

# Eric Schulte | Director of Automated Software Engineering

Pittsburgh, PA

☎ (607) 273 7340 × 190 • ✉ [eschulte@grammatech.com](mailto:eschulte@grammatech.com) • 🌐 [eschulte.github.io](https://eschulte.github.io)  
in [eric-schulte-255bb613](https://github.com/eric-schulte-255bb613) • 🐙 [eschulte](https://github.com/eschulte)

## Experience

---

### GrammaTech

Ithaca, NY

*Director of Automated Software Engineering*

2018 – present

- Propose and lead government funded research and development projects.
- Manage technical staff including software engineers and senior scientists.
- Lead GrammaTech's *Binary Rewriting* and *Program Synthesis* research areas:
  - Program Synthesis*: Programmatic analysis and transformation of software source-code to automate common software engineering tasks including refactoring, maintenance, optimization, bug repair, and bug injection
  - Binary Rewriting*: Programmatic analysis and transformation of stripped COTS binaries, without access to source-code, to improve their security and efficiency
- Maintain and grow technical infrastructure, support commercial transition.
  - Helped spur and manage a company-wide transition from SVN to Git and adoption of continuous integration (CI) practices.
  - Managing refactoring of a 2-decade-old C/C++ code base to modular, testable, modern C++, built with CMake. Already resulting in dramatically increased developer productivity. Done *in-flight* will completing contract deliverables.
  - Manage open-source projects including;
    - *Software Evolution Library*: <https://github.com/grammatech/sel>
    - *GrammaTech Intermediate Representation for Binaries*: <https://github.com/grammatech/gtirb>
    - *Datalog Disassembler*: <https://github.com/grammatech/ddisasm>
- Lead or oversee the following research and development efforts:
  - Debloat*: Principle Investigator leading a project to develop robust binary rewriting for the purpose of removing unwanted functionality from legacy binaries to reduce attack surface and improve efficiency. This effort includes; the creation of novel binary analysis and rewriting techniques (including an industry leading datalog disassembly engine) and intermediate representations, the development and maturation of aggressive novel software optimization techniques, binary rewriting techniques for inlining shared libraries into static executables, and binary rewriting to remove unused code from software.
  - Bug-Injector*: Principle Investigator leading the development of a tool for the automated injection of bugs into software source code. Host programs in multiple languages including C/C++ and JavaScript are supported, and arbitrary bugs may be injected through user-defined bug templates. This produces benchmarks which are useful in the evaluation of static and dynamic software analysis tools.
  - Resolve*: Leading the development of practical tools for the improved difference and merging of branches of software repositories. This project seeks to improve the state of the art available to developers to view software differences and to automate the time-consuming task of merge conflict resolution.
  - BRASS*: Supervise GrammaTech's effort to build tools to automate software maintenance. Automated tasks include upgrading software to use newer versions of third-party dependencies, and performing large scale syntax-aware AST-level search and replace operations.

### Senior Scientist

2014 – 2018

- Contributed to the following research and development efforts:
  - CFAR*: Helped lead an effort to use binary rewriting to generate N-variant systems from a single stripped COTS binary. Developed techniques leveraging SMT solvers to generate provably-secure multi-variant memory configurations.
  - BED*: Led research into the use of evolutionary techniques to evolve *exact* C decompilation of stripped COTS binaries.
  - GenPatcher*: Implemented a tool for the automated repair of flaws in binary executables.

## University of New Mexico

Research Assistant

Albuquerque, NM

2009 – 2014

- Research and development of evolutionary techniques for software maintenance and improvement.
  - Developed novel techniques of software repair applicable to compiled assembler code and stripped binaries.
  - Developed an evolutionary search-based optimization technique applicable to compiled assembly code. Reduced energy consumption of a range of benchmark programs by 20% beyond the best available compiler optimizations.
- Empirical and theoretical investigation of biological properties of software.
  - Empirically explored the robustness of software to random perturbation. Demonstrated mutational robustness across; C/C++ source, compiled assembler, LLVM intermediate representation, and stripped executables.
  - Identified empirical evidence that the modern software development ecosystem is the product of an evolutionary process in which software developers and users perform mutation, reproduction, and fitness evaluation.

## Counsyl

Open Source Software Consultant

Palo Alto, CA

2010 – 2011

- Consultation and development in application of open-source software to automated technical document generation.

## The MITRE Corporation

Senior Artificial Intelligence Engineer

McLean, VA

2005 – 2009

- Lead developer of the Rapid Argus Modeling for Biosurveillance Operations (RAMBO) system for disease modeling and surveillance. RAMBO ran Bayesian models over text media statistics collected from across the world to identify emerging epidemics. RAMBO was used daily by roughly 50 biosecurity analysis at Georgetown University.
- Prototype the STAT (Statistical Tracking and Analysis of Text) system for temporal analysis of multilingual text.
- Systems administration for production Unix/Linux systems.

International Technical Analyst

2004 – 2005

- Research assistant, composed documentation and user manuals.

## Volunteer

---

### Planned Parenthood

Clinic Escort

Pittsburgh, PA

2019 – present

### National Poor People's Campaign

Web Design

2020 – present

- Build and maintain the *Nonviolent Medicaid Army* web site at <http://nonviolentmedicaidarmy.org>.
- Build and maintain the *National Union of the Homeless* web site at <https://nationalunionofthehomeless.org>.

### GNU Emacs

Contributor

2009 – 2014

- Author and maintainer of Emacs Org-mode's facilities for embedding executable source code into documents.

## Education

---

### University of New Mexico

Ph.D., Computer Science

Advisor: Stephanie Forrest

Albuquerque, NM

2014

Thesis: Neutral Networks of Real-World Programs and their Application to Automated Software Evolution

### Kenyon College

B.A., Mathematics, Minor Philosophy

Gambier, OH

2004

## Skills

---

Programming languages: Common Lisp, C/C++, JavaScript, Python, OCaml, Clojure, Scheme, Prolog/Datalog

Technical Expertise: Emacs, Git, Docker, CI, Linux, GDB, SMT/SAT (Z3, CVC4), Markdown,  $\text{\LaTeX}$ , HTML/CSS

Domains: Artificial Intelligence, Programming Languages, Software Engineering, Binary Analysis, Formal Methods

## Publications

---

### Doctoral Thesis

Eric Schulte. *Neutral Networks of Real-World Programs and their Application to Automated Software Evolution*. PhD thesis, University of New Mexico, Albuquerque, USA, July 2014. <https://cs.unm.edu/~eschulte/dissertation>.

### Refereed Conference Publications

Antonio Flores-Montoya and Eric Schulte. Datalog disassembly. In *29th USENIX Security Symposium (USENIX Security 20)*, 2020. Distinguished Paper.

Vineeth Kashyap, Jason Ruchti, Lucja Kot, Emma Turetsky, Rebecca Swords, David Melski, and Eric Schulte. Automated customized bug-benchmark generation. In *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 103–114. IEEE, 2019. Distinguished Paper.

Deborah Katz, Jason Ruchti, and Eric Schulte. Using recurrent neural networks for decompilation. In *Software Analysis, Evolution and Reengineering (SANER)*, 2018. IEEE, 2018.

Eric Schulte, Jonathan Dorn, Stephen Harding, Stephanie Forrest, and Westley Weimer. Post-compiler software optimization for reducing energy. In *Proceedings of the eighteenth international conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '14*. ACM, 2014, *Acceptance Rate: 22.6%*.

Eric Schulte, Jonathan DiLorenzo, Westley Weimer, and Stephanie Forrest. Automated repair of binary and assembly programs for cooperating embedded devices. In *Proceedings of the eighteenth international conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '13*. ACM, 2013, *Acceptance Rate: 22.8%*.

Eric Schulte, Stephanie Forrest, and Westley Weimer. Automated program repair through the evolution of assembly code. In *Proceedings of the IEEE/ACM international conference on Automated software engineering, ASE '10*, pages 313–316, New York, NY, USA, 2010. ACM, *Acceptance Rate: 17.8%*.

### Refereed Journal Articles

Eric Schulte, Zachary. Fry, Ethan Fast, Westley Weimer, and Stephanie Forrest. Software mutational robustness. *Genetic Programming and Evolvable Machines*, pages 1–32, 2013, *Impact Factor: 1.333*.

Eric Schulte, Dan Davison, Thomas Dye, and Carsten Dominik. A multi-language computing environment for literate programming and reproducible research. *Journal of Statistical Software*, 46(3):1–24, 1 2012, *Impact Factor: 4.910*.

Paul Lehner, Charles Worrell, Chrissy Vu, Janet Mittel, Stephen Snyder, Eric Schulte, and Warren Greiff. An application of document filtering in an operational system. *Information Processing & Management*, 46(5):611–627, 2010, *Impact Factor: 0.817*.

### Magazine Articles

Eric Schulte and Dan Davison. Active document with org-mode. *Computing in Science & Engineering*, 13(3):66–73, May/June 2011, *Impact Factor: 1.72*.

### Workshop Papers

Eric Schulte, Suan Yong, and David Melski. Inuring: Live attacker-guided repair. In *Proceedings of the 3rd ACM Workshop on Forming an Ecosystem Around Software Transformation*, pages 39–45, 2019.

Benoit Baudry, Nicolas Harrant, Eric Schulte, Chris Timperley, Shin Hwei Tan, Marija Selakovic, and Emamurho Ugherughe. A spoonful of devops helps the gi go down. 2018.

Eric Schulte, Jason Ruchti, Matt Noonan, David Ciarletta, and Alexey Loginov. Evolving exact decompilation. In *Binary Analysis Research (BAR)*, 2018, 2018.

Vineeth Kashyap, Rebecca Swords, Eric Schulte, and David Melski. Musynth: Program synthesis via code reuse and code manipulation. In *International Symposium on Search Based Software Engineering*, pages 117–123. Springer, 2017.

Eric Schulte, Westley Weimer, and Stephanie Forrest. Repairing COTS router firmware without access to source code or test suites: A case study in evolutionary software repair. In *Genetic Improvement 2015 Workshop*, pages 847–854, Madrid, 11-15 July 2015. ACM. Best Paper.