

Alex David Groce

Business Address:

School of Electrical Engineering and Computer Science
Oregon State University
1148 Kelley Engineering Center
Corvallis, OR 97331-5501

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 ten years of experience in software testing and analysis research: formulating problems, proposing algorithms and approaches, evaluating results, applying my work to critical distributed, embedded, systems, aerospace, and security software and software/hardware systems, and presenting concepts to researchers and students. I contributed significantly to the implementation and design of verification and analysis tools used internationally for research, industrial application, and teaching, including Java PathFinder 2, CBMC, MAGIC, SyMP, JPL's LogScope, and the Concurrency Workbench (NC).

6/2009–present · 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 classifiers. Continued work on unified approaches to testing, with new techniques applied to finding flaws in widely used production-quality C compilers and embedded file systems. Mentored graduate students, taught graduate and undergraduate classes on software engineering and verification, and served on undergraduate curriculum committee and hiring committee.

2/2011–present · 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 mission 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 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.

Teaching

- Summary:** I developed and taught classes covering testing, analysis, 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.
- Winter, Spring 2010, 2011, 2012 · 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.
- 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

Chaoqiang Zhang, Oregon State University, PhD, Committee Chair

Amin Alipour, Oregon State University, PhD, Committee Chair

Shalini Shamasunder, Oregon State University, MS, Committee Chair

Christopher Bogart, Oregon State University, PhD, Committee Member

Sheng Chen, Oregon State University, PhD, Committee Member

Todd Kulesza, Oregon State University, PhD, Committee Member

Duc Le, Oregon State University, PhD, Committee Member

Rahul Gopinath, Oregon State University, PhD, Committee Member

Chris Chambers, Oregon State University, PhD, Committee Member

Darren Forrest, Oregon State University, PhD, Committee Member

Past Students

Aravind Palanisami, Oregon State University, MEng, Major Advisor

Prashanth Ayyavu, Oregon State University, MS, Committee Member

Nitin Mohan, Oregon State University, MS, Committee Member

Madhura Vadvalkar, Oregon State University, MEng, Committee Member

Research Interests

Designing, specifying, coding, testing, verifying, understanding, and debugging complex computer systems. My focus has largely been on real world systems, where testing is useful and productive. My research has combined testing, static analysis, formal methods, programming languages, and machine learning approaches as required. 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.

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.

Error explanation: I developed automatic tools based on model checking for *explaining* and *localizing* errors in systems. Understanding and correcting errors can be as difficult and costly as implementation or design, and better methods and algorithms are critical to improving systems reliability.

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

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

ICSE 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

Professional Activities

Reviewer for *IEEE Transactions on Software Engineering* (TSE), *ACM Transactions on Software Engineering and Methodology* (TOSEM), *Journal of the ACM* (JACM), *Microprocessors and Microsystems: Embedded Hardware Design*, *Software Tools for Technology Transfer* (STTT), *Formal Methods in System Design* (FMSD), *IEEE Transactions on Parallel and Distributed Systems* (TPDS), *IEEE Transactions on Reliability*, *Journal on Satisfiability*, *Boolean Modeling and Computation* (JSAT), *Journal of Logic and Computation*, *Journal of Computer and System Sciences* (JCSS), *Software Testing, Verification and Reliability*, *Automated Software Engineering Journal*, *Annals of Mathematics and Artificial Intelligence* (AMAI), *Journal of Applied Logic* (JAL),

Journal of Computer Science and Technology (JCST), Journal of Systems and Software (JSS), 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), Verification, Model Checking, and Abstract Interpretation (VMCAI), International Conference on Software Testing, Verification and Validation (ICST), Symposium on Automated and Analysis-driven Debugging (AADEBUB), 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), 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), and Workshop on Formal Methods for Industrial Critical Systems (FMICS).

Association for Computing Machinery, Special Interest Group on Software Engineering (ACM SIGSOFT), Special Interest Group on Programming Languages (SIGPLAN) member.

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

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

Program committee · 35th Annual IEEE Software Engineering Workshop (SEW-35), Heraclion, Crete, October 2012.

Program committee · 10th International Workshop on Dynamic Analysis (WODA'12), Minneapolis, Minnesota, July 2012.

Program committee · 19th International Workshop on Model Checking Software (SPIN'12), Oxford, England, July 2012.

Program committee · 7th International Workshop on Constraints in Formal Verification (CFV'11), San Jose, California, November 2011.

Program committee · Java Pathfinder Workshop 2011 (JPF'11), Lawrence, Kansas, November 2011.

Chair · 18th International Workshop on Model Checking Software (SPIN'11), Snowbird, Utah, July 2011.

Program committee · 3rd Workshop on Verification and Validation for Planning and Scheduling Systems (VVPS'11), Freiburg, Germany, June 2011.

Program committee · 34th Annual IEEE Software Engineering Workshop (SEW-34), Limerick, Ireland, June 2011.

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), Saarbrücken, Germany, March 2011.

Program committee · 9th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'10), Santa Fe, New Mexico, November 2010.

Program committee · 8th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'09), Amsterdam, the Netherlands, August 2009.

Program committee · 6th International Workshop on Constraints in Formal Verification (CFV'09), Grenoble, France, June 2009.

Program committee · 7th Workshop on Specification and Verification of Component-Based Systems (SAVCBS'08), Atlanta, Georgia, November 2008.

Program committee · 14th International Workshop on Model Checking Software (SPIN'07), Berlin, Germany, July 2007.

Program committee · 1st International Workshop on Verification and Debugging (V&D'06), Seattle, Washington, August 2006.

Funding

“CAREER: Integrating Automated Software Testing Methods”, PI: Alex Groce, *National Science Foundation* CCF-1054786, \$400K, September 2011-2016.

Books, Edited Volumes

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

Refereed Journal Publications

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*, accepted for publication.

Gerard Holzmann, Rajeev Joshi, and **Alex Groce**. Swarm Verification Techniques. *IEEE Transactions on Software Engineering*, PP Issue:99, December 2010.

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.

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 Martin Erwig. Finding Common Ground: Choose, Assert, and Assume. *Workshop on Dynamic Analysis*, 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*, to appear, 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%).

Amin Alipour and **Alex Groce**. Bounded Model Checking and Feature Omission Diversity. *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. *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*, Patras, Greece, July 2008.

Alex Groce and Rajeev Joshi. Random Testing and Model Checking: Building a Common Framework for Nondeterministic Exploration. *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%).
- 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%).
- 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*, 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 [**ICSE SIGSOFT Distinguished Paper Award**] (acceptance rate 13%).
- 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

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*, to appear in *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**. Model Driven Code Checking. *Automated Software Engineering, Special Issue on Trends in Automated Software Engineering*, 15(3-4):283-197, December 2008.

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.

Technical Reports

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.

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>).

Invited Seminars

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

Invited Talks

“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

“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,” 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,” 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

May, 2012