

Thomas H. Austin

CONTACT

103 Koshland Way (415) 307-5322
Santa Cruz, CA 95064 taustin@ucsc.edu
United States <http://www.bias2build.com>
<http://tomthemighty.blogspot.com>

EDUCATION

Ph.D. student, **University of California at Santa Cruz**.
Graduation date: August 2012 (expected).
Computer Science major, Software and Languages Research Group (SLANG).

M.S., **San José State University**.

Graduation date: May 2008.
Computer Science major.

B.S., **Santa Clara University**.

Graduation date: June 1998.
Operations and Management of Information Systems major, Spanish minor.

PUBLICATIONS

Book Chapters

Thomas H. Austin. Designing a secure programming language. In Peter Stavroulakis and Mark Stamp, editors, *Handbook of Information and Communication Security*, chapter 35, pages 771-785. Springer-Verlag, Berlin, Heidelberg, 2010.

Peer-reviewed Conference Papers

Thomas H. Austin and Cormac Flanagan. Multiple facets for dynamic information flow. In *POPL 2012: Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 165-178, Philadelphia, Pennsylvania, USA, January 22-28, 2012. ACM, 2012.

Thomas H. Austin, Tim Disney, and Cormac Flanagan. Virtual values for language extension. In *OOPSLA 2011: Proceedings of the 26th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, part of SPLASH 2011*, pages 921-938, Portland, OR, USA, October 22 - 27, 2011. ACM, 2011.

Peer-reviewed Workshop Papers

Thomas H. Austin and Cormac Flanagan. Permissive dynamic information flow analysis. In *PLAS 2010: Proceedings of the ACM SIGPLAN Fifth Workshop on Programming Languages and Analysis for Security*, New York, NY, USA, 2010. ACM.

Thomas H. Austin and Cormac Flanagan. Efficient purely-dynamic information flow analysis. In *PLAS 2009: Proceedings of the ACM SIGPLAN Fourth Workshop on Programming Languages and Analysis for Security*, pages 113-124, New York, NY, USA, 2010. ACM.

Master's Thesis

Thomas H. Austin. Expanding JavaScript's metaobject protocol. Master's thesis, San José State University, 2008.

Technical Reports

Thomas H. Austin, Tim Disney, Cormac Flanagan, Alan Jeffrey. Dynamic information flow analysis for Featherweight JavaScript. Technical Report UCSC-SOE-11-19, The University of California at Santa Cruz, 2009. <http://www.soe.ucsc.edu/research/report?ID=1560>.

Trade Publications

Thomas H. Austin. Rails vs. Grails vs. Helma: The JVM web framework smackdown (2 part article). *Linux Magazine*, 4, 2009. Part 1: <http://www.linux-mag.com/cache/7470/1.html>, part 2: <http://www.linux-mag.com/cache/7479/1.html>.

WORK EXPERIENCE

École Supérieure d'Informatique Électronique Automatique Ouest. Laval, France.

Invited Researcher: Laboratoire de Recherche Cryptologie et Virologie Opérationnelles.

September 2011 – December 2011.

I was invited as part of a joint project between San José State University and ESIEA. Our project applied hidden Markov models to the identification of metamorphic computer viruses.

Mozilla Corporation. Mountain View, CA.

Intern: Research Group. June 2011 – September 2011.

I was involved with the dom.js project and the development of the ZaphodFacets Firefox plugin. This plugin integrated information flow controls into the browser, based on research outlined in my POPL 2012 paper: *Faceted Values for Information Flow Control*.

Mozilla Corporation. Mountain View, CA.

Intern: Research Group. June 2010 – September 2010.

I developed Zaphod, a Firefox plugin for integrating experimental JavaScript features. Work involved extending and updating the Narcissus JavaScript engine.

McClatchy Interactive West (formerly Knight Ridder Digital). San Jose, CA.

Senior Software Engineer: Engineering Department. July 1998 – July 2008.

McClatchy Interactive is the online division of the McClatchy Company, the 3rd largest newspaper publisher in the United States. I was the lead developer for classified advertising data feeds, though I was involved with some web development efforts as well. During my time there, I rose from a junior-level programmer to a senior software engineer.

OPEN SOURCE PROJECTS

Narcissus: *JavaScript metacircular interpreter.* <http://github.com/mozilla/narcissus>.

Initially developed by Brendan Eich, Narcissus is a JavaScript implementation written in JavaScript. I removed its dependency on Firefox build-specific features, set up functionality tests, and did some restructuring of the interpreter.

Zaphod: *Narcissus integration for Firefox.* <http://mozillalabs.com/zaphod>

Primary contributor. I developed Zaphod to integrate Narcissus into Firefox as a vehicle for testing experimental JavaScript features.

Dom.js: *Document Object Model in JavaScript.* <http://github.com/andreagal/dom.js>.

Dom.js is an implementation of the Document Object Model written entirely in JavaScript. Part of my role was to integrate dom.js with Narcissus and Zaphod.

ZaphodFacets: *JavaScript with information flow.* <http://github.com/taustin/ZaphodFacets>.

Primary contributor. A Zaphod fork that integrates information flow controls into the interpreter. The design of the controls is based on my work in *Faceted Values for Information Flow Control*, published in POPL 2012.

RhinoFaces: *Rhino JavaScript on the server.* <http://java.net/projects/rhinofaces>.

Primary contributor. This web framework is patterned after Ruby on Rails, but uses Rhino JavaScript and JavaServer Faces. The Rhino JavaScript implementation uses a metaobject protocol that I had developed as part of my Master's thesis.

XMUltra: *Java/XML datafeed framework.* <http://xmultra.sourceforge.net>.

XMUltra a Java and XML based feed processing framework. It has a variety of utilities available for feed processing.

SKILLS

Languages

JavaScript, Java, Scala, Groovy, C/C++, Ruby, Python, PHP, Scheme, Smalltalk, OCaml, Perl, Prolog, and L^AT_EX.

Web Frameworks

Ruby on Rails, JavaServer Faces, Helma Object Publisher, and Groovy on Grails.

Compiler Tools

Flex, Bison, JavaCC, and ANTLR.

Selected School Projects

- *MD5 cryptographic hash project.* Implemented MD5 algorithm in C, implemented the Wang/Yu attack against MD5 in C, and created 2 different postscript files with the same MD5 hash value.
- *ThaTune\$.* A shell to interact with Apples iTunes Store. Worked with iTunes v. 5. Written in Java.
- *GAMSA (Genetic Algorithm for Multiple Sequence Alignment).* Used a genetic algorithm to align sequences of nucleotides and amino acids. Written in Java.
- *Information extraction package for print classified ads.* Included a tool for evaluating the recall and precision of the model. Written in Ruby.
- *Language model for classified ads.* Built with Python scripts and SRI's Language Modeling Toolkit.