PETER C DILLINGER, Ph.D.

peterd@gatech.edu
http://www.peterd.org

2110 N 89th St
Seattle WA 98103
404-509-4879

Overview My work in software has focused on **software tools** that support software engineering. My core strength is seeing **many paths to "success,"** so I'm often the person consulted when others think they're stuck.

Highlights

♦ **Key developer and project lead** in adapting and extending the legendary Coverity static analysis engine, for C/C++ bug finding, to find bugs with high accuracy in Java, C#, JavaScript, PHP, Python, Ruby, Swift, and VB. https://www.synopsys.com/blogs/software-security/author/pdillinger/

♦ **Inventor** of a fast, scalable, and accurate method of detecting mistyped identifiers in dynamic languages such as JavaScript, PHP, Python, and Ruby without use of a natural language dictionary. Patent pending with Synopsys; part of Coverity product. https://stackoverflow.com/a/34796105

♦ **Did the impossible with git**: on wanting to "copy with history" as part of a refactoring, I quickly developed a way to do it despite the consensus wisdom. https://stackoverflow.com/a/44036771

♦ **Did the impossible with Bloom filters**: made the data structure simultaneously fast and accurate with a simple hashing technique. https://en.wikipedia.org/wiki/Bloom_filter (Search "Dillinger")

♦ **Early coder**: in 3rd grade, my school had a lemonade stand game on the Apple][. I noticed the game was written in BASIC and quickly hacked it for consistent sunny weather (→ profit).

Work

Coverity, August 2009 to October 2017, acquired by ***Synopsys*** in 2014
Software developer, tech lead, and manager for static and dynamic program analysis projects.
Promotions: Senior Engineer, 2010. Principle Engineer, 2013. Manager, 2015.

♦ **New code checkers** (**two patents pending**): Coverity software automatically finds coding mistakes that matter to programmers, by going much deeper and by being more clever and more cautious than linters. I conceived, designed, implemented, and/or oversaw many successful checkers for new bug patterns. The mistyped identifier checker was a solo side project while a manager.

♦ **Full support of new languages**: I was a key contributor (Java and JavaScript) or project lead (C#) in adapting the Coverity engine to support new programming languages, sucessfully meeting goals for defect density and false positive rates to claim full support for these languages.

♦ **Fast support of new languages**: When highest leadership called for rapid expansion of language support, I architected, negotiated, and delivered on a strategy for rapid support of new languages (PHP, Python, Ruby, Swift, and VB) that compromised defect depth and variety rather than defect quality and credibility, and left a good technical path to full language support. Ultimately, middle management knew the project was all but doomed without a strategy meeting my approval, and after my compromise strategy came in on scope and on time, no one was saying "if only ...".

→ This job included tons of reading and understanding random open-source code, in all languages listed above, to find false positive defect reports on real code before customers do.

→ Most development was in C++ on Linux, plus some in Java, some on Windows, etc.

→ Also ask me about these accomplishments (D=designed, I=implemented):
efficient and non-lossy type tracking (D&I), exception analysis (D&I),
enhancing the Coverity secret sauce (D&I, very few hands touch this),
improved parallel regression test scheduling (D&I), handling of intermittently failing tests (D&I),
checking whole-program properties of partial programs (D), and
components for checking Web app security / XSS (D/2 & I).

Education ***Doctor of Philosophy in Computer Science*** · Northeastern University, 2007 - 2010; transfer with advisor from Georgia Tech, 2003 - 2007

♦ Adaptive Approximate State Storage (thesis research): Explicit-state model checkers use approximate (probabilistic) state storage techniques such as Bloom filters to reduce memory requirements by orders of magnitude while maintaining very high defect detection ability. My Bloom filter improvements (in highlights above) landed me an internship at JPL where I integrated

them into the award-winning model checker Spin.
But the problem remained that the highest accuracy for available memory was only possible with unrealistic prior knowledge of the reachable state space size. I solved this problem with a much more advanced system for fast, approximate state storage that adapts on-the-fly to an arbitrary state space size. An information theoretic argument shows that a real user of my approach with *2m* bytes of memory will be at least as accurate as, and about as fast as, an oracle with *m* bytes of memory.
♦ ACL2s: 'The ACL2 Sedan" (assistantship research, 2005 - 2009): I am the initial developer of ACL2s, an Eclipse-based development environment for the award-winning ACL2 theorem proving system. This tool sucessfully lowered barriers to entry so that freshman students could be taught to use an industrially successful automatic theorem proving system (see 'teaching' below).

Master of Science in Computer Science · Georgia Tech · 2002 - 2003

Bachelor of Science in Computer Science · Georgia Tech · 1999 - 2002
3.70 / 4.00 GPA, highest honors

Publications (peer-reviewed)

♦ Fast, All-Purpose State Storage. SPIN Workshop, 2009.
Coauthor: Panagiotis Manolios.
♦ Hacking and Extending ACL2. ACL2 Workshop, 2007.
Coauthors: Matt Kaufmann (Erdős number of 2) and Panagiotis Manolios.
♦ ACL2s: 'The ACL2 Sedan". Workshop on User Interfaces for Theorem Proving (UITP), 2006.
Coauthors: Panagiotis Manolios, J Moore, and Daron Vroon. (Another version was a demo at ICSE 2007.)
♦ Bloom Filters in Probabilistic Verification. Conference on Formal Methods in Computer-Aided Design (FMCAD), 2004.
Coauthor: Panagiotis Manolios.
♦ Fast and Accurate Bitstate Verification for SPIN. SPIN Workshop, 2004.
Coauthor: Panagiotis Manolios.

Internships

NASA Ames Research Center · Robust Software Engineering group · Summer 2006
♦ Contributions to Java PathFinder tool, under Willem Visser, Corina Pasareanu, and Peter Mehlitz
NASA/Caltech Jet Propulsion Laboratory · Laboratory for Reliable Software · Summer 2004
♦ Contributions to Spin model checker, under Gerard Holzmann and Rajeev Joshi

Teaching

Instructor for CS U290: Logic and Computation · Northeastern University · Fall 2008
Taught a relatively new class introducing college freshmen to logic through the ACL2 programming language, logic and theorem prover, using my ACL2s development environment. I developed all lectures, assignments, and tests. http://www.ccs.neu.edu/home/pcd/csu290-fall2008/

Instructor in Math/CS · Georgia Governor's Honors Program · Summers 2005 & 2007
Taught summer enrichment courses for enthusiastic high school students. Based on my own curricula, courses included Cryptography, Logic, Theory of Computation, Functional Programming, and Introductory Programming.

Teaching assistant · Georgia Tech and Northeastern University · 2000 - 2003, 2008

References

Available upon request

Other Information

U.S. Citizen