Curriculum Vitae
Shan Shan Huang

College of Computing
Georgia Institute of Technology
266 Ferst Drive
Atlanta, GA 30332-0765

Phone: *Please email for phone number*
Email: ssh@cc.gatech.edu
http://www.freeflygeek.com

Research Interests

Application of programming language techniques to support software engineering. Specifically, designing language abstraction mechanisms to support modular program construction; using advanced type systems to guarantee program correctness properties at compile-time; applying program generation techniques to aid in domain-specific language implementation and automated software engineering.

Education

Ph.D. in Computer Science, expected (August 2003 — July 2009)
Georgia Institute of Technology, College of Computing, Atlanta, GA
Advisor: Yannis Smaragdakis
Design and develop language abstraction mechanisms that support better modularity and code reuse; Apply advanced type system techniques to statically guarantee the various safety properties of programs; Develop program generation tools to aid in automated software engineering.

B.S. in Electrical Engineering and Computer Science (September 1996 — June 2000)
Massachusetts Institute of Technology, Cambridge, MA

Honors and Awards

Fellowships & Scholarships

- National Science Foundation (NSF) Graduate Research Fellowship, 2005 — 2010.
- Intel Ph.D. Fellowship, 2005.
- Presidential Fellowship, Georgia Institute of Technology, 2003 — 2008.
- College of Computing Dean's Supplement Award, Georgia Institute of Technology, 2003 — 2008.

Paper Awards

- Best Paper Award [8], Generative Programming and Component Engineering Conference (GPCE 2004).

Research Experience

Summer Research Intern (June 2007 — August 2007)
IBM T.J. Watson Research Center, Hawthorne, NY
Mentors: David Bacon, Rodric Rabbah
Summer research intern on the Liquid Metal project. Liquid Metal aims to have one unified language, Lime, whose programs can be compiled against and run on a heterogeneous architecture—an architecture where CPUs, FPGAs, cell processors, etc., co-exist. I designed and developed

Lime, an extension of Java that provides the high-level abstractions that software engineers are familiar with, yet at the same time, amenable to bit-level analysis and exposes parallelism—crucial properties for efficient synthesis to hardware. Work resulted in publication [3].

Summer Research Intern (June 2005 — August 2005)
Sandia National Laboratory, Albuquerque, NM
Mentor: Zhaofang Wen
Designed and implemented an extension of C++ supporting SIMD-style parallel programming. Conducted experiments to evaluate the performance benefits of the language on problems with large numbers of fine-grain random accesses. Resulted in publication [13].

Undergraduate Research Assistant (September 1999 — February 2000)
Massachusetts Institute of Technology, Media Laboratory, Cambridge, MA
Integrated a speech synthesis system with an digital interactive agent.

Undergraduate Research Assistant (September 1998 — February 1999)
Massachusetts Institute of Technology, Media Laboratory, Cambridge, MA
Implemented algorithms for image analysis for an image search engine.

Teaching Experience

CIS 410/510 Object Oriented Languages and Systems (2008)
with Professor Yannis Smaragdakis
Delivered lectures on the survey of type systems.

Professional Experience

Principal Software Engineer (May 2001 — May 2003)
Fidelity Investments, Boston, MA
Project lead on the design and development wireless.fidelity.com site, Fidelity's web application built specifically for PDA's.

Software Engineer (July 2000 — April 2003)
ArsDigita, Inc., Cambridge, MA

Software Engineer Summer Intern (June 1999 — August 1999)
GemStar, Inc., Bedford, MA

Publications

Journal Papers

1. Shan Shan Huang, David Zook, and Yannis Smaragdakis. Domain-specific languages and program generation with Meta-AspectJ. ***TOSEM**: ACM Transactions on Software Engineering and Methodologies*, 18(2):1–32, 2008.
2. Shan Shan Huang, David Zook, and Yannis Smaragdakis. Statically safe program generation with SafeGen. ***SCP**: Science of Computer Programming*. To Appear.

Conference Papers

3. Shan Shan Huang, Amir Hormati, David F. Bacon, and Rodric M. Rabbah. Liquid Metal: Object-oriented programming across the hardware/software boundary. In Jan Vitek, editor, ***ECOOP '08**:*

Proceedings of the 22nd European Conference on Object-Oriented Programming, volume 5142 of *Lecture Notes in Computer Science*, pages 76–103. Springer, 2008. *Acceptance rate: 19%*.

4. Shan Shan Huang and Yannis Smaragdakis. Expressive and safe static reflection with MorphJ. In ***PLDI '08****: ACM SIGPLAN Conference on Programming Language Design and Implementation*, volume 43, pages 79–89, New York, NY, 2008. ACM. *Acceptance rate: 18%*.
5. Shan Shan Huang, David Zook, and Yannis Smaragdakis. Morphing: Safely shaping a class in the image of others. In ***ECOOP '07****: Proceedings of the European Conference on Object-Oriented Programming*, pages 399–424. Springer-Verlag, August 2007. *Acceptance rate: 16%*.
6. Shan Shan Huang, David Zook, and Yannis Smaragdakis. cJ: Enhancing Java with safe type conditions. In ***AOSD '07****: Proceedings of the 6th International Conference on Aspect-Oriented Software Development*, pages 185–198, Vancouver, British Columbia, Canada, 2007. ACM Press. *Acceptance rate: 18%*.
7. Shan Shan Huang, David Zook, and Yannis Smaragdakis. Statically safe program generation with SafeGen. In ***GPCE '05****: Proceedings of the 4th International Conference on Generative Programming and Component Engineering*, LNCS 3676, pages 309–326, Tallin, Estonia, 2005. Springer-Verlag. *Acceptance rate: 29%*.
8. David Zook, Shan Shan Huang, and Yannis Smaragdakis. Generating AspectJ programs with Meta-AspectJ. In ***GPCE '04****: Proceedings of the 3rd International Conference on Generative Programming and Component Engineering*, pages 1–18, Vancouver, British Columbia, Canada, October 2004. Springer-Verlag. *Acceptance rate: 33%.* ***Best Paper Award***.

Invited Conference Papers

9. Yannis Smaragdakis, Shan Shan Huang, and David Zook. Program generators and the tools to make them. In ***PEPM '04****: Proceedings of the 2004 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation*, pages 92–100, New York, NY, 2004. ACM Press.

Refereed Workshop Papers

10. Shan Shan Huang and Yannis Smaragdakis. Morphing Software for Easier Evolution. In Walter Cazzola, Shigeru Chiba, Yvonne Coady, Stéphane Ducasse, Günter Kniesel, Manuel Oriol, and Gunter Saake, editors, ***RAM-SE '07****: Proceedings of ECOOP'2007 Workshop on Reflection, AOP and Meta-Data for Software Evolution*, pages 71–80, Berlin, Germany, July 2007.

Refereed Short Papers / Posters

11. Shan Shan Huang and Yannis Smaragdakis. Building scalable libraries with cJ. In ***ICSE COMPANION '07****: Companion to the proceedings of the 29th International Conference on Software Engineering*, pages 45–46, Washington, DC, USA, 2007. IEEE Computer Society.
12. Shan Shan Huang and Yannis Smaragdakis. Easy language extension with Meta-AspectJ. In ***ICSE '06****: Proceedings of International Conference on Software Engineering*, pages 865–868, New York, NY, USA, May 2006. ACM.
13. Jonathan L. Brown, Sue Goudy, Mike Heroux, Shan Shan Huang, and Zhaofang Wen. An evolutionary path towards virtual shared memory with random access. In ***SPAA '06****: Symposium on Parallelism in Algorithms and Architectures*, 2005.

Service

Program Committees

- PEPM ’08: 2008 Workshop on Partial Evaluation and Program Manipulation.

Conference Co-reviewing

- PLDI ’09: 2009 Conference on Programming Languages Design and Implementation.
- CC ’09: 12th International Conference on Compiler Construction.
- GPCE ’08: 7th Conference on Generative Programming and Component Engineering.
- FSE ’08: Sixteenth International Symposium on Foundations of Software Engineering.
- OOPSLA ’08: 2008 International Conference on Object-Oriented Programming, Systems, Languages, and Applications.
- ISSTA ’08: 2008 International Symposium on Software Testing and Analysis.
- PLDI ’08: 2008 Conference on Programming Languages Design and Implementation.
- POPL ’08: 35th Symposium on Principles of Programming Languages.
- OOPSLA ’07: 22nd Conference on Object-Oriented Programming Systems and Applications.
- FSE ’06: 14th Symposium on Foundations of Software Engineering.
- GPCE ’05: 4th Conference on Generative Programming and Component Engineering.

Journal Reviewing

- TOPLAS: Transactions on Programming Languages and Systems.
- SCP: Science of Computer Programming: Special Edition on Program Generation.

On-Campus Committees

- Organizer of Graduate Research Forum, University of Oregon, 2007 — 2008.
- Faculty Recruiting Committee Student Representative, Georgia Institute of Technology, 2006.
- Ph.D. Recruiting Chair, College of Computing, Georgia Institute of Technology, 2003 — 2005.
- Women@CC Membership Chair, College of Computing, Georgia Institute of Technology, 2003.

Invited Talks

“Disciplined Meta-Programming for Object-Oriented Languages”, January 2008, Delft University of Technology, Delft, The Netherlands.

Professional Membership

Association for Computing Machinery (SIGSOFT, SIGPLAN).

Institute of Electrical and Electronic Engineers.

Software

MorphJ: http://code.google.com/p/morphing/
MorphJ is an extension of Java that supports “morphing”. Morphing allows code to be declared by statically reflecting over and pattern-matching on the methods or fields of other types—including unknown type parameters. For instance, one can declare a MorphJ generic class `SynchronizeMe<X>` such that, for every method of `X`, `SynchronizeMe<X>` declares a method with the same signature, but inserts synchronization code in each method. MorphJ is the only language to reach this level of expressiveness while maintaining the guarantee of separate type-checking—a MorphJ generic class is type-checked independently of its uses to guarantee that it is well-typed for all possible instantiations.

cJ: http://www-static.cc.gatech.edu/∼ssh/cj/
cJ is an extension of Java that allows fields and methods of a class or interface to be provided only under some static subtyping conditions. cJ allows the statically type-safe expression of a program’s conditional features—a disciplined implementation `#ifdef`, where not only the consistency of the definition and use of features is statically checked, conditions are also much more expressive. cJ is shown to be useful in a concise and type-safe reimplementation of the Java Collections Framework, the standard data structures library for Java, as well as for applications in the Feature-Oriented Programming domain.

Meta-AspectJ: http://www-static.cc.gatech.edu/maj/
Meta-AspectJ is a language tool for generating AspectJ (and, by extension, Java). Meta-AspectJ can be used to implement Domain-Specific Languages using AspectJ as a bytecode transformation backend, or to enhance AspectJ itself with more expressive pointcuts and introductions.

References

Professor Yannis Smaragdakis
email: *yannis@cs.umass.edu*
phone: +1 (413)577-0233
Department of Computer Science
University of Massachusetts, Amherst
140 Governors Drive
Amherst, MA, USA 01003

Professor Todd Millstein
email: *todd@cs.ucla.edu*
phone: +1 (310)825-5942
Department of Computer Science
University of California, Los Angeles
4531D Boelter Hall
Los Angeles, CA, USA 90095

Dr. David Bacon
email: *dfb@watson.ibm.com*
phone: +1 (914)784-7811
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY, USA 10532

Professor Oege de Moor
email: *oege.de.moor@comlab.ox.ac.uk*
phone: +44 1865 273878
Computing Laboratory
Oxford University
Room 005, Wolfson Building, Parks Road
Oxford OX1 3QD, United Kingdom

Personal Data

Citizenship: United States of America

November 30, 2008