

Alex David Groce

Business Address:

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Mail Stop 301-285
agroce@gmail.com

Home Address:

975 San Pasqual St., Apt. 223
Pasadena, CA 91106
626-616-9229

US Citizen

Education

Ph.D., **Computer Science**, Carnegie Mellon University, March 2005
Thesis Title: **Error Explanation and Fault Localization with Distance Metrics**.
Advisor: Edmund Clarke

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 nearly ten years of experience in software engineering and verification research: formulating problems, proposing algorithms and approaches, evaluating results, applying my work to critical embedded, aerospace, and real-time operating-system software, and presenting concepts to researchers and students. I contributed significantly to the implementation and design of software analysis tools used internationally for research, industrial application, and teaching, including Java PathFinder 2, CBMC, MAGIC, SyMP, and the Concurrency Workbench (NC).

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

4/2005–present · Laboratory for Reliable Software, Jet Propulsion Laboratory: Led test automation development and design, Mars Science Laboratory mission Flight Software Internal Test Team. Contributed to design of telemetry systems, flight software and test system interfaces to VxWorks, and fault protection systems. Tested hardware/software interfaces for Mars Science Laboratory mission. Introduced new techniques for exploiting error traces in static analysis of programs, integrated model checking and dynamic analysis, developed a successful random testing approach for reliable 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 and hardware systems.

8/1999–3/2005 · Doctoral student, Carnegie Mellon University: Invented methods for error explanation and fault isolation using distance metrics, applied to aircraft collision avoidance system and real-time operating system microkernel. Invented and implemented novel approaches for counterexample guided abstraction refinement. 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 Java programs, applied to Honeywell’s DEOS real-time operating system and an autopilot system.

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, applied to DEOS real-time operating system.

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 have six semesters of experience as a teaching assistant for university-level courses, and developed and taught a class on software testing for undergraduates. My teaching evaluations have been strongly positive.

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

Spring 2007 · External Master’s examiner, Stellenbosch University

Fall 2003 · Teaching assistant, Carnegie Mellon University, for undergraduate course 15-212, *Principles of Programming* (introduction to programming in SML, 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.

Research Interests

Software engineering: designing, specifying, *coding*, *testing*, *verifying*, analyzing, and *debugging* software written in widely used languages, such as C, C++, and Java, with a focus on critical embedded and systems software for aerospace applications. 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 engineering* (and thus more reliable systems) and better software engineering education.

Topics

Easing Communication in Aerospace Systems Development: Poor communication of requirements and expected behaviors between hardware designers, embedded software engineers, and aerospace or systems engineers often troubles the design and testing of complex aerospace systems. Preliminary work with the Mars Science Laboratory mission at JPL suggests that stating system requirements as event patterns may aid engineers in understanding and help automate the testing of hardware/software interfaces.

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.

Software model checking: I continue to investigate the use of SAT 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 for *explaining* and *localizing* errors in C and Java source code. Understanding and correcting errors can be as difficult and costly as implementation or design, and better methods and algorithms are critical to an improved software development experience.

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

Honors

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), *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*, *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), Symposium on Automated and Analysis-driven Debugging (AADEBUG), 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), 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), and Workshop on Software Model Checking (SoftMC).

Association for Computing Machinery, Special Interest Group on Software Engineering (ACM SIGSOFT) 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 · 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.

Journal Publications

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

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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**, Daniel Kroening, and Flavio Lerda. Understanding Counterexamples with `explain`. *International Conference on Computer Aided Verification*, pages 453–456, 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.
- 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**, 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.
- Alex Groce** and Willem Visser. What Went Wrong: Explaining Counterexamples. *SPIN Workshop on Model Checking of Software*, pages 121–135, Portland, Oregon, May 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**].

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.

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.

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.

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.

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.

Invited Papers

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

Nicolas Blanc, Daniel Kroening, and **Alex Groce**. Verifying C++ with STL Containers via Predicate Abstraction. Technical Reports 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

“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.

Selected Presentations

“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.

“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.

“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.

“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,” International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Vienna, Austria, March 29, 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.

“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.

- “Explaining Counterexamples,” IBM T. J. Watson Research Center, Hawthorne, NY, December 20, 2004.
- “Counterexample Guided Abstraction Refinement via Program Execution,” International Conference on Formal Engineering Methods, Seattle, WA, November 11, 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.
- “Explaining Abstract Counterexamples,” ACM SIGSOFT International Symposium on the Foundations of Software Engineering, Newport Beach, CA, November 2, 2004.
- “Debugging Code with Model Checkers,” Jet Propulsion Laboratory, Pasadena, CA, November 1, 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.
- “Error Explanation via Model Checking,” Dagstuhl Seminar 03491, Understanding Program Dynamics, Schloss Dagstuhl, Wadern, Germany, December 5, 2003.
- “Explaining Errors,” MURI 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
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
412-268-2628
emc+@cs.cmu.edu

Gerard J. Holzmann
4800 Oak Grove Drive
M/S 301-230
Pasadena, CA 91109
818-393-5937
Gerard.J.Holzmann@jpl.nasa.gov

Willem Visser
SEVEN Networks
Redwood City, CA 94063
408-971-1512
willem@gmail.com

March, 2018