IEEE/ACM International Conference on Automated Software Engineering (ASE'13), Palo Alto, California, November 2013.

Co-chair · 11th International Workshop on Dynamic Analysis (WODA'13), Houston, Texas, March 2013; co-located with ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS).

Workshops selection committee · 35th ACM/IEEE International Conference on Software Engineering (ICSE'13), San Francisco, California, May 2013.

Co-chair · 18th International Workshop on Model Checking Software (SPIN'11), Snowbird, Utah, July 2011; co-located with International Conference on Computer Aided Verification (CAV).

Program committee · 3rd NASA Formal Methods Symposium (NFM'11), Pasadena, California, April 2011.

Program committee · 14th International Conference on Fundamental Approaches to Software Engineering (FASE'11), Saarbrucken, Germany, March 2011.

Program committee · Workshop on Testing, Analysis, and Verification of Cyber-Physical Systems and Internet of Things (TAV-CPS/IoT'20), Los Angeles, California, July 2020; Java Pathfinder Workshop 2019 (JPF'19), San Diego, California, November 2019; IEEE International Workshop on Debugging and Repair (IDEAR'19), Berlin, Germany, October 2019; 7th IEEE International Workshop on Formal Methods Integration (FMi'19), Los Angeles, California, July-August 2019; Second ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems (FAILSAFE'18), Shenzen, China, November 2018; IEEE International Workshop on Debugging and Repair (IDEAR'18), Memphis, Tennessee, October 2018; Workshop on Testing, Analysis, and Verification of Cyber-Physical Systems and Internet of Things (TAV-CPS/IoT'18), Amsterdam, the Netherlands, July 2018; 6th IEEE International Workshop on Formal Methods Integration (FMI'18), Salt Lake City, Utah, July 2018; International Workshop on Software Fairness, (FairWare'18), Gothenburg, Sweden, May 2018; 3rd Workshop on formal reasoning about Causation, Responsibility, and Explanations in Science and Technology (CREST'18), Thessaloniki, Greece, April 2018; 8th IEEE International Workshop on Program Debugging (IWPD'17), Toulouse, France, October 2017; 5th IEEE International Workshop on Formal Methods Integration, San Diego, California, August 2017; 7th IEEE International Workshop on Program Debugging (IWPD'16), Ottawa, Canada, October 2016; 4th IEEE International Workshop on Formal Methods Integration, Pittsburgh, Pennsylvania, April 2016; 1st Workshop on Causal-based Reasoning for Embedded and Safety-critical Systems Technologies (CREST'16), Eindhoven, The Netherlands, April 2016; Java Pathfinder Workshop 2015 (JPF'15); 6th IEEE Workshop on Program Debugging (IWPD'15), Gaithersburg, Maryland, November 2015; 3rd IEEE International Workshop on Formal Methods Integration, San Francisco, California, August 2015; 7th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE'15), San Francisco, California, July 2015; 5th IEEE Workshop on Program Debugging (IWPD'14), Naples, Italy, November 2014; Java Pathfinder Workshop 2014 (JPF'14), Salt Lake City, Utah, November 2014; 36th Annual IEEE Software Engineering Workshop (SEW-36), Mountain View, California, August 2014; 8th International Workshop on Constraints in Formal Verification (CFV'13), San Jose, California, November 2013; Java Pathfinder Workshop 2012 (JPF'12), Raleigh, North Carolina, November 2012; 35th

Annual IEEE Software Engineering Workshop (SEW-35), Heraclion, Crete, October 2012; 10th International Workshop on Dynamic Analysis (WODA'12), Minneapolis, Minnesota, July 2012; 19th International Workshop on Model Checking Software (SPIN'12), Oxford, England, July 2012; 7th International Workshop on Constraints in Formal Verification (CFV'11), San Jose, California, November 2011; Java Pathfinder Workshop 2011 (JPF'11), Lawrence, Kansas, November 2011; 3rd Workshop on Verification and Validation for Planning and Scheduling Systems (VVPS'11), Freiburg, Germany, June 2011; 34th Annual IEEE Software Engineering Workshop (SEW-34), Limerick, Ireland, June 2011; 9th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'10), Santa Fe, New Mexico, November 2010; 8th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'09), Amsterdam, the Netherlands, August 2009; 6th International Workshop on Constraints in Formal Verification (CFV'09), Grenoble, France, June 2009; 7th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'08), Atlanta, Georgia, November 2008; 14th International Workshop on Model Checking Software (SPIN'07), Berlin, Germany, July 2007; 1st International Workshop on Verification and Debugging (V&D'06), Seattle, Washington, August 2006.

Teaching

Summary: I developed and taught classes covering testing, analysis, software security, and software engineering to graduate and undergraduate students. My teaching evaluations over these classes and during my six semesters as a teaching assistant have been strongly positive. I served on the undergraduate curriculum committee at Oregon State from 2009 until 2015, and (co-)chaired the committee from September 2015-2016.

Spring 2018 · Associate Professor, Northern Arizona University. Taught CS 499, a Special Topics course on software security, covering basic security concepts, protocols and protocol vulnerabilities, famous code vulnerabilities (Heartbleed and goto fail), AFL and other fuzzing tools, taint analysis, static and dynamic approaches to detecting vulnerabilities, and the vulnerability-to-exploit path.

Spring 2017 · Associate Professor, Northern Arizona University. Taught CS 499, a Special Topics course in automated software test generation. Topics included basics of automated software testing, and using actual tools in the field, including TSTL, Csmith, afl-fuzz, and Hypothesis. Class included guest lectures by Linux kernel developers, testing tool creators, and security-based testing experts.

Fall 2016 · Associate Professor, Oregon State University. Taught CS 361, first in software engineering sequence, with emphasis on software architecture, testability, foundations of software engineering as a discipline (readings including Brooks, Parnas, Butler Lampson, and DeMarco and Lister).

Fall, Winter 2013 · Assistant Professor, Oregon State University. Developed ecampus online version of class in Applied Software Engineering, focusing on testing, analysis, code review, debugging, and software maintenance; recorded presentations and selected materials for online learning.

Winter, Spring 2010-2015 · Assistant Professor, Oregon State University. Taught CS 362 and CS 562, undergraduate and graduate classes in Applied Software Engineering. Focused on theory and practice of software implementation, including maintenance, code management, static analysis, testing, model checking, hardware interface and simulation, and debugging. Project-centered courses featured use of an open-source social-networking/project repository system and innovative exchange of programs for testing. Developed upper-level class on software security (basic concepts, protocols, security exploits) and reliability. Developed and recorded online version of class on software testing, analysis, and verification. Taught graduate seminar on Static Analysis and Model Checking (CS 569), with focus on software security and combining static and dynamic analysis.

Fall 2009 · Assistant Professor, Oregon State University. Developed required courses (to be taught Winter and Spring terms) on applied software engineering for undergraduate and graduate students (CS 362 and CS 562), with a focus on design for testability, practical debugging and maintenance, test-driven development, code analysis and instrumentation, and automated testing and verification. Covered test-driven development as guest lecturer in first undergraduate software engineering course (CS 361). Mentored graduate students and initiated a

research program involving graduate and undergraduate students. Served on undergraduate curriculum committee. Helped develop new undergraduate concentration in software engineering for sustainability and energy management.

Spring 2008, Spring 2009 · Lecturer, California Institute of Technology, *CS 119 Reliable Software: Testing and Monitoring* (developed and taught with Klaus Havelund), third term 2007-2008 and 2008-2009. Topics included random testing, constraint-based testing, coverage measures, design for testability, static analysis, test-driven development, automated debugging, and the use of model checkers. Focused on practical application (and limits) of state-of-the-art methods.

Spring 2007 · External Master's thesis examiner, Stellenbosch University

Fall 2003 · Teaching assistant, Carnegie Mellon University, for undergraduate course 15-212, *Principles of Programming* (introduction to programming in Standard ML, including type discipline and proof by induction): Formulated assignments, graded assignments and tests, taught a weekly recitation session, and held office hours.

Spring 2000 · Teaching assistant, Carnegie Mellon University, for undergraduate course 15-312, *Foundations of Programming Languages* (advanced type theory, continuations, and concurrency): Formulated tests and assignments, graded assignments and tests, lectured in absence of Professor Harper, taught a weekly recitation session, and held office hours.

Fall 1998 · Teaching assistant, North Carolina State University, for undergraduate course CSC417, *Theory of Programming Languages* (type theory and functional programming in ML): Graded assignments and held weekly office hours.

Fall 1997, Spring 1998 · Teaching assistant, North Carolina State University, for undergraduate course CSC210, *Programming Concepts* (second-level introductory course in C++, including pointers, recursion, and fundamental data structures): Graded assignments, held weekly office hours, and provided on-the-spot teaching and assistance to students in computer labs.

Spring 1997 · Teaching assistant, North Carolina State University, CSC495C, *Special Topics* (class for professional C programmers learning C++ and object-oriented design): Graded assignments and held weekly office hours.

Current Students

Michael Slater, Oregon State University, PhD, Committee Member

Alex Wiggins, Oregon State University, PhD, Committee Member

Austin Sanders, Northern Arizona University, PhD, Committee Member

Advised Students Completing PhD in Computer Science

Arpit Christi, Oregon State University, PhD, Committee Chair (PhD received 2019)

Rahul Gopinath, Oregon State University, PhD, Committee Co-Chair (PhD received 2017)

Mohammad Amin Alipour, Oregon State University, PhD, Committee Chair (PhD received 2017)

Ali Aburas, Oregon State University, PhD, Committee Chair (PhD received 2016)

Chaoqiang Zhang, Oregon State University, PhD, Committee Chair (PhD received 2015)

Other Graduated Students

Xin Liu, Oregon State University, MS, Committee Chair

Kazuki Kaneoka, Oregon State University, MS, Committee Chair

Shalini Shamasunder, Oregon State University, MS, Committee Chair

Gokul Caushik, Oregon State University, MEng, Major Advisor

Aravind Palanisami, Oregon State University, MEng, Major Advisor

Pengfei Chen, Oregon State University, MEng, Major Advisor

Iftekhar Ahmed, Oregon State University, PhD, Committee Member

Soroush Ghorashi, Oregon State University, PhD, Committee Member

Jervis Pinto, Oregon State University, PhD, Committee Member

Todd Kulesza, Oregon State University, PhD, Committee Member

Chris Chambers, Oregon State University, PhD, Committee Member

Yang Chen, University of Utah, PhD, External Committee Member

Christopher Bogart, Oregon State University, PhD, Committee Member

Eric Walkingshaw, Oregon State University, PhD, Committee Member

Duc Le, Oregon State University, MS, Committee Member

Darren Forrest, Oregon State University, MS, Committee Member

Prashanth Ayyavu, Oregon State University, MS, Committee Member

Nishanthini Narayanan, Oregon State University, MS, Committee Member

Nitin Mohan, Oregon State University, MS, Committee Member

David Burri, Oregon State University, MS, Committee Member

Madhura Vadvalkar, Oregon State University, MEng, Committee Member

Michael Tichenor, Oregon State University, MS, Committee Member

Alex Diede, Oregon State University, MEng, Committee Member

Invited Seminars

Invited speaker, 7th Halmstad Summer School on Testing, Halmstad, Sweden, June 12-15, 2017.

Dagstuhl Seminar 03491, Understanding Program Dynamics, Schloss Dagstuhl, Wadern, Germany, November 31-December 5, 2003.

Invited Talks and Panels

"TSTL: a Little (Integrated) Language for Testing", 7th Halmstad Summer School on Testing, Halmstad, Sweden, June 12, 2017.

"Presenting TSTL (and the Quest for One Test to Rule them All)," School of Informatics, Computing, and Cybersystems, Northern Arizona University, Flagstaff, AZ, January 22, 2016.

"Understanding and Exploiting Triggers and Suppressors in Testing," Galois, Inc., Portland, OR, March 3, 2014.

"Making the Most of Random Tests," Google Inc., Mountain View, CA, February 24, 2014.

"Effective Random Testing for Critical Systems Software," Arizona State University, Phoenix, AZ, January 13, 2014.

Panelist, "Program Debugging: Transitioning from Research to Practice," International Workshop on Program Debugging, Pasadena, CA, November 4, 2013.

"Learning-Based Test Programming for Programmers," International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, Heraclion, Crete, October 16, 2012.

"For Truly Thorough Testing, You Have to Leave Things Out," Northern Arizona University, Flagstaff, AZ, April 11, 2012.

"Traces in Spaces: You Can Learn a Lot About a Program by Running It," School of Electrical Engineering and Computer Science Colloquium Series, Oregon State University, Corvallis, OR, February 2, 2009.

"Putting Flight Software Through the Paces with Testing, Model Checking, and Constraint-Solving," International Workshop on Constraints in Formal Verification / International Verification Workshop, Sydney, Australia, August 11, 2008.

"Asking the Right Questions — and Understanding the Answers — in Software Testing," (with Klaus Havelund), Information Science and Technology Lunch Bunch, California Institute of Technology, Pasadena, CA, February 19, 2008.

"How to Break a (Flash) File System," Jet Propulsion Laboratory-Goddard Space Flight Center (JPL-GSFC) Quality Mission Software Workshop, Santa Barbara, CA, May 2, 2006.

"Exploiting Traces in Program Analysis," Workshop on Theories, Methods and Tools for Building Systems from Interacting Components, California Institute of Technology, Pasadena, CA, October 31, 2005.

"Explaining Counterexamples," IBM T. J. Watson Research Center, Hawthorne, NY, December 20, 2004.

"Explaining Counterexamples," Microsoft Research, Redmond, WA, November 8, 2004. Similar version presented as Speakers' Club seminar at Carnegie Mellon University, Pittsburgh, PA, December 9, 2004.

"Debugging Code with Model Checkers," Jet Propulsion Laboratory, Pasadena, CA, November 1, 2004.

"Error Explanation via Model Checking," Dagstuhl Seminar 03491, Understanding Program Dynamics, Schloss Dagstuhl, Wadern, Germany, December 5, 2003.

Selected Presentations

"Who Watches the Watchmen? Using Mutation Testing to Sanity-Check Smart Contract Testing and Analysis," College of Engineering, Informatics, and Applied Sciences Seminar Series, Northern Arizona University, Flagstaff, AZ, March 29, 2019.

"Causal Distance-Metric-Based Assistance for Debugging After Compiler Fuzzing," IEEE International Symposium on Software Reliability Engineering, Memphis, TN, October 18, 2018.

"DeepState: Symbolic Unit Testing for C and C++," Seminar on Softwre Testing, Northern Arizona University, Flagstaff, AZ, September 28, 2018.

"Towards Automated Composition of Heterogeneous Tests for Cyber-Physical Systems," Workshop on Testing Embedded and Cyber-Physical Systems, Santa Barbara, CA, July 13, 2017.

"A Suite of Tools for Making Effective Use of Automatically Generated Tests," International Symposium on Software Testing and Analysis, Santa Barbara, CA, July 10, 2017.

"One Test to Rule Them All," International Symposium on Software Testing and Analysis, Santa Barbara, CA, July 10, 2017.

"Mitigating (and Exploiting) Test Reduction Slippage," Workshop on Automated Software Testing, Seattle, WA, November 18, 2016.

"How Verified is My Code? Falsification-Driven Verification," IEEE/ACM International Conference on Automated Software Engineering, Lincoln, NE, November 13, 2015.

"A Little Language for Testing," NASA Formal Methods Symposium, Pasadena, CA, April 28, 2015.

"Coverage and Its Discontents," Onward! Essays, part of SPLASH (ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity), Portland, OR, October 24, 2014.

"Help! Help! I'm Being Suppressed! The Significance of Suppressors in Software Testing," IEEE International Symposium on Software Reliability Engineering, Pasadena, CA, November 6, 2013.

"New Directions in Random Testing: From Mars Rovers to JavaScript Engines," Galois, Inc. Tech Talk, Portland, OR, September 12, 2013.

"Beyond the Kitchen Sink: Swarm Testing", Jet Propulsion Laboratory, Pasadena, CA, May 7, 2013.

"Extended Program Invariants: Applications in Testing and Fault Localization," International Workshop on Dynamic Analysis, Minneapolis, MN, July 15, 2012.

"Finding Common Ground: Choose, Assert, and Assume," International Workshop on Dynamic Analysis, Minneapolis, MN, July 15, 2012.

"Coverage Rewarded: Test Input Generation via Adaptation-Based Programming," IEEE/ACM International Conference on Automated Software Engineering, Lawrence, KS, November 9, 2011.

"Establishing Appropriate User Trust in Machine-Learned Classifiers," Human/Machine Learning Partnerships, Oregon State University, Corvallis, OR, May 21, 2010.

"Can End Users Test Machine-Learning Classifiers?," End Users and Machine Learning Day, Oregon State University, Corvallis, OR, February 26, 2010.

"Path Coverage and Its Discontents," School of Electrical Engineering and Computer Science Colloquium Series, Oregon State University, Corvallis, OR, February 22, 2010.

"(Quickly) Testing the Tester via Path Coverage," International Workshop on Dynamic Analysis, Chicago, IL, July 20, 2009.

"Advanced Testing Tools," (with Klaus Havelund), Engineering and Science Directorate - Software Engineering Process Group, Jet Propulsion Laboratory, Pasadena CA, April 30, 2009.

"Random Testing and Model Checking: Building a Common Framework for Nondeterministic Exploration," International Workshop on Dynamic Analysis, Seattle, WA, July 21, 2008.

"Model-Driven Software Verification Methods," LaRS Advisory Committee Meeting, Jet Propulsion Laboratory, Pasadena, CA, June 26, 2008.

"Model Checking, Dynamic Analysis, and Unsound Abstractions," Southern California Workshop on Programming Languages and Systems, Claremont, CA, February 2, 2008.

"Extending Model Checking with Dynamic Analysis," Verification, Model Checking and Abstract Interpretation, San Francisco, CA, January 8, 2008.

"Model-Driven Verification," Mission Computing and Autonomy Systems Research Program (982) FY07 Year End Review, Jet Propulsion Laboratory, Pasadena, CA, October 3, 2007.

"Testing the Kepler Flash File System," LaRS Advisory Committee Meeting, Jet Propulsion Laboratory, Pasadena, CA, July 27, 2007.

"Randomized Differential Testing as a Prelude to Formal Verification," ACM/IEEE International Conference on Software Engineering, Minneapolis, MN, May 24, 2007.

"Strengthening Software Testing," LaRS Advisory Committee Meeting, Jet Propulsion Laboratory, Pasadena, CA, July 26, 2006. Similar version presented as Section 316 Brown Bag Lecture at JPL on August 23, 2006.

"LaRS File System Test Approach," Flight Software Applications and Data Management (316D) Group Meeting, Jet Propulsion Laboratory, Pasadena, CA, May 16, 2006.

"Exploiting Traces in Program Analysis," International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Vienna, Austria, March 29, 2006.

"Bounded Model Checking Explained," LaRS Seminar, Jet Propulsion Laboratory, Pasadena, CA, June 14, 2005.

"Error Explanation and Fault Localization with Distance Metrics," Thesis Oral, Carnegie Mellon University, Pittsburgh, PA, March 3, 2005.

"Counterexample Guided Abstraction Refinement via Program Execution," International Conference on Formal Engineering Methods, Seattle, WA, November 11, 2004.

"Explaining Abstract Counterexamples,"ACM SIGSOFT International Symposium on the Foundations of Software Engineering, Newport Beach, CA, November 2, 2004.

"CBMC and C Model Checking," MURI (Multidisciplinary University Research Initiative) Review Meeting, Annapolis, MD, August 16, 2004.

"Java PathFinder," Software Model Checking Seminar, Carnegie Mellon University, Pittsburgh, PA, July 22, 2004.

"Error Explanation with Distance Metrics," International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Barcelona, Spain, March 29, 2004.

"Explaining Errors," MURI (Multidisciplinary University Research Initiative) Workshop, Carnegie Mellon University, Pittsburgh, PA, July 22, 2003.

"Explaining Counterexamples: Causal Analysis and Comparison of Transition Sequences," Specification and Verification Center, Carnegie Mellon University, Pittsburgh, PA, May 20, 2003.

"What Went Wrong: Explaining Counterexamples," SPIN Workshop on Model Checking of Software, Portland, OR, May 9, 2003. Earlier versions presented at Specification and Verification Center, Carnegie Mellon University, Pittsburgh, PA, September 17, 2002, and NASA Ames Research Center/RIACS Seminar, Mountain View, CA, August 8, 2002.

"Model Checking Java Programs using Structural Heuristics," International Symposium on Software Testing and Analysis, Rome, Italy, July 22, 2002.

"Heuristic Model Checking for Java Programs," SPIN Workshop on Model Checking of Software, Grenoble, France, April 13, 2002.

"Adaptive Model Checking," International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Grenoble, France, April 11, 2002.

"Structural Heuristics for Directed Model Checking of Java Programs," Specification and Verification Center, Carnegie Mellon University, Pittsburgh PA, March 19, 2002.

"Efficient Model Checking Via Büchi Tableau Automata," International Conference on Computer Aided Verification, Paris, France, July 19, 2001.

"Black Box Checking," Federal University of Rio Grande do Norte, Natal, Brazil, March 29, 2001.

References

Edmund M. Clarke, Jr.
FORE Systems Professor of Computer Science
Professor of Electrical and Computer Engineering
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
412-268-2628
emc+@cs.cmu.edu

Gerard J. Holzmann
JPL Fellow and Senior Research Scientist
Laboratory for Reliable Software
Jet Propulsion Laboratory
4800 Oak Grove Drive
M/S 301-230
Pasadena, CA 91109
818-393-5937
Gerard.J.Holzmann@jpl.nasa.gov

Willem Visser
Professor of Computer Science
Head, Computer Science Division
University of Stellenbosch
Private Bag X1
7602 Matieland
South Africa
+27 21 808 4235
willem@gmail.com