Gary Todd Leavens

University of Central Florida, Harris Center
4000 Central Florida Blvd., Orlando, FL 32816-2362 USA
phone: (407)823-4758 e-mail: leavens@eecs.ucf.edu
URL: http://www.eecs.ucf.edu/~leavens/

Education

Massachusetts Institute of Technology, Cambridge, Mass. **Sept. 1982–Dec. 1988**

Doctor of Philosophy in Computer Science, received in June 1989. Thesis title: *Verifying Object-Oriented Programs that use Subtypes*. Thesis supervisor: Professor William E. Weihl. Minor in mathematics.

University of Southern California, Los Angeles, Calif. **Sept.1979–June 1980**

Master of Science in Computer Science, received in June, 1980.

The University of Michigan, Ann Arbor, Mich. **Sept. 1975–April 1979**

Bachelor of Science, with High Distinction and Honors in Computer and Communication Sciences, received in December, 1978. Spent the Spring term, 1979 in the graduate school.

Research Interests

Programming and specification language design and semantics, formal methods (program specification and verification), aspect-oriented languages, object-oriented languages, software security, information assurance, distributed languages, type theory, programming methodology, software engineering, computer science education.

Refereed Journal Publications

Submitted Research Papers

John Hatcliff, Gary T. Leavens, K. Rustan M. Leino, Peter Müller, and Matthew Parkinson. Behavioral Interface Specification Languages. Under revision for *ACM Computing Surveys*. 68 pages.

Published Research Papers

Gary T. Leavens, K. Rustan M. Leino, and Peter Müller. Specification and verification challenges for sequential object-oriented programs. *Formal Aspects of Computing*, **19**(2):159–189, June 2007. http://dx.doi.org/10.1007/s00165-007-0026-7

Curtis Clifton and Gary T. Leavens. MiniMAO$_1$: Investigating the Semantics of Proceed. *Science of Computer Programming*, **63**(3):321–374, December 2006. http://dx.doi/10.1016/j.scico.2006.02.009

Peter Müller, Arnd Poetzsch-Heffter, and Gary T. Leavens. Modular Invariants for Layered Object Structures. *Science of Computer Programming*, **62**(3):253–286, October 2006. http://dx.doi.org/10.1016/j.scico.2006.03.001

Curtis Clifton, Todd Millstein, Gary T. Leavens, and Craig Chambers. MultiJava: Design Rationale, Compiler Implementation, and Applications. *ACM Transactions on Programming Languages and Systems*, **28**(3):517–575, May 2006. http://doi.acm.org/10.1145/1133651.1133655

Yoonsik Cheon, Gary T. Leavens, Murali Sitaraman, and Stephen Edwards. Model Variables: Cleanly Supporting Abstraction in Design By Contract. *Software—Practice & Experience*, **35**(6):583–599, May 2005. http://dx.doi.org/10.1002/spe.649

Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. *Science of Computer Programming*, **55**(1-3):185–208, March 2005. (Special issue, "Formal Methods for Components

and Objects: Pragmatic aspects and applications.") http://dx.doi.org/10.1007/b14033

Lilian Burdy, Yoonsik Cheon, David R. Cok, Michael D. Ernst, Joseph R. Kiniry, Gary T. Leavens, K. Rustan M. Leino, and Erik Poll. An Overview of JML Tools and Applications. *International Journal on Software Tools for Technology Transfer*, **7**(3):212–232, Springer-Verlag, June, 2005. http://dx.doi.org/10.1007/s10009-004-0167-4

Peter Müller, Arnd Poetzsch-Heffter, and Gary T. Leavens. Modular specification of frame properties in JML. *Concurrency and Computation: Practice and Experience*, **15**(2):117–154, February 2003. http://dx.doi.org/10.1002/cpe.713

Tim Wahls, Gary T. Leavens, and Albert L. Baker. Executing Formal Specifications with Concurrent Constraint Programming. *Automated Software Engineering*, **7**(4):315–343, December, 2000. http://dx.doi.org/10.1023/A:1026554217992

Gary T. Leavens and Don Pigozzi. A Complete Algebraic Characterization of Behavioral Subtyping. *Acta Informatica*, **36**(8):617–663, March 2000. http://dx.doi.org/10.1007/s002360050168

Gary T. Leavens and Jeannette M. Wing. Protective Interface Specifications. *Formal Aspects of Computing*, **10**:59–75, 1998. http://dx.doi.org/10.1007/PL00003926

Gary T. Leavens and Don Pigozzi. The Behavior-Realization Adjunction and Generalized Homomorphic Relations. *Theoretical Computer Science*, **177**:183–216, 1997. http://dx.doi.org/10.1016/S0304-3975(97)87172-1

Steven Jenkins and Gary T. Leavens. Polymorphic Type-Checking in Scheme. *Computer Languages*, **22**(4):215–233, 1996. http://dx.doi.org/10.1016/S0096-0551(97)00002-7

Kim Bruce, Luca Cardelli, Giuseppe Castagna, The Hopkins Objects Group, Gary T. Leavens, and Benjamin Pierce. On Binary Methods. *Theory and Practice of Object Systems*, **1**(3):221–242, 1995.

Craig Chambers and Gary T. Leavens. Type Checking and Modules for Multimethods. *ACM Transactions on Programming Languages and Systems*, **17**(6):805-843, November 1995. http://doi.acm.org/10.1145/218570.218571

Gary T. Leavens and William E. Weihl. Specification and Verification of Object-Oriented Programs Using Supertype Abstraction. *Acta Informatica*, **32**(8):705–778, November 1995. http://dx.doi.org/10.1007/BF01178658

Yoonsik Cheon and Gary T. Leavens. The Larch/Smalltalk Interface Specification Language. *ACM Transactions on Software Engineering and Methodology*, **3**(3):221-253, July 1994. http://doi.acm.org/10.1145/196092.195325

Yoonsik Cheon and Gary T. Leavens. A Quick Overview of Larch/C++. *Journal of Object-Oriented Programming*, **7**(6):39–49, October 1994.

Gary T. Leavens and Mike Vermeulen. $3x + 1$ Search Programs. *Computers and Mathematics with Applications*, **24**(11):79–99, December 1992. http://dx.doi.org/10.1016/0898-1221(92)90034-F

Gary T. Leavens. Modular Specification and Verification of Object-Oriented Programs. *IEEE Software*, **8**(4):72–80, July 1991. http://dx.doi.org/10.1109/52.300040

Published Teaching Papers

Gary T. Leavens, Albert L. Baker, Vasant Honavar, Steven Lavalle, and Gurpur Prabhu. Programming is Writing: Why Student Programs must be Carefully Read. *Mathematics and Computer Education*, **32**(3):284–295, Fall 1998.

Gary T. Leavens. A Physical Example for Teaching Curried Functions. *Mathematics and Computer Education*, **30**(1):51-60, Winter 1996.

Gary T. Leavens. Aiding Self-motivation with Readings in Introductory Computing. *Mathematics and Computer Education*, **29**(2):124–133, 1995.

Refereed Conference Publications

Published Research Papers

Henrique Rebêlo, Ricardo Lima, Márcio Cornélio, Gary T. Leavens, Alexandre Mota, and César Oliveira. Optimizing JML Features Compilation in Ajmlc Using Aspect-Oriented Refactorings. In *XIII Brazilian Symposium on Programming Languages (SBLP 2009)*, Gramado-RS, Brazil, August 19-21, 2009, pp 117–130. (12 accepted / 33 submitted)
Also School of EECS, University of Central Florida, Technical Report CS-TR-09-05, April 2009.

Hridesh Rajan, Jia Tao, Steve Shaner, and Gary T. Leavens. Tisa: A Language Design and Modular Verification Technique for Temporal Policies in Web Services. In Giuseppe Castagna (ed.), *Programming Langauges and Systems: 18th European Symposium on Programming, ESOP 2009*, York, United Kingdom. Volume 5502 of *Lecture Notes in Computer Science*, Springer-Verlag, March 2009, pages 333-347. (25 selected / 130 submitted) `http://dx.doi.org/10.1007/978-3-642-00590-9_24`

Yoonsik Cheon, Antonio Cortes, Martine Ceberio, and Gary T. Leavens. Integrating Random Testing with Constraints for Improved Efficiency and Diversity. In *Proceedings of SEKE 2008: The 20th International Conference on Software Engineering and Knowledge Engineering*, July 2008, San Francisco, CA, pages 861-866. (48% accepted)

Hridesh Rajan and Gary T. Leavens. Ptolemy: A Language with Quantified, Typed Events. In Jan Vitek (ed.), *ECOOP 2008 – Object-Oriented Programming: 22nd European Conference*, Paphos, Cyprus, pp. 155–179. Volume 5142 of *Lecture Notes in Computer Science*, Springer-Verlag, July 2008. (27 selected / 138 submitted) `http://dx.doi.org/10.1007/978-3-540-70592-5_8`

Gary T. Leavens and Curtis Clifton. Lessons from the JML Project. In Bertrand Meyer and Jim Woodcock (eds.), *Verified Software: Theories, Tools, Experiments*, Zurich, Switzerland, pp. 134-143. Volume 4171 of *Lecture Notes in Computer Science*, Springer-Verlag, 2008.
`http://dx.doi.org/10.1007/978-3-540-69149-5_15`

Steve M. Shaner, Gary T. Leavens, and David A. Naumann. Modular Verification of Higher-Order Methods with Mandatory Calls Specified by Model Programs. In *ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2007)*, Montreal Canada, pp. 351–367, October 2007. Winner of the "Best Student Paper" award. (33 accepted / 156 submitted) `http://doi.acm.org/10.1145/1297027.1297053`

Curtis Clifton, Gary T. Leavens, and James Noble. MAO: Ownership and Effects for more Effective Reasoning about Aspects. In Erik Ernst (ed.), *European Conference on Object-Oriented Programming (ECOOP)* 2007, Berlin, Germany, pp. 451–475. Volume 4609 of *Lecture Notes in Computer Science*, Springer-Verlag, 2007. (25 accepted / 160 submitted) `http://dx.doi.org/10.1007/978-3-540-73589-2_22`

Gary T. Leavens and Peter Müller. Information Hiding and Visibility in Interface Specifications. In *International Conference on Software Engineering (ICSE)*, Minneapolis, Minnesota, May 2007, pages 385–395. (15% accepted.) `http://dx.doi.org/10.1109/ICSE.2007.44`

Patrice Chalin, Joseph R. Kiniry, Gary T. Leavens, and Erik Poll. Beyond Assertions: Advanced Specification and Verification with JML and ESC/Java2. In *Formal Methods for Components and Objects (FMCO) 2005, Revised Lectures*, pages 342-363. Volume 4111 of *Lecture Notes in Computer Science*, Springer-Verlag, 2006. `http://dx.doi.org/10.1007/11804192_16`

Yoonsik Cheon and Gary T. Leavens. A Contextual Interpretation of Undefinedness for Runtime Assertion Checking. In *AADEBUG 2005 Sixth International Symposium On Automated And*

Analysis-Driven Debugging, Monterey, California, September, 2005, pages 149-158. (36% accepted.) http://doi.acm.org/10.1145/1085130.1085150

Edwin Rodríguez, Matthew Dwyer, Cormac Flanagan, John Hatcliff, Gary T. Leavens, and Robby. Extending JML for Modular Specification and Verification of Multi-Threaded Programs. In Andrew P. Black (ed.), *ECOOP 2005 — Object-Oriented Programming 19th European Conference, Glasgow, UK*, pages 551–576. Volume 3586 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2005. (14% accepted) http://dx.doi.org/10.1007/11531142_24

Yoonsik Cheon, Yoshiki Hayashi, and Gary T. Leavens. A Thought on Specification Reflection. N. Callaos, W. Lesso, and B. Sanchez, editors, *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, July 18-21, 2004, Orlando, Florida, USA, Volume II, Computing Techniques, pages 485-490, July 2004. (53% accepted)

Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. In Frank S. de Boer, Marcello M. Bonsangue, Susanne Graf, and Willem-Paul de Roever (eds.), *Formal Methods for Components and Objects: First International Symposium, FMCO 2002, Lieden, The Netherlands, November 2002, Revised Lectures*, pages 262–284. Volume 2852 of *Lecture Notes in Computer Science*, Springer-Verlag, 2003. (Invited paper.) http://dx.doi.org/10.1007/b14033

Yoonsik Cheon and Gary T. Leavens. A Runtime Assertion Checker for the Java Modeling Language (JML). In Hamid R. Arabnia and Youngsong Mun (eds.), *International Conference on Software Engineering Research and Practice (SERP '02), Las Vegas, Nevada.* CSREA Press, June 2002, pages 322-328. (28% accepted)

Yoonsik Cheon and Gary T. Leavens. A Simple and Practical Approach to Unit Testing: The JML and JUnit Way. In Boris Magnusson (ed.), *ECOOP 2002 — Object-Oriented Programming, 16th European Conference, Málaga, Spain, June 2002, Proceedings.* Volume 2374 of *Lecture Notes in Computer Science.* Springer-Verlag, 2002, pages 231-255. (24 accepted / 96 submitted)
http://dx.doi.org/10.1007/3-540-47993-7_10

Tim Wahls and Gary T. Leavens. Formal Semantics of an Algorithm for Translating Model-based Specifications to Concurrent Constraint Programs. In *Proceedings of the 16th ACM Symposium on Applied Computing*, Las Vegas, Nevada, March 2001, pp. 567–575. (118 accepted / 224 submitted)
http://doi.acm.org/10.1145/372202.372465

Clyde Ruby and Gary T. Leavens. Safely Creating Correct Subclasses without Seeing Superclass Code. In *OOPSLA 2000 Conference Proceedings*, pp. 208–228. Volume 35, number 10 of *ACM SIGPLAN Notices*, October 2000. (26 accepted / 142 submitted)
http://doi.acm.org/10.1145/353171.353186

Curtis Clifton, Gary T. Leavens, Craig Chambers, and Todd Millstein. MultiJava: Modular Open Classes and Symmetric Multiple Dispatch for Java. In *OOPSLA 2000 Conference Proceedings*, pp. 130–145. Volume 35, number 10 of *ACM SIGPLAN Notices*, October 2000. (26 accepted / 142 submitted) http://doi.acm.org/10.1145/354222.353181

Gary T. Leavens and Albert L. Baker. Enhancing the Pre- and Postcondition Technique for More Expressive Specifications. In Jeannette M. Wing, Jim Woodcock, and Jim Davies (editors), *FM'99 — Formal Methods: World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, September 1999, Proceedings*, pages 1087–1106. Volume 1709 of *Lecture Notes in Computer Science*, Springer-Verlag, 1999. (92 accepted / 259 submitted)
http://dx.doi.org/10.1007/3-540-48118-4_8

Gary T. Leavens, Tim Wahls, and Albert L. Baker. Formal Semantics for SA Style Data Flow Diagram Specification Languages. In *ACM SAC'99 — 1999 ACM Symposium on Applied Computing*, San Antonio, February/March 1999, pages 526–532. (104 accepted / 156 submitted)

http://doi.acm.org/10.1145/298151.298433

Gary T. Leavens and Todd D. Millstein. Multiple Dispatch as Dispatch on Tuples. In *OOPSLA '98 Conference Proceedings*, pages 374–387, Volume 33, number 10 of *ACM SIGPLAN Notices*, October 1998. (27 accepted / 145 submitted) http://doi.acm.org/10.1145/286936.286977

Gary T. Leavens and Jeannette M. Wing. Protective Interface Specifications. In Michel Bidoit and Max Dauchet (editors), *TAPSOFT '97: Theory and Practice of Software Development, 7th International Joint Conference CAAP/FASE, Lille, France*, pages 520–534. Volume 1214 of *Lecture Notes in Computer Science*, Springer-Verlag, 1997. (23 accepted / 79 submitted to FASE) http://dx.doi.org/10.1007/BFb0030623

Krishna Kishore Dhara and Gary T. Leavens. Forcing Behavioral Subtyping Through Specification Inheritance. In *Proceedings 18th International Conference on Software Engineering (ICSE-18)*, Berlin, Germany, 1996, pages 258–267. (52 accepted / 213 submitted)

Krishna Kishore Dhara and Gary T. Leavens. Weak Behavioral Subtyping for Types with Mutable Objects. In S. Brookes, M. Main, A. Melton, and M. Mislove, editors, *Mathematical Foundations of Programming Semantics, Eleventh Annual Conference*, Preliminary Proceedings, 1995 pages 269-290.
The final version appears in S. Brookes, M. Main, A. Melton, and M. Mislove, editors, *Mathematical Foundations of Programming Semantics, Eleventh Annual Conference*. In Volume 1 of *Electronic Notes in Theoretical Computer Science*, Elsevier, 1995, which is found on-line at http://www.sciencedirect.com/science/journal/15710661.

Craig Chambers and Gary T. Leavens. Typechecking and Modules for Multi-Methods. In *OOPSLA '94 Conference Proceedings*, pages 1–15. Volume 29, number 10 of *ACM SIGPLAN Notices*, October 1994. (29 accepted / 139 submitted) http://doi.acm.org/10.1145/191080.191083

Gary T. Leavens and Don Pigozzi. Typed Homomorphic Relations Extended with Subtypes. In Stephen Brookes, editor, *Mathematical Foundations of Programming Semantics '91*, pages 144–167. Volume 598 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.
http://dx.doi.org/10.1007/3-540-55511-0_7

Gary T. Leavens and William E. Weihl. Reasoning about Object-Oriented Programs that use Subtypes (extended abstract). In Norman Meyrowitz, editor, *OOPSLA ECOOP '90 Proceedings*, pages 212–223. Volume 25, number 10, of *ACM SIGPLAN Notices*, October 1990. (29 accepted / 194 submitted) http://doi.acm.org/10.1145/97945.97970

Invited Research Papers in Workshops

Gary T. Leavens and Don Pigozzi. Class-Based and Algebraic Models of Objects. In Volume 14 of *Electronic Notes in Theoretical Computer Science*, Elsevier, 1999, which is found on-line at http://www.sciencedirect.com/science/journal/15710661. 31 pages.

Gary T. Leavens and Yoonsik Cheon. Preliminary Design of Larch/C++. In U. Martin and J. Wing, editors, *Proc. First International Workshop on Larch, Dedham 1992*, pages 159–184. Workshops in Computing Series. Springer-Verlag, 1993.

Papers in Juried Workshops

Accepted Research Papers to Appear

Tomas Kalibera, Pavel Parizek, Ghaith Haddad, Gary T. Leavens, and Jan Vitek. Challenge Benchmarks for Verification of Real-time Programs. To appear in *ACM SIGPLAN Workshop on Programming Languages meets Program Verification* (PLPV 2010), Madrid, Spain, January 2010.

Published Research Papers

David Cok and Gary T. Leavens. Extensions of the theory of observational purity and a practical

design for JML. In *Seventh International Workshop on Specification and Verification of Component-Based Systems (SAVCBS 2008)*, Atlanta, Georgia, pages 43-50. (Available as School of EECS, UCF, CS-TR-08-07, Nov. 2008.) `http://www.eecs.ucf.edu/SAVCBS/2008/papers/Cok-Leavens.pdf`

Steve Shaner, Hridesh Rajan, and Gary T. Leavens. Model Programs for Preserving Composite Invariants. In *Seventh International Workshop on Specification and Verification of Component-Based Systems (SAVCBS 2008)*, Atlanta, Georgia, pages 95-100. (Available as School of EECS, UCF, CS-TR-08-07, Nov. 2008.)
`http://www.eecs.ucf.edu/SAVCBS/2008/papers/Shaner-Rajan-Leavens.pdf`

Gary T. Leavens and Curtis Clifton. Multiple Concerns in Aspect-Oriented Language Design: A Language Engineering Approach to Balancing Benefits, with Examples. In *SPLAT '07: Proceedings of the 5th workshop on Engineering properties of languages and aspect technologies*, Vancouver, British Columbia, Canada. 5 pages. `http://doi.acm.org/10.1145/1233843.1233849`

Curtis Clifton and Gary T. Leavens. MiniMAO: Investigating the Semantics of Proceed. In Curtis Clifton, Ralf Lämmel, and Gary T. Leavens, editors, *FOAL 2005 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2005*, Chicago, Illinois, pages 57-67. (Available as Department of Computer Science, Iowa State University, TR #05-05, March 2005.)

Lilian Burdy, Yoonsik Cheon, David R. Cok, Michael Ernst, Joe Kiniry, Gary T. Leavens, K. Rustan M. Leino, and Erik Poll. An Overview of JML Tools and Applications. *Eighth International Workshop on Formal Methods for Industrial Critical Systems (FMICS)*, pp. 73–89. *Electronic Notes in Theoretical Computer Science (ENTCS)*, Elsevier, 2003.
`http://www.sciencedirect.com/science/journal/15710661`
Also, Department of Computer Science, University of Nijmegen, technical report NIII-R0309, March 2003.

Curtis Clifton and Gary T. Leavens. Observers and Assistants: A Proposal for Modular Aspect-Oriented Reasoning. In *Foundations of Aspect-Oriented Languages Workshop*, at the *1st International Conference on Aspect-Oriented Software Development*, Enschede, The Netherlands. Also Department of Computer Science, Iowa State University, TR #02-04, March 2002. 12 pages.

Peter Müller, Arnd Poetzsch-Heffter and Gary T. Leavens. Modular Specification of Frame Properties in JML. In the *Formal Techniques for Java Programs 2001* workshop (at *ECOOP 2001*), 2001. Also Department of Computer Science, Iowa State University, TR #01-03, April 2001. 11 pages.

Gary T. Leavens, Albert L. Baker, and Clyde Ruby. JML: a Java Modeling Language. Formal Underpinnings of Java Workshop (at *OOPSLA '98*), October 18, 1998. See `http://www-dse.doc.ic.ac.uk/~sue/oopsla/cfp.html`.

Gary T. Leavens and Don Pigozzi. Towards More Practical Specification Languages for Object-Oriented Software. In *Second US-Brazil Workshop on Formal Foundations of Software Systems, New Orleans, LA*. November 13-16, 1997. 3 pages.

Craig Chambers and Gary T. Leavens. BeCecil, A Core Object-Oriented Language with Block Structure and Multimethods: Semantics and Typing. In *The Fourth International Workshop on Foundations of Object-Oriented Languages, FOOL 4, Paris, France.*
`http://www.cs.williams.edu/~kim/FOOL/FOOL4.html` 49 pages.
(The full version is Department of Computer Science, Iowa State University, TR #96-17a, December, 1996, revised April 1997. 101 pages.)

Gary T. Leavens and Clyde Ruby. Specification Facets for More Precise, Focused Documentation. In *WISR8, the 8th Annual Workshop on Software Reuse, Columbus, Ohio*, March 23–26, 1997, pages Leavens-1–Leavens-5.

Craig Chambers and Gary T. Leavens. Towards Safe Modular Extensible Objects. *OOPSLA '94 Workshop: Subjectivity in Object-Oriented Systems*, pages 1–2.

(Also Department of Computer Science, Iowa State University, TR #94-17a, August 1994, revised September 1994.)

Gary T. Leavens. Inheritance of Interface Specifications (Extended Abstract). In J. M. Wing, editor, *Proceedings of the Workshop on Interface Definition Languages*, pages 129–138. Volume 29, number 8, of *ACM SIGPLAN Notices*, August 1994.

Gary T. Leavens and Yoonsik Cheon. Extending CORBA IDL to Specify Behavior with Larch. In *OOPSLA '93 Workshop Proceedings: Specification of Behavioral Semantics in OO Information Modeling*, pages 77–80.
(Also Department of Computer Science, Iowa State University, TR #93-20, August 1993.)

Published Teaching Papers

Gary T. Leavens. Use Concurrent Programming Models to Motivate Teaching of Programming Languages. In Programming Languages Curriculum Workshop 2008, *ACM SIGPLAN Notices*, **43**(11):93-98, Nov. 2008. `http://doi.acm.org/10.1145/1480828.1480849`

Other Research Publications

Edited Books

Gary T. Leavens and Murali Sitaraman (editors). *Foundations of Component-Based Systems*. Cambridge University Press, New York, NY, 2000.

Invited Columns in Journals

Gary T. Leavens. Not a Number of Floating Point Problems. Invited column in *Journal of Object Technology*, **5**(2):75-83, March-April, 2006.
`http://www.jot.fm/issues/issues_2006_03/column8`

Invited Chapters in Books or Conferences

Gary T. Leavens. JML's Rich, Inherited Specifications for Behavioral Subtypes. Invited keynote paper in Zhiming Liu and He Jifeng (eds), *Formal Methods and Software Engineering: 8th International Conference on Formal Engineering Methods, ICFEM 2006, Macao, China*, pages 2–34. Volume 4260 of *Lecture Notes in Computer Science*, Springer-Verlag, 2006.
`http://dx.doi.org/10.1007/11901433`

Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the Design of JML Accommodates Both Runtime Assertion Checking and Formal Verification. Invited paper in Frank S. de Boer, Marcello M. Bonsangue, Susanne Graf, Willem-Paul de Roever (eds)., *Formal Methods for Components and Objects: First International Symposium, FMCO 2002*, Lieden, The Netherlands, November 2002. Volume 2852 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pages 262-284. `http://dx.doi.org/10.1007/b14033`

Gary T. Leavens and Krishna Kishore Dhara. Concepts of Behavioral Subtyping and a Sketch of their Extension to Component-Based Systems. Invited chapter in Gary T. Leavens and Murali Sitaraman (editors), *Foundations of Component-Based Systems*, Cambridge University Press, 2000. Chapter 6, pages 113–135.

Gary T. Leavens, Albert L. Baker, and Clyde Ruby. JML: a Notation for Detailed Design. Invited chapter in Haim Kilov, Bernhard Rumpe, and Ian Simmonds (editors). *Behavioral specifications of businesses and systems*, Kluwer Academic Publishers, 1999. Chapter 12, pages 175–188.

Gary T. Leavens. Abstract Data Types. Invited article in John G. Webster, editor, *Wiley Encyclopedia of Electrical and Electronics Engineering*, pages 4–13. John Wiley and Sons, 1999.

Gary T. Leavens. An Overview of Larch/C++: Behavioral Specifications for C++ Modules. Invited chapter in Haim Kilov and William Harvey, editors, *Specification of Behavioral Semantics in Object-Oriented Information Modeling*. Kluwer Academic Publishers, 1996. Chapter 8, pages 121–142.
(Also Department of Computer Science, Iowa State University, TR #96-01c, February 1996, revised

March 1996, April 1996, January 1997.)

Unrefereed Publications

Gary T. Leavens, Albert L. Baker, and Clyde Ruby. Preliminary Design of JML: A Behavioral Interface Specification Language for Java. *ACM Software Engineering Notes* **31**(3):1-38, March 2006. http://doi.acm.org/10.1145/1127878.1127884

Tutorials at Conferences

Curtis Clifton, Gary T. Leavens, Robby, Joseph Kiniry, and Erik Poll. JML Tutorial Tutorial at *OOPSLA 2009*.

Joseph Kiniry, Dan Zimmerman, Erik Poll, Gary T. Leavens, and David Cok. Verification-centric Development in Java with JML and ESC/Java2. Tutorial at *ETAPS 2009*. http://kind.ucd.ie/documents/tutorials/etaps09.html

Joseph Kiniry, Dan Zimmerman, Erik Poll, Gary T. Leavens, and David Cok. Verification-centric Development in Java with JML and ESC/Java2. Tutorial at *ETAPS 2008*. http://kind.ucd.ie/documents/tutorials/etaps08.html

Gary T. Leavens. Tutorial on JML, the Java Modeling Language. Invited Tutorial in *ASE '07: Proceedigns of the 22nd ACM/IEEE International Conference on Automated Software Engineering*, Atlanta, Georgia, ACM, November 2007, page 573. http://doi.acm.org/10.1145/1321631.1321747

Gary T. Leavens, Joseph R. Kiniry, and Erik Poll. A JML Tutorial: Modular Specification and Verification of Functional Behavior for Java. Invited tutorial in Werner Damm and Holger Hermanns (eds.) *Computer Aided Verification: 19th International Conference, CAV 2007, Berlin, Germany, July 2007, Proceedings*. Volume 4590 of Lecture Notes in Computer Science, Springer-Verlag, 2007, page 37. http://dx.doi.org/10.1007/978-3-540-73368-3_6

Invited Positions on Panels at Conferences

Dennis de Champeaux, Pierre America, Derek Coleman, Roger Duke, Doug Lea, and Gary T. Leavens. Formal Techniques for OO Software Development. Invited position paper in *OOPSLA '91 Proceedings*, pages 166–170. Volume 26, number 11 of *ACM SIGPLAN Notices*, November 1991.

Reviews

Gary T. Leavens. Programs, Recursion and Unbounded Choice by Wim Hesselink. *SIAM Review*, **36**(1):131–133, March 1994. http://dx.doi.org/10.1137/1036034

Abstracts of Conference Posters (Juried)

Gary T. Leavens, Dimitra Giannakopoulou, and Murali Sitaraman. Specification of Component-Based Systems Workshop. Abstract of an *OOPSLA 2001* poster.

Curtis Clifton, Gary T. Leavens, Craig Chambers, and Todd Millstein. MultiJava: Open Classes and Multiple Dispatch for Java. Abstract of an *OOPSLA 2001* poster.

Gary T. Leavens, K. Rustan M. Leino, Erik Poll, Clyde Ruby, and Bart Jacobs. JML: notations and tools supporting detailed design in Java. Abstract of an *OOPSLA 2000* poster. In *OOPSLA '00 Companion*, pages 105–106, Minneapolis, Minnesota, ACM, October, 2000.

Reports

Eric Allen, Ras Bodik, Kim Bruce, Kathleen Fisher, Stephen Freund, Robert Harper, Chandra Krintz, Shriram Krishnamurthi, Jim Larus, Doug Lea, Gary Leavens, Lori Pollock, Stuart Reges, Martin Rinard, Mark Sheldon, Franklyn Turbak, and Mitchell Wand. SIGPLAN programming language curriculum workshop: Discussion Summaries and recommendations. In *ACM SIGPLAN Notices*, **43**(11):6-29, November 2008. http://doi.acm.org/10.1145/1480828.1480831

John Boyland, Dave Clarke, Gary Leavens, Francesco Logozzo, and Arnd Poetzsch-Heffter. Formal Techniques for Java-Like Programs In *Object-Oriented Technology. ECOOP 2007 Workshop*

Reader, pages 99-107. Volume 4906 of *Lecture Notes in Computer Science*. Springer-Verlag, 2008. http://dx.doi.org/10.1007/978-3-540-78195-0_10

Gary T. Leavens, Jean-Raymond Abrial, Don Batory, Michael Butler, Alessandro Coglio, Kathi Fisler, Eric Hehner, Cliff Jones, Dale Miller, Simon Peyton-Jones, Murali Sitaraman, Douglas R. Smith, and Aaron Stump. Roadmap for Enhanced Languages and Methods to Aid Verification. In *Fifth International Conference on Generative Programming and Component Engineering (GPCE'06)*, October, 2006, pages 221–235. http://doi.acm.org/10.1145/1173706.1173740

Susan Eisenbach, Gary T. Leavens, Peter Müller, Arnd Poetzsch-Heffter, and Erik Poll. Formal Techniques for Java-Like Programs. In Frank Buschmann, Alejandro P. Buschmann, and Mariano A. Cilia (eds.), *Object-Oriented Technology. ECOOP 2003 Workshop Reader*, pages 62-71. Volume 3013 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

Sophia Drossopoulou, Susan Eisenbach, Gary T. Leavens, Arnd Poetzsch-Heffter, and Erik Poll. Formal Techniques for Java-like Programs. In Juan Hernandez and Ana Moreira (eds.), *Object-Oriented Technology. ECOOP 2002 Workshop Reader*, pages 203–210. Volume 2548 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

Gary T. Leavens, Sophia Drossopoulou, Susan Eisenbach, Arnd Poetzsch-Heffter, and Erik Poll. Formal Techniques for Java Programs. In A. Frohner (ed.), *Object-Oriented Technology. ECOOP 2001 Workshop Reader*, pages 30–40. Volume 2323 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

Sophia Drossopoulou, Susan Eisenbach, Bart Jacobs, Gary T. Leavens, Peter Müller, and Arnd Poetzsch-Heffter. Formal Techniques for Java Programs. In Jacques Malenfant and Sabine Moisan and Ana Moreira (eds.), *Object-Oriented Technology. ECOOP 2000 Workshop Reader*, pages 41–54. Volume 1964 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000. http://dx.doi.org/10.1007/3-540-44555-2_4

Kim Bruce, Benjamin Goldberg, Chris Haynes, Gary T. Leavens, and John Mitchell. Proposed knowledge units for programming languages for Curriculum 2001. ACM SIGPLAN Notices, 35(4):29–43, April 2000. http://doi.acm.org/10.1145/346443.346450

B. Jacobs and G. T. Leavens and P. Müller and A. Poetzsch-Heffter. Formal Techniques for Java Programs. In A. Moreira and D. Demeyer (eds.), *Object-Oriented Technology. ECOOP'99 Workshop Reader*. Volume 1743 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

Giuseppe Castagna and Gary T. Leavens. Foundations of Object-Oriented Languages: 2nd Workshop report. *ACM SIGPLAN Notices*, **30**(2):5–11, February, 1995.

Annotated Bibliographies

Gary T. Leavens. Introduction to the Literature on Object-Oriented Design, Programming, and Languages. *OOPS Messenger*, **2**(4):40–53, October 1991.

Gary T. Leavens. Bibliography on Data Types. *ACM SIGPLAN Notices*, **19**(8):41–50, August 1984.

Technical Colloquia

Invited

Gary T. Leavens, "JML's Rich Inherited Specifications for Behavioral Subtypes."

Radboud University Nijmegen, July 3, 2009.

Rose-Hulman Institute of Technology, September 28, 2006.

Gary T. Leavens, "Preventing Cross-Type Aliasing to Promote Weak Behavioral Subtyping"

IFIP Working Group 2.3, Cambridge, Mass., USA, June 12, 2009. Gary T. Leavens, "JML: Expressive Modular Reasoning for Java"

Invited Keynote at the JML "Spec-a-thon", University of Washington, Tacoma, May 26, 2009.

University of Central Florida, April 14, 2007.

The College of William & Mary, April 2, 2007.

Kansas State University, March 29, 2007.

Stanford University, March 5, 2007.

Clemson University, March 2, 2007.

Gary T. Leavens (based on joint work with Curtis C. Clifton, James Noble, Hridesh Rajan), "Introduction to Aspect-Oriented Programming and its Reasoning Problems"

University of North Florida, Jacksonville, FL, February 26, 2009.

IFIP Working Group 2.3, Cambridge, UK, July 21, 2008.

Gary T. Leavens, "Support for Supertype Abstraction in JML"

High Confidence Software and Systems (HCSS) Conference, March 6, 2008.

Gary T. Leavens, "The State of State in JML"

Types, Logics and Semantics for State, Dagstuhl workshop, Germany, February 4, 2008.

Gary T. Leavens (based on joint work with Steve M. Shaner and David A. Naumann), "Modular Verification of Higher-Order Methods in JML"

Types, Logics and Semantics for State, Dagstuhl workshop, Germany, February 7, 2008.

Radboud University Nijmegen, Netherlands, July 6, 2007.

Kansas State University, March 30, 2007.

IFIP Working Group 2.3, Sydney, Australia, January 12, 2007.

Gary T. Leavens, "Discussion of Steimann's Essay 'The Paradoxical Success of Aspect-Oriented Programming'."

OOPSLA, Portland Oregon, October 26, 2006.

Gary T. Leavens (based on joint work with Curtis Clifton and James Noble), "Efficient Reasoning in AspectJ-like Languages using Concern Domains."

Radboud University Nijmegen, Netherlands, July 5, 2007.

Rose-Hulman Institute of Technology, September 29, 2006.

Universidade Federal de Pernambuco, Recife, Brazil, June 14, 2006.

University of Illinois at Urbana-Champaign, Urbana-Champaign, May 31, 2006.

Gary T. Leavens (based on joint work with Peter Müller), "Information Hiding and Visibility in Interface Specifications"

Kansas State University, May 7, 2007.

Gary T. Leavens, "A Research Roadmap for Enhanced Languages and Methods."

The Challenge of Software Verification, Dagstuhl workshop, Germany, June 13, 2006.

Mini Workshop on Verified Software, SRI International, April 1, 2006.

IFIP Working Group 2.3, Brugges, Belgium, March 17, 006.

Gary T. Leavens (based on joint work with Edwin Rodríguez, Matthew Dwyer, Cormac Flanagan, John Hatcliff, and Robby), "Extending JML for Modular Specification and Verification of Multi-Threaded Programs."

Michigan State University, East Lansing, February 2, 2006.

Imperial College, London, September 23, 2005.

FATS Seminar, ETH Zürich, Switzerland, June 15, 2005.

Gary T. Leavens, "JML and its Unit Testing Tool."

ETH Zürich, June 29, 2006 (for Peter Müller's class).

Michigan State University, East Lansing, February 3, 2006.

Memorial University of Newfoundland, St. John's, Newfoundland, Canada, August 9, 2005.

University of Alabama, Tuscaloosa, Alabama, April 8, 2005.

University of Alabama, Birmingham, Alabama, April 7, 2005.

University of Minnesota, Minneapolis, Minnesota, March 30, 2005.

University of Nebraska, Lincoln, November 23, 2004.

University of Texas, El Paso, April 14, 2004.

ETH Zürich, March 18, 2004.

Gary T. Leavens (based on joint work with Peter Müller and Arnd Poetzsch-Heffter), "Why Specification Languages Need Ownership Types."

Dagstuhl workshop *Types for Tools: Applications of Type Theoretic Techniques*, Dagstuhl, Germany, June 20, 2005.

Gary T. Leavens and Peter Müller, "Privacy in JML Specifications."

Tuesday afternoon formal methods club, ETH Zürich, Switzerland, June 14, 2005.

Gary T. Leavens (based on joint work with David Cok), "Adapting Observational Purity to JML."

IFIP WG 2.3, Niagara Falls, Canada, June 9, 2005.

Gary T. Leavens, "Alias-free Formal Parameters."

University of Minnesota, Minneapolis, Minnesota, March 30, 2005.

IFIP WG 2.3, Prato, Italy, September 7, 2004.

Gary T. Leavens, "Advances and Issues in JML."

IFIP WG 2.3, Philadelphia, Pennsylvania, January 9, 2004.

Microsoft Research, Redmond, Washington, November 21, 2003.

First International Symposium on Formal Methods for Components and Objects, Leiden, The Netherlands, November 13, 2002.

Java Verification Workshop 2002, Portland Oregon, January 12, 2002. Yoonsik Cheon and Gary T. Leavens, "A Simple and Practical Approach to Unit Testing: The JML and JUnit Way."

Purdue University, West Lafayette, Indiana, February 2, 2004.

Microsoft Research, Redmond, Washington, November 20, 2003.

Katholieke Universiteit Leuven, Belgium, November 28, 2002.

Virginia Tech, October 18, 2002.

Clemson University, May 31, 2002.

ABC Virtual, Des Moines, February 5, 2002.

Java Verification Workshop 2002, Portland Oregon, January 13, 2002.

Gary T. Leavens, "A Java Modeling Language and also Specification Inheritance."

Department of Computer Science, Purdue University, West Lafayette, February 26, 2002.

Department of Computer Science, University of Wisconsin, Milwaukee, November 17, 2000.

Department of Computer Science, University of Iowa, September 5, 2000.

Department of Computer and Information Science, The Ohio State University, March 26, 2000.

Department of Computer and Information Science, Kansas State University, November 2, 1999.

Gary T. Leavens, "Calling All Arguments: Adding Multiple Dispatch to Object-Oriented Languages."

Vrije Universiteit Brussels, November 29, 2002.

Department of Computer and Information Science, The Ohio State University, March 25, 2000.

Distinguished lecture series, Department of Computer and Information Science, Kansas State University, November 1, 1999.

Gary T. Leavens, Albert L. Baker, and Clyde Ruby. "JML: a Java Modeling Language."

Formal Underpinnings of Java Workshop (at *OOPSLA '98*), October 18, 1998.

Gary T. Leavens, "Applying Larch/C++ to the STL."

Workshop on Generic Programming, Schloss Dagsthul, Germany, April 30, 1998.

Gary T. Leavens, "Java as a Programming Language."

Association for Systems Management, Des Moines, Iowa, April 13, 1998.

Gary T. Leavens, "Forcing Behavioral Subtyping by Specification Inheritance in Larch/C++."

Rockwell International, Cedar-Rapids, Iowa, May 19, 1997.

Department of Computer Science, University of Iowa, Iowa City, Iowa, November 21, 1995.

Department of Computer Science, Concordia University, Montréal, Québec, October 23, 1995.

Computer and Information Science, The Ohio State University, Columbus, Ohio, January 31, 1995.

Gary T. Leavens. "Information Hiding and Modularity in an Object-Oriented Language with Multimethods."

Dept. of Computer Science and Engineering, University of Washington, Seattle, WA, November 21, 1996.

Gary T. Leavens, "Advanced Programming Languages for Engineering Computation."

Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, Nov. 28, 1995.

Gary T. Leavens, "Larch/C++: An Interface Specification Language for C++."

INRS-Telecommunications (Univ. of Québec and Bell Northern Research), Montréal, Canada, December 15, 1994.

Technical University of Munich, Munich, Germany, July 14, 1994.

Imperial College, London, England, July 13, 1994.

British Telecom, Ipswich, England, July 12, 1994.

Mathematical Center (CWI), Amsterdam, Holland, July 11, 1994.

Department of Computer Science and Engineering, University of Washington, January 13, 1994.

Department of Computer Science, Concordia University (Montreal, Quebec), March 8, 1993.

Gary T. Leavens, "Modules for Multi-methods: Information Hiding and Program Composition."

Technical University of Munich, Munich, Germany, July 15, 1994.

Workshop on *Foundations of Object-Oriented Programming Languages*, Paris, France, July 1-2, 1994.

Gary T. Leavens, "Semantics for Object-Oriented Language Design, Specification, and Correctness."

Norand Corp., Cedar Rapids, September 12, 1995.

IBM, Rochester, Minnesota, March 9, 1994.

Gary T. Leavens, "A Model Theory for Subtyping in Imperative OOPLs, Part I."

Workshop on *Foundations of Object-Oriented Languages*, Stanford University, October 17-18, 1993.

Gary T. Leavens, "Reasoning about Object-Oriented Programs that use Subtypes."

Kansas State University, Department of Computing and Information Sciences, September 24, 1991.

UNISYS, September 18, 1990.

Gary T. Leavens, "Formal Reasoning about Object-Oriented Programs."

Keynote talk at the workshop *Reliability Issues in Object-Oriented Programming*, University of Iowa, Department of Computer Science Iowa City, Iowa, July 31, 1990.

Gary T. Leavens, "Behavioral Subtyping as a Reasoning Tool."

Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, Massachusetts, May 23, 1990.

University of Pennsylvania, Department of Computer and Information Science, Philadelphia, Pennsylvania, April 30, 1990.

Hewlett-Packard Laboratories, Software Technology Laboratory, Palo Alto, California, January 16, 1990.

Xerox Palo Alto Research Center, Computer Science Laboratory, Palo Alto, California, January 11, 1990.

Gary T. Leavens, "Behavioral Subtyping for Abstract Data Types."

Rockwell-Collins, Cedar-Rapids, Iowa, December 11, 1989.

University of Iowa, Department of Computer Science, Iowa City, Iowa, December 5, 1989.

Carnegie Mellon University, School of Computer Science, Pittsburgh, Pennsylvania, November 20, 1989.

Gary T. Leavens, "Subtyping among Abstract Data Types."

Indiana University, Department of Computer Science, Bloomington, Indiana, March 30, 1988.

Wayne State University, Department of Computer Science, Detroit, Michigan, March 29, 1988.

Iowa State University, Department of Computer Science, Ames, Iowa, March 24, 1988.

University of Pennsylvania, Department of Computer and Information Science, Philadelphia, Penn., March 15, 1988.

University of California, San Diego, Department of Computer Science and Engineering, San Diego, California, April 14, 1988.

Michigan State University, Department of Computer Science, East Lansing, Michigan, April 13, 1988.

Committee Selected (technical colloquia)

Gary T. Leavens and Krishna Kishore Dhara, "A Model Theory for Abstract Data Types with Mutable Objects."

Workshop *Mathematical Foundations of Programming Semantics*, Manhattan, Kansas, March 21, 1994.

Krishna Kishore Dhara and Gary T. Leavens "Subtyping for mutable types in object-oriented programming languages," talk given by Krishna Kishore Dhara.

Workshop *Mathematical Foundations of Programming Semantics*, Manhattan, Kansas, March 22, 1994.

Gary T. Leavens and Don Pigozzi, "A Complete Algebraic Characterization of Behavior for Incompletely Specified Types and Subtypes," talk given by Don Pigozzi.

Workshop *Mathematical Foundations of Programming Semantics*, Manhattan, Kansas, March 23, 1994.

Colloquia about Teaching Computer Science

Invited

Gary T. Leavens, "A Viral, Infectious Approach to Outreach in Computing Education"

FM Outreach Meeting, Stanford Research Institute, Palo Alto, California, June 9, 2008.

Gary T. Leavens, "Previous Proposal and Critique: The CC 2001 PL Unit"

ACM SIGPLAN Programming Languages Curriculum Workshop, Harvard University, Cambridge, Mass., May 29, 2008.

Gary T. Leavens, "Programming Languages Knowledge Focus Group"

Iowa Undergraduate Computer Science Consortium Cornell College, Mt. Vernon, Iowa, November 4, 2000.

Gary T. Leavens and Albert L. Baker, "More Science in the Introduction to Computer Science" (Baker gave the second half of the talk).

Iowa Undergraduate Computer Science Consortium (Organizational Meeting), Grinell, Iowa, March 26, 1994

Other Colloquia

Invited

Gary T. Leavens, "Making a Positive Impact."

OOPSLA Doctoral Symposium, Nashville, Tennessee, October 20, 2008

Gary T. Leavens, "Lessons and Stories from My Career."

ECOOP Doctoral Symposium, Nantes, France, July 5, 2006.

Software Artifacts and Unpublished Documentation

JML is a behavioral interface specification language for Java modules. It was originally developed with Al Baker and my student Clyde Ruby, and further development has been done with many students (notably Yoonsik Cheon and Curtis Clifton) and host of international collaborators. More information is available on the web from the URL `http://www.jmlspecs.org`.

MultiJava is an extension to the Java programming language which adds support for open classes and symmetric multiple dispatch. It is primarily developed by my student Curtis Clifton in collaboration with Craig Chambers of the University of Washington and his (former) student Todd Millstein (now at UCLA). More information is available on the web from the URL `http://www.multijava.org`.

Larch/C++ is a behavioral interface specification language for C++ modules. It is available from the URL `ftp://ftp.cs.iastate.edu/pub/larchc++/`. The Larch/C++ reference manual is available from the following URL `http://www.eecs.ucf.edu/~leavens/larchc++.html`.

In support of Larch, I have written *Larch Frequently Asked Questions*, which was posted monthly to the usenet news groups comp.specification.larch, comp.answers, and news.answers. It is now available from the URL `http://www.eecs.ucf.edu/~leavens/larch-faq.html`.

Funding History

Funded Research Grants

Gary T. Leavens. REU Supplement for CRI:CRD: Collaborative Research: A JML Community Infrastructure – Revitalizing Tools and Documentation to Aid Formal Methods Research. National Science Foundation, $16,000 for undergraduate research from June 1, 2009 to August 14, 2010.

Gary T. Leavens (PI, with Jan Vitek). SHF: Small: Collaborative Research: Specification and Verification of Safety Critical Java. National Science Foundation, $249,981 from August 1, 2009 to July 31, 2011.

Gary T. Leavens (PI, with David A. Naumann). SHF: Small: Collaborative Research: Specification Language Foundations for Modular Reasoning Methodologies. National Science Foundation, $249,981 from August 1, 2009 to July 31, 2012.

Gary T. Leavens (PI), Samik Basu, and Hridesh Rajan. (with Yoonsik Cheon, Curtis Clifton, Cormac Flanagan, David Naumann, and Robby). CRI:CRD: Collaborative Research: A JML Community Infrastructure – Revitalizing Tools and Documentation to Aid Formal Methods Research. National Science Foundation, $275,000 (of which $235,809 is subcontracts to Iowa State) from August 15, 2007 to August 14, 2010.

Gary T. Leavens (PI, with David A. Naumann). Collaborative Research: Formal Methods for Behavioral Subclassing and Callbacks. National Science Foundation, $119,999 from September 1, 2004 to August 31, 2006, with a no-cost extension to August 31, 2007.

Gary T. Leavens (PI). More Modular Reasoning for Aspect-Oriented Programs. National Science Foundation, $54,999 from September 1, 2004 to August 31, 2005, with a no-cost extension to August 31, 2006.

Gary T. Leavens (PI). ITR/SY: Modular Interface Violation Checking Using Formally-Specified Contracts. Subcontract on a proposal by Murali Sitaraman of Clemson University, and Stephen Edwards of Virginia Tech. National Science Foundation, $109,383 for salary, research assistants, and travel, and $6,250 for undergraduate research assistants, August 16, 2001 to August 15, 2004; Extension and $10,000 for undergraduate research assistants, June 1, 2005 to May 31, 2007.

Gary T. Leavens (PI) and Don L. Pigozzi. Formal Methods for Extensible Object-Oriented Software. National Science Foundation, $200,000 for salary, research assistants, and travel. June 1, 2001 to May 31, 2003.

Gary T. Leavens (PI). Simultaneous Round-Trip Engineering for UML and Java/EJB. ETRI Computer & Software Technology Labs (Korea), $70,000 for salary, research assistants, and travel. June 1, 2001 to May 31, 2002.

Gary T. Leavens (PI) and Don L. Pigozzi and Albert L. Baker. Formal Methods for Multimethod Software Components. National Science Foundation, $210,000 for salary, equipment, and research assistants. June 1, 1998–May 30, 2000.

Gary T. Leavens (PI). Practical and Effective Interface Specifications. Rockwell International Corporation, $40,000 for research assistants. Jan 1, 1998–September 30, 1998.

Gary T. Leavens (PI) and Don L. Pigozzi. A Theoretical and Practical Basis for Applying Formal Methods to Object-Oriented Programming and C++. National Science Foundation, $239,998 for salary, equipment, and research assistants, July 15, 1995–June 1, 1998.

Gary T. Leavens (PI). Formal Specification for C++ Programs. National Science Foundation Research Initiation Award. $55,282 for salary, equipment, and research assistants, July 15, 1991–December 1993.

Gary T. Leavens (PI). Practical Reasoning for Object-Oriented Programs. Iowa State University Research Grant. $3,795 for equipment, May 1990–May 1991.

Gary T. Leavens (PI). Using Subtypes to Specify Object-Oriented Programs. Iowa State University Research Grant. $1,342 for travel, May 1989–May 1990.

Research Summary

The long term goal of my research is to better understand how to solve programming problems: how to specify such problems, methods for thinking about such problems, notations for expressing solutions, and ways to check that the solutions are correct. In pursuing this goal, I have worked in two main areas: formal methods and programming languages.

Formal Methods

My work in formal methods has been focused on ways to specify and verify object-oriented (OO) software components. The specification work involves the design and formal description of behavioral interface specification languages (BISLs). BISLs record information about detailed design: the interfaces and functional behavior of program modules. My group has designed a BISL for Smalltalk, called Larch/Smalltalk, a BISL for C++, called Larch/C++, and a BISL for Java called JML. Current work focuses on JML, and is being done with a large and growing international team of collaborators. Work on JML focuses on the problem of how to make it expressive enough for documenting existing code; we measure this using both theoretical analysis and case studies. We have made some progress; for example, work with Clyde Ruby reported at *OOPSLA 2000* and in his dissertation aims to explain modular subclass verification: how to specify enough about superclasses so that subclasses can be verified without seeing the superclass's code. There have also been papers (with Peter Müller and Arnd Poetzsch-Heffter) that uses ownership to solve modularity problems in the specification and verification of layered systems. JML has also fostered much interesting work by other researchers. However, there are several important features that need more work, such as concurrency and performance constraints. Our current work on JML funded by the National Science Foundation (NSF) tries to address some of these issues, as well as issues of practicality. Early work on JML was done jointly with (former ISU professor) Al Baker, and former Ph.D. student Clyde Ruby. Former Ph.D. student Yoonsik Cheon, designed Larch/Smalltalk, helped in the design of Larch/C++, tools for JML, and is also contributing to the design of JML as an assistant professor at Univ. of Texas, El Paso. Several of my other students have also helped with the JML project. I have also worked with Baker and a former Ph.D. student, Tim Wahls, on executing such specifications.

A long term interest in formal methods for OO software components is behavioral subtyping. The message passing mechanism of OO languages, such as C++ and Java, allows one to easily extend programs by adding new types. This works best if objects of the new types behave like the objects of the old types; the old types are *supertypes* of the new types, which are called *subtypes*. How should one reason about programs that use subtyping and message passing? We seek a reasoning method, formalized by specification and verification techniques, that is *modular* in the sense that when subtypes are added to a program, unchanged modules do not have to be respecified or reverified.

The idea of *behavioral subtyping*, which I helped develop, supports a programming discipline, *supertype abstraction*, that solves this problem. To use supertype abstraction, one specifies and verifies code in terms of the static types of expressions written in a program (as usual), uses a type checker to ensure that the static types are supertypes of the run-time types, and then must prove that subtypes obey the specifications of their supertypes. Behavioral subtyping makes this technique sound. Krishna Kishore Dhara, a former Ph.D. student, has extended the formal theory of behavioral subtyping to types whose instances have time-varying state in a way that takes into account the possibility of cross-type aliases. Professor Don Pigozzi (now retired from the Mathematics Department at ISU) and I have found an exact algebraic characterization of behavioral subtyping for immutable types. We have also been working on effective techniques for proving behavioral subtyping. Recent work with Professor David Naumann (of Stevens) on behavioral subtyping has precisely characterized modular reasoning (supertype abstraction) for Java-like languages with mutation, and formally justifies its soundness. The work with Dhara, Pigozzi, and Naumann on the above topics was funded by various grants from the NSF.

The potential impact of the work in formal methods is possibly great; it might lead to the engineering of software, instead of hacking. It also seems necessary for high quality software components and reuse. But more realistically, I view my research as trying to formally understand what one needs to think about when documenting and reasoning about a program or program component. This can be of great value for teaching and for the construction of tools, even if people do not use the formalism directly or on a daily basis.

Programming Language Design and Semantics

The other main aspect of my work has been on the design and semantics of programming languages. This falls in two areas: object-oriented languages with generic functions and aspect-oriented languages.

Languages with generic functions are also known as *multimethod* programming languages, because method calls can dispatch a message send on all arguments, unlike a *single-dispatching* OO language, such as Smalltalk, C++, or Java. Multimethod languages are interesting because they can more easily express solutions to certain problems in OO programming (binary methods). My work on multimethod languages is joint with former Ph. D. student Curtis Clifton (now a professor at Rose-Hulman) and has been done in collaboration with Craig Chambers of the University of Washington and his former student Todd D. Millstein (now a professor at UCLA). (My students Jianbing Chen and Sevtap Karakoy also have contributed to this work.) The work focuses on the semantics and type systems for variants of the Cecil and Java languages. To date we have published papers about an algorithm for type checking such languages with very expressive features (orthogonal inheritance and subtyping), about how to add multimethods to existing languages, and a way to add multimethods to Java. The latter has been developed into the language MultiJava and is featured in an *ACM TOPLAS* article.

One big problem I worked on (with Craig Chambers) was how to get a language with both a (sound) static type system and a sensible module system. This problem was solved by Millstein and Chambers (as reported in *ECOOP '99*). We have applied their ideas to the design of an extension to the Java programming language called MultiJava. The ideas also seem applicable to the design of OO languages that are more flexible.

A second interest in programming languages has been in aspect-oriented (AO) languages. As typified by AspectJ, AO languages offer advanced features for modularizing cross-cutting concerns, such as advice and intertype declarations (like MultiJava's "open classes"). As part of a long term effort to better understand AO languages, my former Ph.D. student Curtis Clifton and I have been working

on reasoning (e.g., about correctness) in AO languages. This work has been backed by a study of the operational semantics and type systems of AO languages. Curtis developed a simple and easily understood operational semantics for an AspectJ-like language, with a sound type system. This work also involved an effect system (concern domains) that makes it possible to more efficiently and effectively reason about AO programs. Recent results towards this end (with James Noble) appear in *ECOOP 2007*.

More recently, Hridesh Rajan and I have been working on a programming language that is less expressive than AspectJ-like languages, but which has advantages for modular reasoning. This language, Ptolemy, is a hybrid of an implicit-invocation language and an aspect-oriented language. It has promise both as a low-level language for virtual machines and for supporting easier static reasoning.

The potential impact of these research directions might be large if this leads to more flexible, modular, and reusable coding practices. An NSF grant funded Curtis Clifton's research on AO languages.

Research Employment

Massachusetts Institute of Technology, Cambridge, Mass **Sept. 1982–Dec. 1988**

Research Assistant in the Programming Methodology Group under the supervision of Professor Barbara Liskov. Active in the design of the Argus distributed programming language and system, especially in the design of the Argus library and features for extending the utility of polymorphism. Coauthor and editor of the *Argus Reference Manual*. Design and implementation of parts of the Argus run-time system. Testing of the Argus implementation by building small distributed systems.

Other Employment

Member of Technical Staff, from May 1979 to August 1982 at Bell Telephone Laboratories, 11900 Pecos, Denver, Colorado 80234. Development and maintenance of a large software management system.

Programmer-Coder, from May to September 1978, at Cutler-Hammer, Inc., 9475 Center Road, Fenton, Michigan 48480. Design of real-time control and monitoring software, code conversion.

Teaching Interests

Programming languages (semantics, design, surveys of paradigms, principles), formal methods, aspect-oriented programming, object-oriented programming, logic programming, functional programming, introductory programming, data structures, software engineering, compiler construction, distributed programming, distributed systems, formal logic, discrete mathematics, research methods and writing.

Teaching Employment

University of Central Florida, Orlando, FL **August 2007 – present**

Associate Director of EECS (August 2008 – present). Professor (August 2007 – present). Teaching undergraduate and graduate courses in programming languages.

Iowa State University, Ames, Iowa **January 1989 – August 2007**

Professor of Computer Science (2000 – 2007). Associate professor of Computer Science (1995 – 2000). Assistant professor (1989 – 1995). Teaching seminars, and graduate and undergraduate courses in programming languages and programming language semantics. Developed and taught a new undergraduate course "Object-Oriented Analysis and Design" (2002-2003). Developed and taught a undergraduate course "Introduction to Computer Programming," which taught freshmen programming using Scheme (1992 – 1993).

University of Iowa, Iowa City, Iowa **August 2000 – May 2001**

Visiting Professor. Taught an undergraduate course in programming languages and a graduate course in formal methods.

Massachusetts Institute of Technology, Cambridge, Mass. **1982 – 1986**

Teaching assistant for graduate courses in programming languages, undergraduate courses in software

engineering and compiler construction.

The University of Michigan, Ann Arbor, Mich. **Spring 1979**

Teaching Assistant for graduate course in compiler construction.

Students

Ph.D. Students at the University of Central Florida

Alexandre Bassel, January 2009–present, potential student

Rochelle Elva (joint with David Workman), "Investigating a Model for the Extraction of Program DNA to Uniquely Identify Java Programs and Measure Program Similarity," Fall 2009–present, in progress.

Faraz Hussain Summer 2008–present, in progress.

Ghaith Haddad, "Specification and Verification of Timing Constraints in Safety-Critical Java," Fall 2007–present, in progress.

Ph.D. Students at Iowa State University

Steve Shaner, Summer 2007–present, in progress.

Curtis Clifton, "A design discipline and language features for modular reasoning in aspect-oriented programs," Spring 2002–July 2005, complete.

Steven Jenkins, "Automating Systems Configuration Management," Summer 1995–present, inactive.

Clyde Ruby, "Modular subclass verification: safely creating correct subclasses without superclass code," Spring 1995–Fall 2006, complete.

Yoonsik Cheon, "A Runtime Assertion Checker for the Java Modeling Language," Summer 1991–April 2003, complete.

Krishna Kishore Dhara, "Behavioral Subtyping in Object-Oriented Programming Languages," Fall 1992–May 1997, complete.

Timothy Wahls (joint with Albert Baker), "On the Execution of High Level Formal Specifications," Fall 1992–Spring 1995, complete.

M.S. Students at Iowa State University

Faraz Hussain, "Enhancing a Behavioral Interface Specification Language with Temporal Logic Features," Fall 2006–April 2009.

Neeraj Khanolkar, "Formal Specification, Refinement, and Implementation of a Framelet Architecture," Fall 2005–May 2009 (co-major with Soma Chaudhuri), changed topics in May 2009

Kristina Boysen Taylor, "A Specification Language Design for the Java Modeling Language (JML) Using Java 5 Annotations," Summer 2005–May 2008.

Jason Jones, Spring 2005–present (co-major with Masha Sosonkina).

Steve Shaner, "Modular verification of higher-order methods with mandatory calls specified by model programs," January 2005–December 2008.

Cui Ye, "Improving JML's assignable clause analysis" January 2005–July 2006, complete.

Kun Liang (co-major with Simanta Mitra), "Topics in Verification," Fall 2004–present, inactive.

Brian Dorn, "Design and implementation of a reusable type inference engine and its application to Scheme," Summer 2004–May 2005, complete.

Douglas Fuller (co-major with Ricky Kendall), "Translation techniques for distributed-shared memory programming models" Fall 2003–July 2005, complete.

Tongjie Chen, "Extending MultiJava with Generics," Spring 2002–Fall 2004, complete.

Jeremiah Patterson, "An object-oriented event calculus," Fall 1999–Summer 2002, complete.

Medhat Assaad, "Alias-free parameters in C using multibodies," Spring 2000–Summer 2001, complete.

Arun Raghavan, "Design of a JML documentation generator," Spring 2000–Summer 2000, complete.

Abhay Bhorkar, "Run-Time Assertion Checker for JML," Spring 1998–Spring 2000, complete.

Curtis Clifton, "MultiJava: Design, implementation, and evaluation of a Java-compatible language supporting modular open classes and symmetric multiple dispatch," Spring 1998–Fall 2001, complete.

Anand Ganapathy, "Design and Implementation of a JML Type Checker," Summer 1998–Spring 1999, complete.

Jianbing Chen, "Dynamic Semantics and Type-checking of Tuple," Summer 1998–Fall 1998, complete.

Olga Antropova, "ACL Programming Language: Eliminating Parameter Aliasing with Dynamic Dispatch," September 1997–Summer 1998, complete

Sevtap Karakoy, "Evaluating the Expressiveness of a Multi-Method Programming Language," September 1997–Summer 1998, complete.

Hua Zhong, "LSL Traits for Using Z with Larch," January 1997–December 1997, complete.

Matthew Markland, "Design and Implementation of the Larch/C++ Type System," Fall 1995–August 1997, complete.

Robert J. Lavey, "Tanager: A case study of iterative development in object-oriented analysis and design," Spring 1994–April 2007, complete.

Gowri Sankar Sivaprasad, "Larch/CORBA: Specifying the Behavior of CORBA-IDL interfaces," Fall 1993–Fall 1995, complete.

Steven Jenkins, "Polymorphic Type Checking in Scheme," Summer 1993–Spring 1995, complete.

David Egle, "Evaluating Larch/C++ as a Specification Language: A Case Study Using the Microsoft Foundation Class Library," Fall 1994–Summer 1995, complete.

Joseph Reynolds, "A Literate Executable, Denotational Semantics of Simple C++ Declarations," Fall 1991–Spring 1993, complete.

Kari Lyle, "Refinement in Data Flow Diagrams," Fall 1991–July 1992, complete.

Krishna Kishore Dhara, "Subtyping Among Mutable Types in Object-Oriented Programming Languages," Spring 1990–Fall 1992, complete.

Timothy Wahls (joint with Professor Albert Baker), "A Formal (and Executable) Semantics for RT-SPECS," Fall 1991–Summer 1992, complete.

Yoonsik Cheon, "Larch/Smalltalk: A Specification Language for Smalltalk," Fall 1990–Summer 1991, complete.

Patricia O'Donnell, "Implementation of the Programming Language Prosper," Fall 1989–Fall 1990, inactive.

B.S. Honors Students at Iowa State University

Jeff Beach, undergraduate honors research project, Spring 1990.

Professional Activities

Editorial board member for *Transactions on Aspect-Oriented Software Development* (published by Springer-Verlag), since November 2009.
Assistant editor for *Software and Systems Modeling (SoSyM)* (published by SpringerVerlag), since 2001.

Research program committee chair for the ACM conference *Object-Oriented Programming: Systems, Languages, and Applications (OOPSLA)* 2009.

Co-founder (1999) and co-organizer (yearly, since 1999) with Sophia Drossopoulou, Susan Eisenbach, Peter Müller, Arnd Poetzsch-Heffter, Erik Poll, and others of the international workshop series *Formal Techniques for Java-Like Programs (FTfJP)*, held yearly at the *the European Conference on Object-Oriented Programming (ECOOP)*. Program committee member in 2008, 2007, 2002, 2001, 2000, 1999.
`http://www.cs.kun.nl/~erikpoll/ftfjp/`

Co-founder (2002) and co-organizer (yearly, since 2002) with Curtis Clifton, Mira Mezini, Shmuel Katz, and others of the international workshop series *Foundations of Aspect-Oriented Languages (FOAL)*, held yearly at the ACM Conference on *Aspect-Oriented Software Development (AOSD)*. Program committee chair in 2003. Program committee member in 2007, 2005, 2004, 2003, 2002.
`http://www.eecs.ucf.edu/FOAL/`

Co-founder (2001) and co-organizer (since 2003, and in 2001) with Jonathan Aldrich, Mike Barnett, Dimitra Giannakopoulou, Natasha Sharygina, and others of the *Specification and Verification of Component-Based Systems (SAVCBS)*, held yearly at the ACM Conference on *Foundations of Software Engineering (FSE)* (since 2003), and at *OOPSLA* in 2001. Program committee chair in 2005. Program committee member in 2005, 2004, 2003. `http://www.eecs.ucf.edu/SAVCBS/`

ACM representative on IFIP Technical Committee 2 (acting as liaison from ACM SIGSOFT and SIGPLAN to IFIP TC2), since April 2009.

Co-organizer, with Steven Freund of the "What" working group of the *Programming Languages Curriculum Workshop*, June–July 2008. See the report in *ACM SIGPLAN Notices*, **43**(11):6-29, November 2008.

Organizer of the "Verified by Construction" working group of the *IFIP Working Conference on Verified Software: Theories, Tools, Experiments*, October 2005 to October 2006.

Co-editor, with Susan Eisenbach, of a special issue of the journal *Concurrency, Practice and Experience*, volume 13, number 13, 2001. This issue is devoted to papers from the *Formal Techniques for Java Programs* Workshop, at *ECOOP 2000*, Cannes, France, 2000.

Co-organizer with Murali Sitaraman and program committee member of the *Foundations of Component-Based Systems Workshop*, Zurich, Switzerland, September 26, 1997.
See `http://www.eecs.ucf.edu/~leavens/FoCBS/index.html` and the report in *ACM Software Engineering Notes*, **23**(1):38–41, January 1998.

Member of program committee for *Object-Oriented Programming: Systems, Languages, and Applications (OOPSLA)* in 2009, 2008, 2006, 2004, '97, '95, '93, *European Conference on Object-Oriented Programming (ECOOP)* in 2009 and 2003, *Verified Software: Theories, Tools, Experiments (VSTTE)* in 2008, *Algebraic Methodology and Software Technology (AMAST)* in 2008 and 2000, *ICSE Research Demonstrations Track* in 2008, *Fundamental Approaches to Software Engineering (FASE)* in 2008, *TOOLS Europe* in 2007, *TAP International Conference: Tests And Proofs* in 2007, *Foundations of Object-Oriented Languages / Workshop on Object-Oriented Developments (FOOL/WOOD)* in 2006, *Formal Methods* in 2006, *Principles of Programming Languages (POPL)* in 2006, *Aspect-Oriented Software Development (AOSD)* in 2003, *IFIP Working Group 2.1 Working Conference on Generic Programming* in 2002, *Foundations of Object-Oriented Languages Workshop (FOOL)* in 2000, *22nd International Conference on Software En-*

gineering (ISCE) in 2000, *Second International Conference on the Unified Modeling Language (UML'99)*, *European Software Engineering Conference and ACM SIGSOFT International Symposium on the Foundations of Software Engineering (ESEC/FSE '99)*, *ACM SIGSOFT Symposium on Software Reusability (SSR'99)*, *Conference on Object-Oriented Technologies and Systems (COOTS) '99*, *COOTS '98*, *International Conference on Software Reuse ICSR5 '98*, *ICSR4 '96*, *Mathematical Foundations of Programming Semantics, MFPS '93*, *International Workshop on Specification Techniques and Formal Methods '98*, and *Component-Based Software Development in Computational Logic Workshop '98*.

Journal referee for *Formal Aspects of Computing* (twice in 2009-2008, 2007, 2000, 1996, 1993, 1991), *Science of Computer Programming* (in 2008, 2007, 2002), *IET Journal* (in 2008), *Software and Systems Modeling* (in 2008, 2007), *Journal of the ACM* (in 2007), *Formal Aspects of Computing* (in 2007, 2000, 1996, 1993, 1991), *Transactions on Aspect-Oriented Software Development* (in 2006), *Software Practice and Experience* (in 2006, 1993, 1992, 1991), *ACM Transactions on Software Engineering and Methodology* (in 2006, 2005, 2003, 2002, 2001, 1998, 1997), *IEEE Transactions on Software Engineering* (in 2005, twice in 2000, and 1999, 1998, 1996, 1991), *Proceedings of the IEEE* (in 2005), *Journal of the Brazilian Computer Society* (twice in 2005), *ACM Transactions on Programming Languages and Systems* (in 2004, twice in 2003, 2002, twice in 1994, 1987), *Information and Software Technology* (2004), *Acta Informatica* (in 2003), *Automated Software Engineering* (in 2003), *Theoretical Computer Science* (in 2002, 1996), *Information and Computation* (in 2000, 1999), *Theory and Practice of Object Systems* (in 1998, 1995), *IEEE Concurrency* (in 1998), *IEEE Computer* (in 1997, 1992), *Journal of Functional Programming* (in 1997), *Object-Oriented Systems* (in 1996, 1995, 1994), *ACM Computing Surveys* (in 1995, 1993/4), *International Journal of Microcomputer Applications* (in 1994), *Journal of Systems and Software* (in 1992), *Communications of the ACM* (in 1983). Conference referee for *CADE 2005*, *OOPSLA 2002*, *ECOOP '99* (in 1998), *OOPSLA '98*, *POPL '97* (in 1996), *AMAST '95* (in 1995), *AMAST II* (in 1991), and various other conferences (in the 1980s).

Chair of the OOPSLA 2010 doctoral symposium.

Reviewer/discussant for the OOPSLA 2008 and '96 doctoral symposia. "Expert" at the AOSD 2005 and 2003 Student Research Extravaganzas.

Proposal reviewer for the US National Science Foundation (a panel with 2 reviews in 2009, a panel with 7 reviews in 2007, a panel in 2006, a panel with 7 reviews in 2005, a panel with 8 reviews in 2004, once in 2003, once in 2001, a panel with 9 reviews in 1998, a panel with 8 reviews in 1996, four times in 1995, twice in 1994, and in 1993, 1991), the Natural Sciences and Engineering Research Council of Canada (in 2006, 2005, and 2002), the Dutch National Science Foundation (in 2004), and the US Department of Energy (in 1991).

Participant in the US National Science Foundation's Directorate for Computer and Information Science and Engineering's area study, during 2005.

Consultant book reviewer for MIT Press (1995), Kluwer Academic Publishers (1995), Richard D. Irwin, Co. (1994), Addison-Wesley Publishing Co. (1994), McGraw-Hill (1993 and 1992), Benjamin-Cummings (1993), Prentice-Hall (1993), Franklin, Beedle, & Associates (1992).

External Ph.D. examiner for: Peter Hui, Depaul University, Chicago, USA (2009); Mohamed El Bendary, University of Wisconsin, Milwaukee, Wisconsin, USA (2008); Devi Prasad, Motilal Nehru National Institute of Technology, Allahabad, India (2006); Ulf Schünemann, Memorial University of Newfoundland, St. John's, Newfoundland, Canada (2005); Peter Müller, FernUniversität, Hagen, Germany (2001); Andrew Bancroft Boake, University of Pretoria, Pretoria, South Africa (1998); Patrice Chalin, Concordia University, Montréal, Québec (1995).

Organizer of a working group at *WISR8: 8th Annual Workshop on Software Reuse*; see the report in *ACM Software Engineering Notes*, **22**(5):17–19, September 1997.

Member of the IEEE Distinguished Visitor Program, Spring 2003–Spring 2005

Member of an ad-hoc committee to select the most influential OOPSLA papers from 1986–1996 (in 2006).

Member of the ACM SIGPLAN Education Board from 2009 to present.

Co-organizer, with Stephen Freund, of the "What" report for ACM SIGPLAN's Programming Language Curriculum Workshop, in 2008.

Member of the Knowledge Area Focus Group on Programming Languages for the ACM/IEEE Computer Science Curriculum 2001 effort (during 1999).

Member of panel on "Software Testing in the Computer Science Curriculum," Iowa Undergraduate Computer Science Consortium, Simpson College, Iowa, March 25, 2000.

Organized and moderated the panel "Science vs. Engineering in Computing," Iowa Undergraduate Computer Science Consortium, Simpson College, Iowa, December 2, 1995.

Professional Societies

Senior member of the Association for Computing Machinery (since 1977, senior since 2007) and its special interest groups for programming languages and software engineering. Senior member of the IEEE Computer Society (since 1986, senior since 1999) and its technical committee on software engineering. Member of the American Association of University Professors (since 1992). Member of the European Association for Programming Languages and Systems (since 1998). Member of IFIP Working Group 2.3 (Programming Methodology) (since 2005).

Honors and Awards

College of Liberal Arts and Sciences Award for Outstanding Teaching, Iowa State University, 2007.

"Memorable Teacher" award, Iowa State University, March 2007.

Senior Member of the ACM, 2007.

IEEE Distinguished Visitor Program speaker, 2003-2005.

Senior Member of the IEEE, 2000.

Full member of the graduate faculty, Iowa State University, 1993.

GenRad/AEA Faculty Development Fellowship, 1983.

Phi Beta Kappa, December 1978.

Angell Scholar, March 1979.

Class Honors, 1977, 1978, 1979.

College Honors Program, 1975-1978.

Regents Alumni Scholar, 1975.

Service

University of Central Florida

Showcase of Undergraduate Research Excellence judge, Spring 2009.

College of Engineering and Computer Science at the University of Central Florida

Promotion and Tenure Review Committee, since Fall 2008, chair starting Summer 2009.

RIA Selection Committee, Spring 2009.

School of EECS at the University of Central Florida

Associate Director, August 2008–present.

Chair, Awards Committee, August 2008–present.

Chair, Faculty Search and Software Engineering Search Committee, October 2008–present.

Chair, Space and Laboratory Committee, August 2008–present.

Chair, Undergraduate Committee, August 2008–present.

Executive Committee, August 2008–present.

Computer Science Curriculum Oversight and Review Committee, Spring 2008–present.

Promotion and Tenure review committee, (anually since Fall 2007).

Undergraduate student advisor for 6 students, Spring 2008–present.

University at Iowa State University

Ad-hoc Post Tenure Review Committee, for a professor's review in Computer Engineering, September 2006.

Faculty Senate, February 1999–2000, 2001–Spring 2005.

Graduate Curriculum and Catalog Committee, Fall 2003–Spring 2004.

Liaison of the Faculty Senate's LAS caucus to the LAS representative assembly, Fall 2002–Spring 2005.

Panelist on Faculty/Adviser panel, for summer orientation, June 25, 2003.

Faculty Senate representative to the computation advisory committee, Fall 2001–Spring 2002.

Faculty Senate academic affairs committee, Aug. 1999–2000.

College of Liberal Arts and Sciences (LAS) at Iowa State University

Phi Beta Kappa Members in Course Committee, Spring 2007.

Curriculum committee, Spring 2002–Spring 2004.

Budget Dean Search (ad hoc committee) Spring 2000.

Representative Assembly, Sept. 1991–Spring 1998. Served on its Executive Committee in the 1995-1996, and 1997-1998 academic years.

Drafting committee for the LAS College Strategic Plan, Spring 1995.

Department, Computer Science at Iowa State University

Director of Graduate Education, Fall 2001–May 2007.

Chair, Graduate Committee, Fall 2001–May 2007.

Web Committee, Spring 2006–May 2007.

Promotion and Tenure Steering Committee, Fall 2005–Spring 2006.

Promotion and Tenure Steering Committee, Fall 2003–Spring 2004.

Chair, Promotion and Tenure Steering Committee, Fall 2004.

Undergraduate Committee, Sept. 1989–2000. (Chair of the committee in the 1999-2000 academic year.)

Catalog coordinator, Spring 2000.

Faculty Search Committee, 1999–2000 academic year.

Ad Hoc Committee on Post Tenure Review, Spring 2000.

Promotion and Tenure Steering Committee, 1995–1996 academic year.

Reform of introductory courses, June 1991–April 1992, including service on ad hoc committee, Spring 1992.

Graduate Admissions Committee, Jan. 1989–Aug. 1989.

Ad Hoc Committee for Teaching Awards, Oct. 1989 and Oct. 1990.

Mentor for Dalei Li in the "Preparing Future Faculty" program, Fall 2003.

Mentor for Adrian Silvescu in the "Preparing Future Faculty" program, Spring 2003.

Mentor for Curtis Clifton in the "Preparing Future Faculty" program, Fall 2002.

Member of the Information Assurance (INFAS) program. Member of the Iowa State Institute of Science and Society.

Temporary adviser for several graduate students, Sept. 1989–2007.

References

Available on request.

Citizenship

Citizen of the United States of America.