

## Jonathan Aldrich

Home: 5426 Beacon St.  
Pittsburgh, PA 15217  
(412) 521-1918

Office: Institute for Software Research International  
School of Computer Science  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
(412) 268-7278

Email: [jonathan.aldrich@cs.cmu.edu](mailto:jonathan.aldrich@cs.cmu.edu)  
WWW: <http://www.cs.cmu.edu/~aldrich/>  
Citizenship: United States

### Research Interests

Lightweight software verification using programming language and program analysis techniques.

### Positions

2003-present. Assistant Professor, Carnegie Mellon University.  
1997-2003. Graduate Student and Research Assistant, University of Washington.  
Summer 1997. Research Assistant, California Institute of Technology.  
Summers 1992-1996. Summer projects at Sequent Computer Systems, Inc.

### Education

Ph.D., Computer Science, University of Washington, August 2003.  
Advisors: Craig Chambers and David Notkin  
Thesis: Using Types to Enforce Architectural Design  
M.S., Computer Science, University of Washington, June 1999.  
B.S., Engineering and Applied Science (Computer Science), California Institute of Technology, June 1997.

### Selected Honors

2006 National Science Foundation CAREER award, "Lightweight Modeling and Enforcement of Architectural Behavior"  
2003 William Chan Memorial Dissertation Award, University of Washington, Dept. Comp. Sci. and Eng.  
1997-2000 National Defense Science and Engineering Graduate Fellowship  
1997-2000 Achievement Rewards for College Scientists Fellowship  
1997 National Science Foundation Fellowship Honorable Mention  
1995-1997 Caltech Merit Scholarship  
1993 National Merit Scholarship  
1996 Winner, Caltech-Occidental Symphony Concerto Competition (violin)  
Honor societies: Sigma Xi (scientific research), Tau Beta Pi (engineering)

### Book Chapters

Donna Malayeri and Jonathan Aldrich. Practical Exception Specifications. In Christophe Dony, Jørgen Lindskov Knudsen, Alexander B. Romanovsky, and Anand Tripathi, editors, *Advanced Topics in Exception Handling Techniques*, volume 4119 of Lecture Notes in Computer Science, pages 200–220. Springer, 2006. Originally published in the *ECOOP 2005 Workshop on Exception Handling in Object-Oriented Systems*, July 2005. <http://www.cs.cmu.edu/~donna/public/malayeri.exceptions.06.pdf>

## Refereed Journal and Conference Publications

Marwan Abi-Antoun, Jonathan Aldrich, Nagi Nahas, Bradley Schmerl, and David Garlan. Differencing and Merging of Architectural Views. Extended version of the ASE '06 paper, invited to the *Automated Software Engineering Journal*, submitted January 2007. Under review.

Marwan Abi-Antoun, Jonathan Aldrich, and Wesley Coelho. A Case Study in Re-engineering to Enforce Architectural Control Flow and Data Sharing. In *Journal of Systems and Software*, 80(2), pp. 240–264, 2007.

Bradley Schmerl, Jonathan Aldrich, David Garlan, Rick Kazman, and Hong Yan. Discovering Architectures from Running Systems. In *IEEE Transactions on Software Engineering*, 32(7):454-466, July 2006.

Marwan Abi-Antoun, Jonathan Aldrich, Nagi Nahas, Bradley Schmerl, and David Garlan. Differencing and Merging of Architectural Views. In Proceedings of *Automated Software Engineering (ASE '06)*, September 2006. An extended version is available as Carnegie Mellon University Technical Report CMU-ISRI-05-128, August 2005. <http://www.cs.cmu.edu/~aldrich/papers/ase06.pdf>

Kevin Bierhoff and Jonathan Aldrich. Lightweight Object Specification with Typestates. In Proceedings of *Foundations of Software Engineering (FSE '05)*, September 2005. <http://www.cs.cmu.edu/~aldrich/papers/fse05.pdf>

Neel Krishnaswami and Jonathan Aldrich. Permission-Based Ownership: Encapsulating State in Higher-Order Typed Languages. In Proceedings of *Programming Language Design and Implementation (PLDI '05)*, June 2005. <http://www.cs.cmu.edu/~aldrich/papers/pldi05.pdf>

Jonathan Aldrich. Open Modules: Modular Reasoning about Advice. In Proceedings of the *European Conference on Object-Oriented Programming (ECOOP '05)*, July 2005. An earlier version appeared in *Foundations of Aspect Languages*, March 2004. <http://www.cs.cmu.edu/~aldrich/papers/ecoop05open-modules.pdf>

Lee Salzman and Jonathan Aldrich. Prototypes with Multiple Dispatch: An Expressive and Dynamic Object Model. In Proceedings of the *European Conference on Object-Oriented Programming (ECOOP '05)*, July 2005. <http://www.cs.cmu.edu/~aldrich/papers/ecoop05pmd.pdf>

Marwan Abi-Antoun, Jonathan Aldrich, David Garlan, Bradley Schmerl, Nagi Nahas, and Tony Tseng. Modeling and Implementing Software Architecture with Acme and ArchJava. Demonstration, in proceedings of the *International Conference on Software Engineering (ICSE '05)*, May 2005. <http://www.cs.cmu.edu/~aldrich/papers/icse05-demo.pdf>

Jonathan Aldrich and Craig Chambers. Ownership Domains: Separating Aliasing Policy from Mechanism. In Proceedings of the *European Conference on Object-Oriented Programming (ECOOP '04)*, June 2004. <http://www.cs.cmu.edu/~aldrich/papers/ecoop04.pdf>

Hong Yan, David Garlan, Bradley Schmerl, Jonathan Aldrich, and Rick Kazman. DiscoTect: A System for Discovering Architectures from Running Systems. In Proceedings of *International Conference on Software Engineering (ICSE '04)*, May 2004. <http://www.cs.cmu.edu/~aldrich/papers/icse04.pdf>

Jonathan Aldrich, Vibha Sazawal, Craig Chambers, and David Notkin. Language Support for Connector Abstractions. In Proceedings of the *European Conference on Object-Oriented Programming (ECOOP '03)*, July 2003. <http://archjava.fluid.cs.cmu.edu/papers/ecoop03.pdf>

Jonathan Aldrich, Valentin Kostadinov, and Craig Chambers. Alias Annotations for Program Understanding. In Proceedings of *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '02)*, November 2002. <http://archjava.fluid.cs.cmu.edu/papers/oopsla02.pdf>

Jonathan Aldrich, Craig Chambers, and David Notkin. Architectural Reasoning in ArchJava. In Proceedings of the *European Conference on Object-Oriented Programming (ECOOP '02)*, June 2002. <http://archjava.fluid.cs.cmu.edu/papers/ecoop02.pdf>

Jonathan Aldrich, Craig Chambers, and David Notkin. ArchJava: Connecting Software Architecture to Implementation. In Proceedings of the *International Conference on Software Engineering (ICSE '02)*, May 2002. <http://archjava.fluid.cs.cmu.edu/papers/icse02.pdf>

Jonathan Aldrich, Emin Gun Sirer, Craig Chambers, and Susan Eggers. Comprehensive Synchronization Elimination for Java. *Science of Computer Programming*, 47(2-3):91-120, May-June 2003. <http://www.cs.cmu.edu/~aldrich/papers/scp-camera.pdf>

Jonathan Aldrich, Craig Chambers, Emin Gun Sirer, and Susan Eggers. Static Analyses for Eliminating Unnecessary Synchronization from Java Programs. In Proceedings of the *Sixth International Static Analysis Symposium (SAS '99)*, September 1999. <http://www.cs.cmu.edu/~aldrich/papers/sas99.pdf>

Jonathan Aldrich, James Dooley, Scott Mandelsohn, and Adam Rifkin. Providing Easier Access to Remote Objects in Client Server Systems. In *Thirty-first Hawaii International Conference on System Sciences (HICSS-31)*, January 1998. <http://www.cs.cmu.edu/~aldrich/papers/hicss31.pdf>

## Workshop Papers, Demonstrations, and Technical Reports

Nels Beckman and Jonathan Aldrich. A Programming Model for Failure-Prone, Collaborative Robots. To appear in the 2nd International Workshop on Software Development and Integration in Robotics (SDIR). Rome, Italy. April 2007.

Donna Malayeri and Jonathan Aldrich. Combining structural subtyping and external dispatch. In *2007 International Workshop on Foundations and Development of Object-Oriented Languages (FOOL/WOOD'07)*, Nice, France, January 2007. An expanded version is available as Carnegie Mellon Technical Report CMU-CS-06-178. <http://www.cs.cmu.edu/~donna/public/malayeri.fool07.pdf>

Marwan Abi-Antoun and Jonathan Aldrich. JavaD: Bringing Ownership Domains to Mainstream Java. Carnegie Mellon University Technical Report CMU-ISRI-06-110, May 2006.

Kevin Bierhoff, Jonathan Aldrich, and Sangjin Han. A Language-based Approach to Specification and Enforcement of Architectural Protocols (Expanded Version). Technical Report CMU-CS-06-119, April 2006. <http://www.cs.cmu.edu/~aldrich/papers/CMU-CS-06-119.pdf>

Andi Bejleri, Jonathan Aldrich, and Kevin Bierhoff. Ego: Controlling the Power of Simplicity. In proceedings of the POPL '06 Workshop on Foundations of Object-Oriented Languages (*FOOL '06*), January 2006. An expanded version with complete proofs is published as Carnegie Mellon Technical Report CMU-ISRI-04-142, December 2004. <http://www.cs.cmu.edu/~aldrich/papers/ego.pdf>

Marwan Abi-Antoun, Jonathan Aldrich, David Garlan, Bradley Schmerl, Nagi Nahas, and Tony Tseng. Improving System Dependability by Enforcing Architectural Intent. In proceedings of the *ICSE 2005 Workshop on Architecting Dependable Systems (WADS '05)*, May 2005. <http://www.cs.cmu.edu/~aldrich/papers/wads05.pdf>

Jonathan Aldrich and Kevin Donnelly. Selective Open Recursion: Modular Reasoning about Components and Inheritance. In proceedings of the *FSE 2004 Workshop on Specification and Verification of Component-Based Systems (SAVCBS '04)*, November 2004. <http://www.cs.cmu.edu/~aldrich/papers/savcbs04.pdf>

Jonathan Aldrich. Open Modules: Reconciling Extensibility and Modularity. In Proceedings of the *Workshop on Software Engineering Properties of Languages for Aspect Technologies (SPLAT '04)*, March 2004. <http://www.cs.cmu.edu/~aldrich/papers/splat04.pdf>

Neel Krishnaswami and Jonathan Aldrich. Statically-Scoped Exceptions: a Typed Foundation for Aspect-Oriented Error Handling. Carnegie Mellon Technical Report CMU-ISRI-05-102. Published on the web January 2004, as a tech report January 2005. <http://www.cs.cmu.edu/~aldrich/papers/CMU-ISRI-05-102.pdf>

Jonathan Aldrich, Vibha Sazawal, Craig Chambers, and David Notkin. Architecture-Centric Programming for Adaptive Systems. In Proceedings of the *Workshop on Self-Healing Systems (WOSS '02)*, November 2002. <http://archjava.fluid.cs.cmu.edu/papers/woss02.pdf>

Vibha Sazawal and Jonathan Aldrich. Architecture-Centric Programming for Context-Aware Configuration. In Proceedings of the *OOPSLA '02 Workshop on Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE '02)*, November 2002.

<http://archjava.fluid.cs.cmu.edu/papers/oopsla02-ecoose.pdf>

Jonathan Aldrich. Challenge Problems for Separation of Concerns. In Proceedings of the *OOPSLA 2000 Workshop on Advanced Separation of Concerns*, October 2000.

<http://www.cs.cmu.edu/~aldrich/papers/challenge.pdf>

Jonathan Aldrich. Evaluating Module Systems for Crosscutting Concerns. *University of Washington Ph.D. Generals Examination Report*, September 2000. <http://www.cs.cmu.edu/~aldrich/papers/generals.pdf>

## Professional Activity

Major Conference Program Committees:

European Conference on Object-Oriented Programming (*ECOOP*) 2007

European Conference on Object-Oriented Programming (*ECOOP*) 2006

Joint Modular Languages Conference (*JMLC*) 2006

International Conference on Software Engineering (*ICSE*) 2005

Conference on Aspect-Oriented Software Development (*AOSD*) 2005

Workshop Steering Committees:

Workshop on Specification and Verification of Component-Based Systems, 2007-present.

Workshop Program Committees:

Program Committee Chair, FSE '06 workshop on Specification and Verification of Component-Based Systems (SAVCBS '06)

PC member at SAVCBS '04, '05, '07 workshops

ECOOP '03, '07 Workshop on Ownership, Confinement, and Aliasing

FOAL '04, '05, '06 workshops at AOSD

FSE '06 posters

4<sup>th</sup> Working IEEE/IFIP Conference on Software Architecture

ICSE '05 demonstrations

External reviewer: OOPSLA '02 & '06, POPL '03, '06, and '07, PLDI '06, TOPLAS, ECOOP '04 & '05, J. ACM, J. Functional Programming, Distributed Objects and Applications, ESOP '05, FOSSACS '05, Trans. Software Engineering, Trans. Software Engineering and Methodology, Trans. Aspect Oriented Software Development, Information and Computation, OSDI '06, TLDI '05, The Internet Encyclopedia

Carnegie Mellon University

2005 Chair, ISRI Undergraduate Software Engineering committee.

2004-2006 ISRI admissions committee

2004 CSD admissions committee

University of Washington Department of Computer Science & Engineering:

2002 Graduate Admissions Committee

2001-2002 Faculty Recruitment Liaison

1998-2000 Volunteer Tutor

Green Lake Church, Seattle, WA: 1999-2002 church board

Caltech Christian Fellowship: 1996-1997 President

## Funding

- DARPA Computer Science Study Group member. \$100,000 for one year, potentially renewable for 2 years. 2007.
- 2006 National Science Foundation CAREER award, "Lightweight Modeling and Enforcement of Architectural Behavior," \$450,000 over 5 years. 2006-2010.
- Human and Robotic Technology: Dependable Real-Time and Embedded Space Software, NASA (Michael Shafto, Program Manager), \$1.1 million over one year (2005-2006). Project Lead.
- ITR: Synthetic Reality: Physically Rendering Dynamic 3D Objects from Programmable Matter. National Science Foundation (Helen Gill, Program Manager), \$662,000 over 2 years. Senior Personnel.
- Integrating Software Architecture and Software Development. National Science Foundation (Sol Greenspan, Program Director), \$300,000 over 3 years. Co-written with advisor Craig Chambers while a graduate student; portions subcontracted to my research group at CMU after graduation.
- IRAD: Architecture-Based Self-Adapting Systems. Carnegie Mellon University Software Engineering Institute. With David Garlan and Rick Kazman.
- Open Source Software Evaluation and Assurance. Carnegie Mellon University CyLab. With William Scherlis, Jim Herbsleb, and Aaron Greenhouse.

## Teaching Experience

- Spring 2007. Co-instructor, 15-313 (Foundations of Software Engineering), Carnegie Mellon University. 19 students in class.
- Spring 2007. Instructor, 17-654/17-754 (Analysis of Software Artifacts), Carnegie Mellon University. 44 students in class.
- Spring 2006. Instructor, 17-654/17-754 (Analysis of Software Artifacts), Carnegie Mellon University. Refined course and recorded for distance education. 40 students in class.
- Fall 2005. Instructor, 15-413 (Introduction to Software Engineering), Carnegie Mellon University. Redeveloped course to combine a strong technical focus with a capstone project providing the opportunity to practice engineering knowledge, skills, and practices in a realistic development setting with a real client. 18 students in class.
- Spring 2005. Instructor, 17-654/17-754 (Analysis of Software Artifacts), Carnegie Mellon University. Redeveloped course to focus on a broad range of static and dynamic analysis techniques for programs and other artifacts in the software process. Extended course to fulfill software engineering Ph.D. star course requirement in Analysis. 40 students in class.
- Spring 2005. Co-instructor, 17-898 (Software engineering graduate reading seminar - Modeling Dynamic Software Architectures), Carnegie Mellon University. 9 students and faculty in class.
- Fall 2004. Instructor, 15-819 (Programming languages graduate reading seminar - *Objects and Aspects: Language Support for Extensible and Evolvable Software*), Carnegie Mellon University.
- Fall 2003. Co-instructor, 15-312 (Undergraduate Programming Languages), Carnegie Mellon University. Developed new material on object-oriented programming languages.
- Winter 2002. Teaching Assistant, CSE 503 (Graduate Software Engineering), University of Washington.
- Winter 2001. Teaching Assistant, CSE 501 (Graduate Compilers), University of Washington.
- Summer 1999. Pre-Doctoral Lecturer, CSE 143 (Computer Programming II), University of Washington.
- Spring 1999. Teaching Assistant, CSE 143 (Computer Programming II), University of Washington.
- Fall 1998. Teaching Assistant, CSE 505 (Graduate Programming Languages). University of Washington.

## Students

Donna Malayeri	Fourth-year CSD Ph.D. student.
Neel Krishnaswami	Third-year CSD Ph.D. student co-advised with John Reynolds.
Marwan Abi-Antoun	Second-year ISRI Ph.D. student.
Kevin Bierhoff	Second-year ISRI Ph.D. student.
Ciera Christopher	First-year ISRI Ph.D. student.
Nels Beckman	First-year ISRI Ph.D. student.
Key Shin	Undergraduate thesis advisee in CS.
Matthew Kehrt	Undergraduate thesis advisee in CS. Completed 2006.
Will Cooper	Undergraduate thesis advisee in CS. Completed 2006.
Andi Bejleri	Undergraduate exchange student thesis advisee. Completed 2005.
Lee Salzman	Undergraduate thesis advisee in Logic and Computation. Completed 2004.

Masters independent study/practicum advisees:

Majid Al-Meshari  
Tim Kral  
Joseph Ayo Akinyele  
Lutz Wrage (Completed independent study 2006)  
Varun Dutt (Completed 2006)  
Monica Page (Completed 2006)  
Sangjin Han (Completed 2006)  
Bhavana Rehani (Completed 2006)  
David Dickey (Completed 2006)  
Min Chen (Completed 2005)  
Soumya Simanta (Completed 2005)  
Prasanth Ramanand (Completed 2005)  
Michael German (Completed 2005)  
Animesh Kejriwal (Completed 2005)  
Ben Madore (Completed 2005)

Undergraduate independent study advisees:

Trisha Quan  
Tye Wang (Completed independent study 2006)

## Software Artifacts

The ArchJava Compiler and IDE. Open source software, available 2001-present.  
Acme-ArchJava integration plugin. Open source software, available 2004-present.  
ExnJava Eclipse plugin. Internal development, 2004-2005.  
Dynamic typestate checking tool. Internal development, 2005.  
Architecture synchronization tool. Available on request for research use. Developed 2004-present.  
Architecture protocol checking tool. Internal development, 2006-present.  
JavaD ownership system. Internal development, 2006-present.  
EasyLF proof assistant. Internal development, 2006-present.  
Error reporting system. Internal development, 2006-present.

## External Talks

Young Guns/OO: The Next Generation. Panel participant at OOPSLA 2006, October 2006.

Prospects for Software Assurance. Lockheed Martin, August 2006.

Tales from Dissertationland and the Job Hunt. Invited keynote at the *ECOOP 2005 Doctoral Symposium*, July 2005.

Aspects and Modularity: The Hope and the Challenge. Invited talk, panel on modularity at the *AOSD '05 Workshop on Foundations of Aspect Languages*, March 2005.

Engineering More Dependable Space Software. Talk at NASA Exploration Systems Technology & Systems Integration Technical Interchange Meeting, December 2004.

Using Types to Enforce Architectural Design. Invited talk, California Institute of Technology, October 2003.

Using Types to Enforce Architectural Design. Invited talk, University of Southern California, October 2003.

## Professional Societies

Member, Association for Computing Machinery

Member of SIGSOFT and SIGPLAN special interest groups

Member, Institute of Electrical and Electronics Engineers