

John Tang Boyland

November 2013

UWM Address:
Department of Electrical Engineering
and Computer Science
University of Wisconsin–Milwaukee
Milwaukee, WI 53201
(414) 229–6986
boyland@uwm.edu

Sabbatical Address:
ETH Zurich, CAB F 53.1
Universitätstrasse 6
CH-8092 Zurich
Switzerland
+41 (044) 632 94 39

URL: <http://www.cs.uwm.edu/~boyland>

Education Ph.D. in Computer Science, 1996, University of California, Berkeley.
Dissertation Title: “Descriptive Composition of Compiler Components,” supervised by Prof. S. L. Graham.

M.S. in Computer Science, 1989, University of California, Berkeley.
Report Title: “Sequencing without Sublists.”

B.S. in Computer Science and Mathematics, 1987, University of California, Davis.
Graduated with highest honors.

Experience University of Wisconsin–Milwaukee.
Professor, 2012–present.
Associate Professor, 2004–2012.
Assistant Professor, 1998–2004.

Carnegie Mellon University.
Post-Doctoral Fellow, 1996–1998.

Committees University of Wisconsin–Milwaukee:

campus-wide APCC: Academic Program and Curriculum Committee, 2003–2006, 2007–2013. Academic Planning and Budget Committee, 2010–2013. APCC Chair: 2010–2013. Ad hoc working group on strategic planning, 2012–13.

college APC: Academic Planning Committee, 2002–2008, Graduate Program Subcommittee, 2008–2013. GPSC Chair: 2010–2013. various search and screen committees, including “biomedical engineering” (2008–09).

department Graduate Program Representative, 2007–2010.
Graduate Admissions Committee, 1998–2003. GAC Chair 2007–2010.
Undergraduate Curriculum Committee 2003–2007.

Grants Pending John Boyland and Tian Zhao. *Proof Engineering: Applying Software Practice to Proofs*, \$1M over four years. Submitted to NSF November 2013.

Grants Awarded John Boyland. *Modular Static Checking of Software Design Intent Using Permissions*. \$201,000 (summer salary and graduate student support), NSF. 9/2007–8/2011.

John Boyland. *Static Analysis of Software for Reliable Computing* \$30,000 (graduate student support), subcontract through CMU from NASA, 2005–06.

John Boyland. *Analysis for High-Dependability Computing*. \$200,000 (summer salary, graduate student support), Subcontract through CMU from NASA. Awarded December 2001–04.

John Boyland. *CAREER: Analysis for Evolution of Modular Software*. \$299,676 (summer salary, graduate student support). NSF. Awarded June 2000 for four years.

John Boyland. *FOUR-A: Agent Adaptation And Assurance*. \$152,000 (buy-out and summer salary). Subcontract through CMU from DARPA. Awarded July 1999 for three years.

Ethan Munson and John Boyland. *Tools for Evolving Java Programs and Software Documents*. \$44,320 (equipment). Sun Microsystems. Awarded December 1998.

John Boyland. *Adaptation and Commitment Technology for Evolutionary Software Development*. \$28,294 (instructional buy-out). Subcontract through CMU from DARPA. Awarded October 1998 for one year.

Professional Activities Reviewed manuscripts for over 20 workshops and conferences in the last five years, and also for several journals including J. of Science of Computer Programming, New Generation Computing, Concurrency Pract. and Exp., Journal of ACM, Trans. on Prog. Lang. and Syst. Reviewed books for MIT Press (2) and Morgan Kaufmann Publishers (complete manuscript). Served/serving on following recent program committees:

ECOOP 2013 European Conference on Object-Oriented Programming

FOOL 2012 Foundation of Object-Oriented Languages (**chair**)

FTfJP 2012 Formal techniques for Java-like Programs

ECOOP 2012 European Conference on Object-Oriented Programming (Beijing)

OOPSLA 2011 Object-Oriented Programming: Software, Languages and Applications.

SLE 2011 Software Language Engineering

LDTA 2011 Language Descriptions, Tools and Applications

CATS 2011 17th Computing: the Australasian Theory Symposium

SLE 2010 Software Language Engineering

LDTA 2010 Language Descriptions, Tools and Applications

OOPSLA 2009 Object-Oriented Programming: Software, Languages and Applications.

SLE 2009 Software Language Engineering

FTfJP 2009 Formal techniques for Java-like Programs

ECOOP 2009 European Conference on Object-Oriented Programming

ESOP 2009 European Symposium on Programming.

IWACO 2008 International Workshop on Aliasing, Confinement and Ownership

Guest Professor at ETH Zürich for academic year 2013-14.

Taught a summer course at Nanjing University of Science and Technology, 2012.

Visiting researcher and instructor at Nanjing University (2006–07). Invited talks given at Zhenjiang University (2006), Wuxi University (2007). Invited to speak at the POP (Principles of Programming) Seminar at Carnegie Mellon, Fall 2003.

**Current
Papers**

John Boyland and Tian Zhao. *Tuples for LF: Variable Arity Functions*. 20 pages. Submitted October 2013 to ESOP 2014 (European Symposium on Programming).

**Refereed
Publications**

John Boyland. *Fractional permissions*. In “Aliasing in Object-Oriented Programming,” Clarke, Noble and Wrigstad (eds.), LNCS 7850, 2013. pages 270–288.

John Boyland and Chris Sun. *Proving the Correctness of Fractional Permissions for a Java-like Kernel Language*, presented at FOOL 2011 (co-located with SPLASH 2011).

John Boyland. *Semantics of fractional permissions with nesting*, *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 32, 6 (August 2010), 35 pages. (originally published 12/2007 as a technical report.)

John Boyland. *Generating Bijections between HOAS and the Natural Numbers*. LFMTTP: Logical Frameworks and Meta-languages: Theory and Practice. July 2010.

John Boyland, William Retert and Yang Zhao. *Comprehending Annotations on Object-Oriented Programs using Fractional Permissions*. ECOOP 2009 Workshop on Aliasing, Confinement and Ownership, July 2009.

John Boyland and Daniel Spiewak. *ScalaBison: A recursive ascent-descent parser generator*. ETAPS 2009 workshop on Language Descriptions Tools and Applications, March 2009.

Mohamed ElBendary and John Boyland. *Enhancing Base-Code Protection in Aspect-Oriented Programs*. In *Proceedings of FOAL 2009, Workshop on Foundations of Aspect-Oriented Languages*, pp. 19–24, March 2009.

John Boyland. *An Operational Semantics including “Volatile” for Safe Concurrency*, *Journal of Object Technology*, vol. 8, no. 4, June 2009, pp. 33-53. Originally presented at ECOOP 2008 workshop on Formal Techniques for Java-like Programs, July 2008.

John Boyland. *The Non-Linearity of volatile in Java*. International Workshop on Aliasing, Confinement and Ownership (IWACO '08). July 2008.

Yang Zhao and John Boyland. *A Fundamental Permission Interpretation for Ownership Types*. 2nd IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2008), June 2008.

John Boyland, William Retert, Yang Zhao. *Iterators can be Independent “from” Their Collections*. International Workshop on Aliasing, Confinement and Ownership in object-oriented programming. July 2007.

John Boyland. *Why we should not add readonly to Java (yet)* *Journal of Object Technology*, vol. 5, no. 5, June 2006, pp. 5-29. Originally presented at ECOOP 2005 Workshop on Formal Techniques for Java-like Programs

John Tang Boyland. *Remote Attribute Grammars*. *Journal of the ACM*, 52, 4 (July 2005) 627–687.

Tien Nguyen, Ethan V. Munson, John Tang Boyland. *An infrastructure for development of object-oriented, multi-level configuration management services* In *ICSE 2005, Proceedings of the 27th International Conference on Software Engineering*, 2005 pp. 215–224.

John Boyland and William Retert. *Connecting Effects and Uniqueness with Adoption*. In *Proceedings of ACM Principles of Programming Languages*, 2005 (San Diego, CA), pages 283–295.

Tien Nguyen, Ethan V. Munson, John Tang Boyland. *The Molhado Hypertext Versioning System*. In *Hypertext 2004, Proceedings of the 15th ACM Conference on Hypertext and Hypermedia* pp. 185–194.

Tien Nguyen, Ethan V. Munson, John Boyland. *Configuration management in a hypermedia-based software development environment*. In *Hypertext 2003, Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, page 194–195, August 2003.

John Boyland. *Connecting Effects and Uniqueness with Adoption*. In *International Workshop on Aliasing, Confinement and Ownership in object-oriented programming*. Co-located with ECOOP 2003, Darmstadt, Germany, July 2003.

John Boyland. *Checking Interference with Fractional Permissions*. In *SAS 2003—Static Analysis: 10th International Symposium* (San Diego, June 2003), Volume 2694 of *Lecture Notes in Computer Science*, Springer Verlag, 2003, pages 55-72.

William Retert and John Boyland. *Interprocedural Analysis for JVMML verification*. In *Workshop on Formal Techniques for Java-Like Programs*, co-located with ECOOP 2002, Málaga, Spain, June 2002.

John Boyland. *Incremental Evaluators for Remote Attribute Grammars*. *Electronic Notes in Theoretical Computer Science*, 65(3), 2002. Presented at LDTA (Language Descriptions, Tools and Applications) 2002, co-located with ETAPS 2002, Grenoble, France, April 2002.

John Boyland. *The Interdependence of Effects and Uniqueness*. In *3rd Workshop on Formal Techniques for Java Programs*, co-located with ECOOP 2001, Budapest, June 2001.

John Boyland, James Noble and William Retert. *Capabilities for Sharing: A Generalization of Uniqueness and Read-Only*. In *ECOOP 2001—Object-Oriented Programming, 15th European Conference*, pages 2–27. Volume 2072 of *Lecture Notes in Computer Science*, Springer Verlag, 2001.

John Boyland. *Alias Burying: Unique Variables Without Destructive Reads*. *Journal of Software Practice and Experience*, 31(6):533–553, May 2001.

Aaron Greenhouse and John Boyland. *An Object-Oriented Effects System*. In *ECOOP '99—Object-Oriented Programming, 13th European Conference*, pages 205-229. Volume 1628 of *Lecture Notes in Computer Science*, Springer Verlag, 1999.

John Boyland. Analyzing direct non-local dependencies in attribute grammars. In *CC '98: Compiler Construction, 7th International Conference*, pages 31-49. Volume 1383 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, 1998.

Edwin C. Chan, John T. Boyland, and William L. Scherlis. Promises: Limited specifications for analysis and manipulation. In *Proc. IEEE International Conference on Software Engineering (ICSE '98)*, Kyoto, Japan, April 19–25, pages 167–176.

John Tang Boyland and Giuseppe Castagna. Parasitic methods: An implementation of multi-methods for Java. In *Conference Proceedings of OOPSLA'97 – Object Oriented Programming Systems, Languages and Applications*, pages 66–76, October 1997.

Manuel Fähndrich and John Boyland. Statically checkable pattern abstractions. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP '97)*, pages 75–84, June 1997.

John Boyland. *Descriptive composition of compiler components*. PhD dissertation, December 1996.

John Tang Boyland and Giuseppe Castagna. Type-safe compilation of covariant specialization: A practical case. In *ECOOP '96 — Object-Oriented Programming (10th European Conference)*, volume 1098 of *Lecture Notes in Computer Science*, pages 3–25. Springer, July 1996.

John Tang Boyland. Conditional attribute grammars. *ACM Transactions on Programming Languages and Systems*, 18(1):73–108, January 1996.

John Boyland and Susan L. Graham. Composing tree attributions. In *Conference Record of the Twenty-first Annual ACM SIGACT/SIGPLAN Symposium on Principles of Programming Languages*, pages 375–388, January 1994.

John Boyland, Charles Farnum, and Susan L. Graham. Attributed transformational code generation for dynamic compilers. In *Code Generation - Concepts, Tools, Techniques. Workshops in Computer Science*, pages 227–254. Springer, 1992.

Other Publications

Giuseppe Castagna and John Boyland, eds. Special Issue for ESOP 2009 papers. *Theoretical Computer Science*, 411(51–52), December 2010.

John Boyland, ed. *Formal Techniques for Java-like Programs, 2007*. ECOOP 2007 Workshop Proceedings, Nanjing University Technical Report. 2007.

Courses

CompSci 351: Data Structures and Algorithms (formerly CS 252). I have completely redesigned the course to use Java instead of C++.

CompSci 431: Programming Languages Concepts. I redesigned the middle portion of this class in Spring 2011 to use Squeak rather than Java to showcase Object-Oriented features.

CompSci 552: Object-Oriented Programming. This course teaches software design architecture in the context of a semester-long project.

CompSci 654: Introduction to Compilers. Redesigned in 2009 to use Scala rather than C++ as implementation language. Recently redesigned to be joint with CS 754: Compiler Construction with a separate lab section for programming.

CompSci 732: Type Theory. Redesigned after first sabbatical year (2007) to use a machine-checking proof system for all proofs.

CompSci 854: Advanced Compiler Techniques. (Formerly CS 754 before that course was merged with CS 654.) Redesigned Fall 2010 to use Scala.

CompSci 838: Program Analysis. Redesigned over first sabbatical year and used with students at Nanjing University, China.

Students

Pho Zanakan: master's student. Completed May 2000.

Kwankamol Nongpong: Ph.D. student, supported through a research assistantship. Completed December 2012.

William Retert: Ph.D. student, supported through a research assistantship. Thesis title: Implementing permission analysis. Completed May 2009.

Mohamed ElBendary: Ph.D. student. Thesis title: Enhancing modularity in aspect-oriented software systems. Completed December 2008.

Daniel Graves: master's student. Completed December 2004.

Yang Zhao: Ph.D. student, supported through research assistantship. Thesis title: Concurrency Analysis Based on Fractional Permission System. Completed December 2007.

Brandon Konop: master's student. Completed May 2006.

Chao (Chris) Sun: Ph.D. student, supported through a research assistantship. Achieved dissertator status, August 2011. Anticipated completion 2014.

Lidia Bonilla: Ph.D. student. Thesis title: Static Analysis of Class Invariants in Java Programs. Completed May 2011.

Michael Welch: master's student. Completed May 2011.

Jonathan Bake: master's student. Completed August 2011.

Drew Douglas: master's student. Completed May 2012.

Other Service

Hosted CompSci graduate students and faculty at home (2009,2010,2011,2012).

Faculty adviser for six student groups, including IEEE-CS.

Hosted "Tour of Milwaukee" for international students, yearly 2007–11.

Taught "Scratch" programming to upper elementary pupils (four hour-long sessions with 12 children), 2008.

Organizer of Christian Faculty/Staff Network at UWM, 2007–present.

University Christian Ministries, Board membership, 2000–2006, 2007–2013.