

Aaron Greenhouse, Ph.D.

Software Engineering Institute

4500 Fifth Ave.

Pittsburgh, PA 15213

(412) 268-6464 (Office)

(412) 268-5758 (Fax)

120 Ruskin Ave, Apt 110

Pittsburgh, PA 15213

(412) 683-6934 (Home)

(412) 965-5018 (Cell)

aarong@cs.cmu.edu

Current Position

Software Engineering Institute, Pittsburgh, PA

**Member of the Technical Staff in the Performance Critical Systems group in the Dynamic Systems initiative
(Since February 2004)**

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. The SEI staff has advanced software engineering principles and practices and has served as a national resource in software engineering, computer security, and process improvement.

As a member of the Model-Based Engineering project, my work has focused on advancing the use of models and analyses during the development of real-time and embedded systems to document and assure intended system behavior, even at the earliest stages of system design. Our project uses the Society for Automotive Engineers (SAE) Architecture Analysis & Design Language (AADL) for expressing architectural models, and has been instrumental in the development of

- AADL as an international standard
- AADL tutorial material
- Architectural analyses that leverage the expressive power of AADL to document and assure properties and behavior of model systems.

Our project continues to interact with an international mix of governmental and industrial organizations including the US Army, Lockheed Martin, ENST, Airbus, the European Space Agency, Rockwell Collins, Bosch, and Honeywell.

My responsibilities include

- *Developing novel architecture analyses.* Specifically, I have collaborated on the development of a family of “data quality” analyses for attributes such as data security, data accuracy, and data confidence. I am the principal implementer of these analyses for AADL models. I have also contributed to the development and implementation of end-to-end latency, resource scheduling, and other performance-related analyses.
- *Fostering industry adoption of the AADL.* I have contributed to formal tutorials on AADL and OSATE (see below for details), as well as to informal tutorial material including use guides, presentation material, and AADL example models and analysis scenarios that demonstrate the power of model-based engineering.
- *Developing an open-source AADL environment.* I am a primary developer of the Open-Source AADL Tool Environment (OSATE), an extension to the Eclipse IDE that provides industry a freely available toolset with which engineers can explore AADL and appreciate the power of MBE and architectural analysis. It is also a platform on top of which other researchers, including those at University of Pennsylvania, University of Virginia, and Embry-Riddle Aeronautical University, are building new architectural analyses.
- *Developing the AADL standard.* OSATE serves as the prototype implementation of AADL by the SAE standards committee and thus as a vehicle to implement and test proposed extensions to the language. As an implementer of OSATE, I am intimately familiar with AADL and have contributed to the development of the standard by identifying semantic gaps and ambiguities in the language.

Research Interests

Object-oriented design; programming languages; static analysis for program understanding; safe and assured evolution of software systems; concurrent programming; programming environments; architecture design languages; static analysis of architecture attributes

Education

- 1996–2003 **Carnegie Mellon University, Pittsburgh, Pennsylvania**
Ph.D. in Computer Science, May 2003
Thesis: *A Programmer-Oriented Approach to Safe Concurrency*
Committee: Dr. William L. Scherlis (Chair), Dr. Thomas Gross (Co-chair), Dr. Guy Blelloch, Dr. John Boyland (University of Wisconsin–Milwaukee)
- M.S. in Computer Science, May 1999**
- 1992–1996 **Brandeis University, Waltham, Massachusetts**
B.A. in Computer Science (Mathematics minor), May 1996
Summa cum laude, High honors
Senior thesis: *A Computer-Assisted Micro-Analysis of the NESL Programming Language*

Citizenship

I am a citizen of the **United States of America**.

Past Experience

Institute for Software Research, International, Carnegie Mellon University, Pittsburgh, PA
Post-Doctoral Fellow in the Fluid Group (<http://www.fluid.cs.cmu.edu>)
(June 2003–February 2004)

The research agenda of the Fluid group led by Dr. William Scherlis is to develop program analysis techniques to assure program safety with respect to programmer-declared design intent. For example, assurance that pointers are unaliased, or that locks are acquired before accessing shared state. A dominant design consideration is that our techniques be adoptable at scale by practicing programmers. My research agenda and responsibilities included

- Presenting assurance results in a manner useful to practicing programmers.
- Demonstrating the effectiveness of our approach by applying our tool to production software.
- Overseeing the research and development of our prototype analysis tool.
- Mentoring and managing the project's graduate students and undergraduate researchers.

Computer Science Department, International, Carnegie Mellon University, Pittsburgh, PA
Graduate Student
(Fall 1996–May 2003)

Graduate student in the Fluid group (see description above). In addition to completing my dissertation research, I also

- Designed and implemented two generations of a Model–View–Controller framework in Java enabling fine-grained control over the display of complex data structures based on ternary (identifier–attribute–value) representations.
- Contributed to the design, implementation, and maintenance of a 160kLOC Java application for program analysis and source code manipulation.
- Mentored and managed the project's junior graduate students and undergraduate researchers.

The Aerospace Corporation, El Segundo, CA
Summer Intern
(Summer 1997)

Primary designer and implementer of a client–server Java application to send rocket telemetry information to desktop computers. As part of the project, I developed specialized GUI components based on the model–view paradigm.

Clinical Center, National Institutes of Health, Bethesda, MD
Summer Intern
(Summer 1996)

Intern in the nuclear medicine department where I assisted in the development of image processing techniques with the goal of using preprocessing to improve medical image registration results. In particular, I focused on developing preprocessing steps that would enable the alignment of images originating from complementary (*e.g.*, CAT vs. MRI) scanning technologies.

National Institute of Standards and Technology, Gaithersburg, MD
Summer Intern
(Summers 1991–1995)

Intern for the Video Technology Project, whose mission was to develop quantitative techniques for the evaluation of the quality of video sequences. Such metrics could be used, for example, to objectively compare the effects of lossy compression algorithms. I was the primary designer and implementer of several applications, including

- A suite of programs that generate video sequences used in evaluating MPEG artifacts.
- A tool suite and library for the manipulation of digital video sequences.
- A block-motion estimation routine on a SIMD, massively parallel, video supercomputer to be used to evaluate the suitability of using motion vectors for frame-rate conversion.

Publications

Dissertation

- Aaron Greenhouse. *A Programmer-Oriented Approach to Safe Concurrency*. Ph.D. Thesis. Carnegie Mellon University School of Computer Science, Pittsburgh, PA. May 2003. Technical Report CMU-CS-03-135.

Refereed Journal Papers

- Aaron Greenhouse, T. J. Halloran, and William L. Scherlis. “Observations on the Assured Evolution of Concurrent Java Programs.” *Science of Computer Programming* 58(3):384–411. doi:10.1016/j.scico.2005.03.002

Refereed Conference Papers

- Dean F. Sutherland, Aaron Greenhouse, and William L. Scherlis. “The Code of Many Colors: Relating Threads to Code and Shared State.” *Workshop on Program Analysis for Software Tools and Engineering (PASTE ’02)*, November 2002.
- Aaron Greenhouse and William L. Scherlis. “Assuring and Evolving Concurrent Programs: Annotations and Policy.” *International Conference on Software Engineering (ICSE ’02)*, pages 453–463. May 2002.
- Aaron Greenhouse and John Boyland. “An Object-Oriented Effects System.” *The 13th European Conference on Object-Oriented Programming (ECOOP ’99)*, Volume 1628 of *Lecture Notes in Computer Science*, pages 205–229, June 1999.

Refereed Workshop Papers

- Aaron Greenhouse, T. J. Halloran, and William L. Scherlis. “Observations on the Assured Evolution of Concurrent Java Programs.” *2004 Workshop on Concurrency and Synchronization in Java Programs (CSJP)*. Memorial University of Newfoundland, Computer Science Technical Report #2004-01, pages 90–99. July 2004.
[11 papers accepted out of 16 submissions]
- Aaron Greenhouse, T. J. Halloran, and William L. Scherlis. “Using Eclipse to Demonstrate Positive Static Assurance of Java Program Concurrency Design Intent.” *Proceedings of the 2003 OOPSLA workshop on eclipse Technology eXchange (eTX)*, pages 99–103. October 2003.
[21 papers accepted out of >50 submissions]

Workshop Papers

- Elissa Newman, Aaron Greenhouse, and William L. Scherlis. “Annotation-Based Diagrams for Shared-Data Concurrency.” *Workshop on Concurrency Issues in UML at UML 2001*. October 2001.
- John Boyland and Aaron Greenhouse, “MayEqual: A New Alias Question.” *The Intercontinental Workshop on Aliasing in Object-Oriented Systems at ECOOP ’99*, June 1999. Summarized in *Object-Oriented Technology: ECOOP 1999 Workshop Reader*, Volume 1743 of *Lecture Notes in Computer Science*, pages 141–143.

Tutorials

- “Plug-in Development for the Open Source AADL Tool Environment.” Given in 7 sessions during April and May 2006 to an audience at Rockwell Collins. A total of 13 hours of material delivered remotely using WebEx.
- “Plug-in Development for the Open Source AADL Tool Environment.” Joint instructor with Peter Feiler and Lutz Wrage. An SEI-hosted tutorial in four parts to an international audience: 9 Dec 2004, 15 Dec 2004, 16 Dec 2004, and 5 Jan 2005.

Presentations

- November 2006 “Modeling Security in System Architectures.” Invited talk, Distributed Processing Laboratory, Lockheed Martin Advanced Technology Labs, Cherry Hill, NJ.

February 2006 “Fluid: Incremental Concurrency Analysis.” Guest lecture, CMU MSE/SE PhD course 17-654/17-754
“Analysis of Software Artifacts” (Prof. Jonathan Aldrich).

July 2005 “OSATE: Open Source AADL Tool Environment” (with Bruce Lewis).
Prometheus Software Tools Infrastructure Working Group, hosted by Northrop Grumman Space
Technology, Redondo Beach, CA. Presented remotely.

July 2004 “Observations on the Assured Evolution of Concurrent Java Programs.”
Workshop on Concurrency and Synchronization in Java Programs at PODC 2004, St. John’s,
Newfoundland.

October 2003 “Using Eclipse to Demonstrate Positive Static Assurance of Java Concurrency Design Intent.”
eclipse Technology eXchange (eTX) Workshop at OOPSLA 2003, Anaheim, CA.

August 2003 “A Programmer-Oriented Approach to Safe Concurrency.” Invited talk, Intel Research Pittsburgh.

March 2003 “A Programmer-Oriented Approach to Safe Concurrency.”
Public thesis oral, Carnegie Mellon University

February 2003 “A Programmer-Oriented Approach to Safe Concurrency.”
Software Research Seminar, Carnegie Mellon University

May 2002 “Assuring and Evolving Concurrent Programs.”
24th International Conference on Software Engineering, Orlando, Florida

April 2002 “Assuring and Evolving Concurrent Programs.” Software Research Seminar, Carnegie Mellon University

October 2001 “Extended Static Checking: An Introduction.” Software Research Seminar, Carnegie Mellon University

April 2001 “An Introduction to Concurrency Policy.” Software Research Seminar, Carnegie Mellon University

September 2000 “Evolving and Maintaining Thread-Safe Classes using Program Transformation.”
Software Systems Study Group, Carnegie Mellon University

July 2000 “Concurrency Policy: Its Role and Specification.”
Computer Systems Seminar, Carnegie Mellon University

November 1999 “Synchronization Manipulations.” Computer Systems Seminar, Carnegie Mellon University

October 1999 “An Object-Oriented Effects System.” Software Systems Study Group, Carnegie Mellon University

June 1999 “MayEqual: A New Alias Question.” Computer Systems Seminar, Carnegie Mellon University

June 1999 “An Object-Oriented Effects System.”
13th European Conference on Object-Oriented Programming, Lisbon, Portugal

April 1999 “An Object-Oriented Effects System.” Principles of Programming Seminar, Carnegie Mellon University

April 1999 “A Tool for Systematic Program Evolution and its Object-Oriented Effects System.”
Student Seminar Series, Carnegie Mellon University

February 1999 “An Object-Oriented Effects System.” Computer Systems Seminar, Carnegie Mellon University

November 1998 “Interacting with a Tool for Program Evolution.”
Computer Systems Seminar, Carnegie Mellon University

July 1998 “Moving Fields to a Superclass in Java.” Computer Systems Seminar, Carnegie Mellon University

March 1998 “An Effects Analysis for Object-Oriented Programs.”
Computer Systems Seminar, Carnegie Mellon University

November 1996 “A Computer-Assisted Micro-Analysis of the NESL Programming Language.”
Software Systems Study Group, Carnegie Mellon University

Service

Program Committees

- *Synchronization and Concurrency in Object-Oriented Languages (SCOOL)*, OOPSLA 2005 Workshop.
Reviewer for subsequent special issue of *Science of Computer Programming*.

Reviewer

- *ECOOP* 2002
- *Formal Techniques for Java-like Programs* 2004, 2005
- *Generative Programming and Component Engineering* 2005

- *International Conference on Software Reuse* 2004
- *Language Descriptions, Tools, and Applications* 2004, 2005
- *Workshop on Object-Oriented Developments* 2003

Academics—Guest Lecturer

- CMU MSE/SE PhD course 17-654/17-754 “Analysis of Software Artifacts” (Prof. Jonathan Aldrich).
Topic: “Fluid: Incremental Concurrency Analysis.” 2 February 2006.

Academics—Thesis Committee

- Master’s Thesis Committee, Capt Scott C. Hale, USAF, *FlashLight: A Dynamic Detector of Shared State, Race Conditions, and Locking Models in Concurrent Java Programs*. Air Force Institute of Technology. Completed: March 2006.

Academics—Reader

- Faculty reader for Lee Salzman’s Logic and Computation senior thesis *Prototypes with Multiple Dispatch* in the Department of Philosophy at Carnegie Mellon University, May 2004.
- Faculty reader for Software Engineering Ph.D. practicum reports, CMU School of Computer Science:
 - Timothy J. Halloran. “Development of the StratWar Wargame Software.” December 2003.
 - Timothy J. Halloran. “Software Modernization: Leading the Transition of a Software Maintenance Organization.” December 2003.
 - Dean F. Sutherland. “A Tale of Three Processes: Reflection on Software Development Process Change at Tartan.” December 2004.

University Service

- Faculty member, 2004 Software Engineering Ph.D. admission committee, CMU School of Computer Science.
- Student member, 1999–2002, CMU Computer Science Department “Speakers Club:”

Graduate students in the Computer Science Department must demonstrate the “ability to communicate technical ideas clearly orally” via a public presentation. Members of the “speakers club” evaluate the presentation skills of students so that they may pass this oral communication requirement. Members are chosen based on recognition of their own outstanding presentation skills.

Society Memberships

- Member, Association for Computing Machinery (SIGPLAN, SIGSOFT)
- Member, IEEE and IEEE Computer Society

Honors

October 2002 Siebel Scholar, Class of 2003; <http://www.siebelscholars.com/about.asp>
 May 1996 Phi Beta Kappa, Mu Chapter of Massachusetts