

Steven P. Reiss

Curriculum Vitae
January 2025

Personal

Professor
Department of Computer Science
Brown University
Providence, RI 02912
spr @ cs.brown.edu

Home Address

22 Palmer Meadow Lane
Rehoboth, MA 02769

Education

A.B., Mathematics, Dartmouth College, 1972
M.S., Computer Science, Yale University, 1975
Ph.D., Computer Science, Yale University, 1977
Dissertation topic: *Inverse Translation: The Theory of Practical Automatic Programming*
(supervised by D. Dobkin)

Professional Appointments

1969-1972	<i>Systems Programmer, Dartmouth Time Sharing System, Dartmouth College, Hanover, NH (full-time during summer and part-time during school year)</i>
1972-1974	<i>Programmer for DENDRAL project, Stanford University, Stanford, CA</i>
1977-1979	<i>Assistant Professor of Computer Science and Engineering, Brown University</i>
1979-1984	<i>Assistant Professor of Computer Science, Brown University</i>
1984-1989	<i>Associate Professor of Computer Science, Brown University</i>
1989-2025	<i>Professor of Computer Science, Brown University</i>

Publications

Books

FIELD: A Friendly Interactive Environment for Learning and Development, Kluwer, 1994.
A Practical Introduction to Software Design with C++, Wiley, 1998.
Inverse Translation: The Theory of Practical Automatic Programming, Ph.D. Thesis, Yale University (1977).

Book Chapters

“Medians and Database Security,” in *Foundations of Secure Computation*, Academic Press (1978).
“Programming Environments Today,” in *Annual Review Computer Science* 1:181-95 (1986).
“An Object-Oriented Framework for Graphical Programming,” in *Research Directions in Object-Oriented Programming*, ed. Bruce Shriver and Peter Wegner, M.I.T. Press (1987).

- “Visual Languages and the GARDEN System,” in *Visualization in Programming*, ed., P. Gorny and M.J. Tauber, Springer Verlag (1987).
- “Compiler Aspects of an Environment for Programming by Demonstration” (with Robert V. Rubin and Eric J. Golin), in *Visualization in Programming* Springer Verlag (1987).
- “The Brown Workstation Environment: A User Interface Design Toolkit” (with J. Stasko), *Engineering for Human-Computer Interaction*, North-Holland, pp. 215–232, 1990.
- “On the Use of Annotations for Integrating the Source in a Program Development Environment,” in *Human Factors in Analysis and Design of Information Systems*, North-Holland, pp. 25-36, 1990.
- “Constraining the Structure and Style of Object-Oriented Programs”, (with Scott Meyers and Carolyn Duby), in *Principles and Practice of Constraint Programming*, ed. Vijay Saraswat and Pascal Van Hentenryck, MIT Press, 1995.
- “Software Tools and Environments,” *The Computer Science and Engineering Handbook*, ed. Allen B. Tucker, CRC Press, pp. 2419-2439 (1997).
- “Visualization for Software Engineering — Programming Environments”, *Software Visualization*, ed. John T. Stasko, John B. Dominque, Marc H. Brown and Blaine A. Price, MIT Press, pp. 259-276 (1998).
- “Software Visualization in Teaching at Brown University”, (with John Basik, Roberto Tamassia and Andries van Dam), *Software Visualization*, ed. John T. Stasko, John B. Dominque, Marc H. Brown and Blaine A. Price, MIT Press, pp. 383-398 (1998).
- “The BLOOM Software Visualization System”, (with Manos Renieris), *Software Visualization — From Theory to Practice*, ed. Kang Zhang, Kluwer, 2003.

Refereed Articles

- “Aspects of the Database Security Problem” (with David P. Dobkin and Richard J. Lipton), in *Proceedings of A Conference on Theoretical Computer Science*, Waterloo (1977).
- “Security in Data Bases: A Combinatorial Study,” in *JACM* **26** (1), pp. 45-57 (1979).
- “Rational Search,” in *Information Processing Letters* **8** (2), pp. 89-90 (1979).
- “Linear Programming is Log-Space Hard for P” (with David P. Dobkin and Richard J. Lipton), in *Information Processing Letters* **8** (2), pp. 96-97 (1979).
- “The Complexity of Linear Programming” (with David P. Dobkin), in *Theoretical Computer Science* **II**, pp. 1-18 (1980).
- “Data-Swapping – A Technique for Disclosure Control” (with T. Dalenius), in *Journal of Statistical Planning and Inference* (1982).
- “Non-reversible Privacy Transformations” (with T. Dalenius and M. Post), in *Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (1982).
- “SLAP – A Methodology for Silicon Layout” (with John Savage), in *Proceedings of IEEE International Conference on Circuits and Computers* (1982).
- “Generation of Compiler Symbol Processing Mechanisms from Specifications,” in *ACM Transactions on Programming Languages and Systems* (1983).
- “Practical Data-Swapping: The First Steps,” in *ACM Transactions on Database Systems* (1984).
- “Toward a Computer Science Environment for Powerful Personal Machines” (with Marc H. Brown), in *Proceedings of the 17th Hawaii System Sciences Conference* (1984).

- “Graphical Program Development with PECAN Program Development Systems,” *ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environment*, SIGPLAN NOTICES **19** (5) (May, 1984).
- “An Approach to Incremental Compilation,” *SIGPLAN '84 Symposium on Compiler Construction*, SIGPLAN NOTICES **19** (6) (June, 1984).
- “PECAN: Program Development Systems that Support Multiple Views,” in *IEEE Transactions on Software Engineering* (March, 1985).
- “Thinkpad – A Graphical System for Programming by Demonstration” (with R. Rubin and E. Golin), in *IEEE Software* (March, 1985).
- “The Brown Workstation Environment” (with Joseph N. Pato and Marc H. Brown), in *Proceedings of Conference on Software Tools* (April, 1985).
- “Visual Languages and the GARDEN System,” in *Proceedings of an International Workshop on Visual Aids to Programming* (May, 1986).
- “Compiler Aspects of an Environment for Programming by Demonstration” (with Robert V. Rubin and Eric J. Golin), in *Proceedings of an International Workshop on Visual Aids to Programming* (May, 1986).
- “GARDEN TOOLS: Support for Graphical Programming,” in *Workshop on Advanced Program Development Environments* (June, 1986).
- “Prototyping Visual Languages with the GARDEN System” (with Eric J. Golin and Robert V. Rubin), in *Proceedings of IEEE Symposium on Visual Languages* (June, 1986).
- “An Object Server for an Object-Oriented Database System” (with Andrea H. Skarra and Stanley B. Zdonik, Jr.), in *International Workshop on Object-Oriented Database Systems*, Pacific Grove, CA (September, 1986).
- “Displaying Program and Data Structures” (with Joseph N. Pato), in *Proceedings of the 20th Hawaii International Conference on System Sciences* (January, 1987).
- “A Conceptual Programming Environment,” *9th International Conference on Software Engineering* (March, 1987).
- “Automatic Compiler Production: The Front End,” *IEEE Transactions on Software Engineering* Vol. SE13 **6** (June 1987).
- “Working in the Garden Environment for Conceptual Programming” *IEEE Software*, Vol. III **6**, (November, 1987).
- “A Semantic Basis for the Multiple Views of Programs,” *Advanced Working Papers of the 2nd International Workshop on Computer-Aided Software Engineering*, July 1988.
- “Representing Programs in Multiparadigm Software Development Environments” (with S. Meyers), in *Proc. COMPSAC '89*, 1989.
- “The Specification of Visual Language Syntax” (with Eric Golin), *Proc. 1989 IEEE Workshop on Visual Languages*, 1989.
- “Conceptual Programming,” *Proc. 5th International Software Process Workshop*, 1989.
- “Using GELO to Visualize Software Systems,” *Proc. UIST*, 1989.
- “Connecting Tools Using Message Passing in the FIELD Program Development Environment”. *IEEE Software*, (4), pp. 57–67, July, 1990.

- “Interacting with the FIELD Environment,” *Software Practice and Experience* 20 (S1), pp. 89–115, June 1990.
- “FIELD Support for C++” (with Scott Meyers), in *Proceedings of USENIX C++ Conference*, pp. 293-300, April 1990.
- “Parsing in a Visual Language Environment,” (with Eric Golin), *Proceedings of the 1989 IEEE Workshop on Visual Languages*, October 1989.
- “Adding Semantic Information to C++ Development,” (with Moises Lejter and Scott Meyers), *Proceedings of C++ at Work*, September 1990.
- “Support for Maintaining Object-Oriented Programs,” (with Scott Meyers and Lejter Moises), *IEEE Trans. on Software Engineering*, December 1992.
- “A System for Multiparadigm Development of Software System” (with Scott Meyers), *Proceedings of the 6th International Workshop on Software Specification and Design*, Oct. 1991.
- “Tools for Object-Oriented Redesign,” *Proceedings of the 5th International Conference TOOLS*, July 1991.
- “An Empirical Study of Multiple-View Software Development,” (with Scott Meyers), in *Proc. of SIGSOFT '92: Fifth Symposium on Software Development Environments*, 1992.
- “CCEL: A Metalanguage for C++,” (with Carolyn K. Duby and Scott Meyers), in *Usenix C++ Conference Proceedings*, August 1992.
- “Program Visualization: Where We Go From Here,” *Proc. IFIP World Conference on Computing*, Sept. 1992.
- “Constraining the Structure and Style of Object-Oriented Programs,” (with Scott Meyers and Carolyn K. Duby), *Proc. First Workshop on Principle and Practice of Constraint Programming*, April 1993.
- “Trace-Based Debugging,” *Proc. AADeBUG '93*, May 1993.
- “Presentation and Editing of Structured 3-D Graphics,” in *Proc. HCI'93*, August 1993.
- “A Framework for Abstract 3-D Visualization,” in *Proc. Visual Languages '93*, August 1993.
- “Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens,” (with Manojit Sarkar, Scott S. Snibbe, and Oren J. Tversky), *Proc. ACM SIGGRAPH UIST*, Nov. 1993, pp. 81-91.
- “Practical Software Visualization”, (with Isabel Cruz), *Proc. CHI Workshop on Program Visualization*, April 1994.
- “3D Visualization of Program Information”, *Proc. of Graph Drawing '94*, Oct. 1994.
- “VALLEY: 3D Visualization of Program Information”, *Proc. FADIVA Workshop of 3D Visualization*, Nov. 1994.
- “An engine for the 3D visualization of program information”, *J. Visual Languages*, 1995.
- “Configuration management in terms of modules”, (with Yi-Jing Lin) *Proc. Workshop on Software Configuration Management*, May 1995.
- “Software Tools and Environments,” *Computing Surveys*, June 1996.
- “Simplifying data integration: the design of the desert software development environment”, *Proc. Intl. Conf on Software Engineering*, March 1996.
- “Configuration management with logical structures”, (with Yi-Jing Lin), *Proc. Intl. Conf on Software Engineering*, March 1996.
- “Cacti: A front end for program visualization”, *Proc. Information Visualization '97*, October 1997.

- “Software visualization in the Desert environment”, *Proc. ACM SIGPLAN/SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, June 1998, pp. 59-66.
- “The Desert Environment”, *ACM Trans. on Software Engineering and Methodology*, 8(4), October 1999, pp. 297-342.
- “ALMOST: exploring program traces”, (with Manos Renieris), *Proc. 1999 Workshop on New Paradigms in Information Visualization and Manipulation*, October 1999.
- “Working with patterns and code”, *Proc. HICSS-33*, January 2000.
- “Generating Java trace data”, (with Manos Renieris), *Proc. Java Grande*, June 2000.
- “Encoding program executions”, (with Manos Renieris), *Proc. ICSE 2001*, May 2001.
- “Bee/Hive: A software visualization back end”, *IEEE Workshop on Software Visualization*, May 2001.
- “An Overview of BLOOM”, *Dagstuhl Workshop on Software Visualization*, May 2001.
- “An Overview of BLOOM”, *ACM SIGPLAN-SIGSOFT PASTE ‘01*, June 2001.
- “Consistent Software Evolution”, *SDP Workshop on New Visions for Software Design and Productivity*, December 2001.
- “Pervasive programs”, *Proc. CRA Grand Challenges Conference*, June 2002. URL
- “A visual query language for software visualization”, *Proc. Human Centric Computing Language and Environments*, September 2002.
- “Constraining software evolution”, *Proc. Intl. Conf. on Software Maintenance*, October 2002.
- “CLIME: An environment for constrained evolution (demonstration)”, (with Christina M. Kennedy, Tom Wooldridge, Shriram Krishnamurthi), *Proc. 25th Intl. Conf. on Software Engineering*, May 2003.
- “JIVE: visualizing Java in action (demonstration)”, *Proc. 25th Intl. Conf. on Software Engineering*, May 2003.
- “Languages for dynamic instrumentation”, (with Manos Renieris), *Proc. WODA 2003, ICSE Workshop on Dynamic Analysis*, May 2003.
- “Event-based performance analysis”, *Proc. 11th Intl. Workshop on Program Comprehension*, May 2003.
- “Visualizing Java in action”, *Proc. ACM Symp. on Software Visualization*, June 2003 pp 57-65.
- “Automated fault localization using potential invariants”, (with Brock Pytlik, Manos Renieris, Shriram Krishnamurthi), *Proc. AADeBUG 2003*, September 2003.
- “Fault localization with nearest neighbor queries”, (with Manos Renieris), *Proc. 18th Conf. on Automated Software Engineering*, October 2003.
- “Designing what you can’t”, *Proc. NSF workshop on the Science of Design: Software-Intensive Systems*, November 2003.
- “Elided conditionals” (with Manos Renieris and Sebastien Chan-Tin), *Proc. PASTE 2004*.
- “Dataflow language for scriptable debugging” (with Guillaume Marceau, Gregory H. Cooper, and Shriram Krishnamurthi), *Proc. ASE 2004*.
- “CHET: A system for checking dynamic specifications”, *Proc. ASE 2004*.
- “JOVE: Java as it happens” (with Manos Renieris), *Proc SoftVis 2005*, May 2005, pp 115-124.
- “Efficient monitoring and display of thread state in Java”, *Proc IWPC 2005*, May 2005, pp. 247-258.

“CHET: Checking specifications in Java systems”, Proc IWPC 2005, May 2005, pp. 165-168.

“JIVE and JOVE: Java as it happens”, Proc IWPC 2005, May 2005, pp. 169-172.

“Dynamic detection and validation of software phases”, Proc. WODA 2005, May 2005, pp. 50-55.

“Demonstration of JIVE and JOVE: Java as it happens”, Proc ICSE 2005, May 2005, pp 662-663.

“Myrrh: a transaction-based model for autonomic recovery” (with Guy Eddon), Proc 2nd Intl Conf. on Autonomic Computing, June 2005, pp. 315-325.

“Checking component specifications in Java systems”, Proc. SoftMC 2005, July 2005.

“From the concrete to the abstract: visual representations of program execution” (with Guy Eddon), Proc. of the 11th Intl Conf. on Distributed Multimedia Systems, pp 315-320, Sept. 2005.

“Evolving evolution”, Proc. 8th Intl Workshop on Principles of Software Evolution, September 2005, pp. 136-139.

“Arithmetic program paths” (with Manos Renieris and Shashank Ramaprasad), Proc ESEC/FSE ‘05, September 2005, pp. 90-98,

“Visualizing what people are doing on the web” (with Guy Eddon), Proc. VL/HCC 2005, September 2005, pp. 305-307.

“Specifying and checking component usage”, Proc AADEBUG ‘05, September 2005, pp. 13-22.

“The paradox of software visualization”, VISSOFT 2005, September 2005, pp. 59-63.

“Incremental maintenance of software artifacts”, Proc ICSM 2005, September 2005, pp. 113-122.

“A component model for Internet-scale applications”, Proc. ASE 2005, November 2005, pp. 34-43.

“Visualizing program execution using user abstractions”, Proc. SOFTVIS ‘06, September 2006, pp. 125134.

“Incremental maintenance of software artifacts”, IEEE Trans. on Software Engineering, 32(9), September 2006.

“The design and implementation of a dataflow language for scriptable debugging” (with Guillaume Marceau, Gregory H. Cooper, Jonathan P. Spiro, and Shriram Krishnamurthi), Automated Software Engineering Journal 14(1), pp. 59-86, 2007.

“Visual representations of executing programs”, Journal of Visual Languages and Computing, 18(2), pp. 126-148, 2007.

“Designing Internet-based software,” Second International Conference on Design Science Research, May 2007.

“Finding unusual code”, Proc ICSM 2007, October 2007, pp. 34-43.

“Automatic code stylizing”, Proc. ASE 2007, November 2007, pp. 74-83.

“Tracking source locations,” Proc. ICSE 2008, May 2008, pp. 11-20.

“Controlled dynamic performance analysis,” Proc. Workshop on Software Performance, June 2008, pp. 4354.

“Dynamic detection of event handlers,” Proc. Workshop on Dynamic Analysis, July 2008.

“Automatic detection of internal queues and stages in message processing systems using dynamic program analysis (with Suman Karumuri), ICPC 2009, pp. 315-316.

“Semantics-based code search,” ICSE 2009, pp. 243-253 (May 2009)

- “DYVISE: performance analysis of production systems (Research Demonstration), ICSE 2009 Companion, pp. 407-408.
- “Visualizing the Java heap to detect memory problems”, VISSOFT 2009, pp. 73-80.
- “Semantics-based code search (Research Demonstration)”, ICSM 2009, pp. 385-386.
- “Visualizing the Java Heap (Research Demonstration)”, ICSM 2009, pp. 389-390.
- “Code bubbles: a working set-based interface for code understanding and maintenance,” (with Andrew Bragdon, Robert Zeleznik, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Joseph J. LaViola, Jr.), CHI 2010, April 2010, pp. 2503-2512.
- “Visualizing the Java heap (Research Demonstration),” Proc ICSE 2010 Vol. 2, May 2010, pp. 251-254.
- “Code bubbles: rethinking the user interface paradigm of integrated development environments,” (with Andrew Bragdon, Robert Zeleznik, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Joseph J. LaViola, Jr.), ICSE 2010, May 2010, pp. 455-464.
- “A research demonstration of Code bubbles,” (with Andrew Bragdon, Robert Zeleznik, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Joseph J. LaViola, Jr.), ICSE 2010 Vol. 2, May 2010, pp. 293-296.
- “Visualizing threads, transactions, and tasks,” (with Suman Karamuri) PASTE 2010, June 2010, pp. 9-16.
- “Writing and using program specifications,” Proc. FoSER 2010, November 2010.
- “What is my program doing? Program Dynamics in Programmer’s Terms,” (with Alex Tarvo), Proc. Runtime Validation 2011, September 2011.
- “Using computer simulation to predict the performance of multithreaded programs,” (with Alex Tarvo). Proc. International Conference on Performance Engineering, April 2012.
- “Automatic programming as code search: a research agenda,” Proc. SUITE 2012, pp. 1-4, June 2012.
- “Plugging in and into Code Bubbles,” Proc. TOPI 2012, pp. 55-60, June 2012.
- “Code Bubbles: A Practical Working-Set Programming Environment,” (with Jared Bott, Joseph LaViola), ICSE 2012, pp. 1411-1414, June 2012.
- “Debugger canvas: industrial experience with the Code Bubbles paradigm,” (with Robert DeLine, Andrew Bragdon, Karl Rowan, and Jens Jacobsen), ICSE 2012, June 2012.
- “Motherhood and apple pie: modularity in modern applications and tools to support it,” Keynote at Modularity 2013, (2013).
- “Tool demonstration: the visualizations of Code Bubbles,” (with Alexander Tarvo) Proceedings of First IEEE Working Conference of Software Visualization, (September 2013).
- “Automatic categorization and visualization of lock behavior,” (with Alexander Tarvo) Proceedings of First IEEE Working Conference of Software Visualization, (September 2013).
- “Integrating S6 code search and Code Bubbles,” 3rd International Workshop on Developing Tools as Plugins, pp. 25-30 (May 2013).
- “Plugging in and into Code Bubbles: the Code Bubbles architecture,” (with Jared N. Bott and Joseph J. LaViola, Jr.) Software Practice and Experience, (2013).
- “Towards creating test cases using code search,” *IEEE International Conference on Software Maintenance and Evolution*, pp. 436-440 (2014).
- “Seeking the user interface,” *Proceedings of 29th ACM/IEEE International Conference on Automated Software Engineering*, pp. 103-114 (2014).

- “Tool demonstration: browsing software repositories,” *Proceedings of 29th ACM/IEEE International Conference on Automated Software Engineering*, pp. 589-592 (2014).
- “The challenge of helping the programmer during debugging,” *Proceedings of VISSOFT 2014*, (2014).
- “Automated analysis of multithreaded programs for performance modeling,” (with Alexander Tarvo) pp. 718 in *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*, (2014).
- “Automated analysis of multithreaded programs for performance modeling,” (with Alexander Tarvo) pp. 557-558 in *SIGMETRICS 2014*, (2014).
- “Building dynamic, long-running systems” (with Qi Xin), pp. 19-24, in *Proceedings of the 4th International Workshop on Software Engineering for Systems-of-Systems* (2016).
- “Hunter: next-generation code reuse for Java” (with Yuepeng Wang, Yu Feng, Ruben Martins, Arati Kaushik, Isil Dillig), pp. 1028-1032, in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, (2016).
- “A framework for a programmer's minion” (with Qi Xin), *Proceedings of the 39th International Conference on Software Engineering Companion*, pp. 112-114 (2017).
- “A demonstration of simultaneous execution and editing in a development environment” (with Qi Xin), *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, (2017).
- “Identifying test-suite-overfitted patches through test case generation” (with Qi Xin) pp. 226-236 in *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, , Santa Barbara, CA, USA (2017).
- “Leveraging syntax-related code for automated program repair” (with Qi Xin) *Proceeding of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, (2017).
- “GUIfetch: supporting app design and development through GUI search” (with Farnaz Behrang and Alessandro Orso), pp. 236-246, *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft '18)*.
- “Automatic performance prediction of multithreaded programs: a simulation approach” (with Alexander Tarvo), *Automated Software Engineering*, 25(1), pp. 101-155 (2018).
- “Seeking the user interface” (with Yun Miao and Qi Xin), *Automated Software Engineering*, 25(1), pp. 157-193 (2018).
- “SEEDE: simultaneous execution and editing in a development environment” (with Qi Xin and Jeff Huang), *ASE 2018*: 270-281
- “IoT end user programming models”, *Proceedings of SERP4IoT@ICSE*, pp. 1-8 (2019).
- “Better code search and reuse for better program repair” (with Qi Xin), *GI@ICSE*, pp. 10-17 (2019).
- “RoboMem: giving long term memory to robots” (with Ifrah Idrees, Stephanie Tellex), *Proceeding of MoRobAE -- Mobile Robot Assistants for the Elderly* (2019).
- “Aggregating Spatial-Temporal Instances of Objects for Efficient Retrieval over Long Periods of Time” (with Ifrah Idrees, Stephanie Tellex), *Proceedings of AAAI Fall Symposium on AI for HRI* (2021).
- “Building an Interface for Controlling IoT Devices”, *Proceedings of SERP4IoT@ICSE*, 2023.
- “Quick Repair of Semantics Errors for Debugging”, *Proceedings of the 2023 IEEE/ACM International Workshop on Automatic Program Repair (APR@ICSE)*, 2023.

- “Towards practical and useful automated program repair for debugging” (with Qi Xin, Haojun Wu, and Jifeng Xuan), SE2030: 2030 Software Engineering Workshop, (2024).
- “Detecting, creating, repairing, and understanding indivisible multi-hunk bugs” (with Qi Xin, Haojun Wu, Jinran Tang, Xinyu Liu, Jifeng Xuan), Proceedings of ACM Software Engineering (FSE), pp. 2747-2770, (2024)
- “ROSE: An IDE-Based Interactive Repair Framework for Debugging” (with Xuan Wei, Jiahao Yuan, Qi Xin), to appear in ACM Transaction on Software Engineering and Methodology (TOSEM), (2025).
- “Studying and Understanding the Effectiveness and Failures of Conversational LLM-Based Repair” (with Aolin Chen, Haojun Wu, Qi Xin, and Jifeng Xuan), APR 2025: The 6th International Workshop on Automated Program Repair, (2025).

Technical Reports

- “The Practicality of Data-Swapping,” Technical Report No. CS-48, Department of Computer Science, Brown University (1979).
- “Preliminary Comparison of Red and Green Languages” (with P. Wegner and T. Doeppner), Technical Report No. CS-47, Department of Computer Science, Brown University (1979).
- “Symbol Processing: A Compiler View,” Technical Report No. CS-65, Department of Computer Science, Brown University (1981).
- “bb Reference Manual” (with L. Medvene and M. Lustig), Department of Computer Science, Brown University (1981).
- “The Efficient Implementation of Flexible Relational Database Systems,” Technical Report No. CS-82-26, Department of Computer Science, Brown University (1982).
- “The Path Model of Relational Database Implementation,” Technical Report No. CS-82-02, Department of Computer Science, Brown University (1982).
- “Eris Reference Manual,” Technical Report No. CS-82-04, Department of Computer Science, Brown University (1982).
- “Debugging in the BALSAs-PECAN Integrated Environment” (with Marc H. Brown), Position Statement, Department of Computer Science, Brown University (1982).
- “SLAP—A Silicon Layout Program” (with John Savage), Technical Report No. CS-82-17, Department of Computer Science, Brown University (1982).
- “Eris: The Design and Implementation of an Experimental Relational Database System,” Technical Report No. CS-83-02, Department of Computer Science, Brown University (1983).
- “First Steps Toward a Computer Science Environment for Powerful Personal Machines” (with Marc H. Brown), Technical Report No. CS-83-04, Department of Computer Science, Brown University (1983).
- “WILLOW: A Window Manager,” Technical Memorandum No. TM-100-83, Department of Computer Science, Brown University (1983).
- “MAPLE Reference Manual” (with Marc H. Brown), Technical Memorandum No. TM-101-83, Department of Computer Science, Brown University (1983).
- “A Screen Handler,” Technical Memorandum No. TM-102-84, Department of Computer Science, Brown University (1983).

- “Virtual Terminal Package,” Technical Memorandum No. TM-103-84, Department of Computer Science, Brown University (1983).
- “PLUM—A Data Management Package,” Technical Memorandum No. TM-104-84, Department of Computer Science, Brown University (1983).
- “GARDEN: An Environment for Graphical Programming, Reference and Programmer’s Manual,” Department of Computer Science, Brown University (1987).
- “Integration Mechanisms in the FIELD Environment,” CS-88-18, Department of Computer Science, Brown University (1988).
- “Creating Graphical Languages in Garden” (with S. Meyers), CS-89-02, Department of Computer Science, Brown University (1989).
- “Representing Visual Programs with Object-Graphs” (with E. Golin), CS-89-05 Department of Computer Science, Brown University (1989).
- “Parsing in a Visual Language Environment” (with E. Golin), CS-89-06, Department of Computer Science, Brown University (1989).
- “Manipulating Screen Space with Stretch Tools: Visualize Large Structures on a Small Screen,” (with Manojit Sarkar), Technical Report No. CS-92-42, Department of Computer Science, Brown University (1992).
- “An Object-Centered Approach to Designing Programming Environments,” (with Y.-Jing Lin). Technical Report CS-93-38, Department of Computer Science, Brown University, Sept. 1993.
- “Type-directed code reuse using integer linear programming” (with Y. Wang, Y. Feng, R. Martins, A. Kaushik, I. Dillig). <http://arxiv.org/abs/1608.07745>, 2016.
- “Learning program component order” (with Qi Xin), <http://arxiv.org/abs/1707.06737>, (2017).
- “Revisiting ssFix for better program repair” (with Qi Xin), <http://arxiv.org/abs/1903.04583>, (2019).
- “Continuous flow analysis to detect security problems”, <http://arxiv.org/abs/1909.13683>, (2019).
- “Assisted specification of code using search”, <https://doi.org/10.48550/arXiv.2209.09804> (2022).
- “A quick repair facility for debugging”, <https://doi.org/10.48550/arXiv.2202.05577> (2022).

Invited Lectures

- 1978 *Automatic Compiler Production, Raytheon Corporation*
- 1980 *Introduction to Database Management Systems, Raytheon Corporation*
- 1981 *Automatic Compiler Production: The Front End, General Electric Corporation*
- 1982 *Automatic Compiler Production: The Front End, Bell Laboratories*
The Path Model of Relational Database Implementation, SUNY/Stony Brook
The Efficient Implementation of Flexible Relational Database Systems, Princeton University
The Efficient Implementation of Flexible Relational Database Systems, New York University
- 1983 *Generation of Compiler Symbol Processing Mechanisms from Specifications, Bell Laboratories*
PECAN: Program Development Systems that Support Multiple Views, General Electric Corporation
- 1984 *PECAN: Program Development Systems that Support Multiple Views*

- PECAN: Program Development Systems that Support Multiple Views, U.C. Berkeley*
PECAN: Program Development Systems that Support Multiple Views, Stanford University
PECAN: Program Development Systems that Support Multiple Views, DEC/SRC
Steps Toward Graphical Programming, USC/ISI, Oregon Graduate Center, IBM
- 1985 *Steps Toward Graphical Programming, CCA, Xerox PARC*
Working in the Garden System, SIGPLAN Boston
Working in the Garden System, University of Maryland
Working in the Garden System, University of Washington
- 1986 *COMPSAC Panel on Visual Programming*
FJCC Panel on Integrated Programming Environments
Working in the Garden System, DEC, Washington University-St. Louis
- 1987 *Working in the Garden System, University of California-Irvine*
Working in the Garden System, Stanford
Working in the Garden System, University of California-Berkeley
Working in the Garden System, DEC/SRC
Working in the Garden System, Bell Laboratories
- 1988 *Working in the Garden System, SUN*
FIELD: A Friendly Integrated Environment for Learning and Development, SUN
FIELD: A Friendly Integrated Environment for Learning and Development, Siemens
Conceptual Programming, Dedication of Thomas J. Watson, Sr. Center for Information Technology, Providence, RI
- 1989 *FIELD: A Friendly Integrated Environment for Learning and Development, DEC/SRC*
FIELD: A Friendly Integrated Environment for Learning and Development, Apple Computer
FIELD: A Friendly Integrated Environment for Learning and Development, IBM Cambridge Scientific Center
FIELD: A Friendly Integrated Environment for Learning and Development, University of Connecticut
Conceptual Programming, IBM
- 1990 *FIELD: A Friendly Integrated Environment for Learning and Development, Sun Microsystems*
FIELD: A Friendly Integrated Environment for Learning and Development, GTE
FIELD: A Friendly Integrated Environment for Learning and Development, General Electric
- 1991 *FIELD: A Friendly Integrated Environment for Learning and Development, Carnegie-Mellon University*
FIELD: A Friendly Integrated Environment for Learning and Development, Purdue University
FIELD: A Friendly Integrated Environment for Learning and Development, Matsushita Information Technology Laboratory
FIELD: A Friendly Integrated Environment for Learning and Development, Bell Laboratories
FIELD: A Friendly Integrated Environment for Learning and Development, Silicon Graphics
- 1992 *FIELD: A Friendly Integrated Environment for Learning and Development, UNIX System Laboratories*
Program Visualization: Where We go From Here, IBM T.J. Watson Laboratory

- Program Visualization: Where We go From Here, Workshop on Programming Environments (Dagstuhl)*
- Program Visualization: Where We go From Here, Sun Microsystems*
- Program Visualization: Where We go From Here, DEC CRL*
- Program Visualization: Where We go From Here, Dartmouth College*
- 1993 *Program Visualization: Where We Go from Here, University of Massachusetts Amherst*
- Program Visualization: Where We go From Here, Siemens Research Laboratories*
- Program Visualization: Where We go From Here, Northeastern University*
- Program Visualization: Where We go From Here, NSF CER Conference*
- Program Visualization: Where We go From Here, NYNEX Science & Technology Center*
- User Interfaces for Virtual Environments, NRL Workshop on Virtual Environments*
- 1994 *Program Visualization: Where We Go from Here, Sun Microsystems*
- Program Visualization: Where We go From Here, Columbia University*
- 1995 *Landscapes: Visions of Programming Environments, Bell Labs*
- Landscapes: Visions of Programming Environments, ICSE 17 Demo Track*
- Landscapes: Visions of Programming Environments, Silicon Graphics.*
- 1996 *Landscapes: Visions of Programming Environments, GTECH symposium.*
- Information Visualization, DARPA Visualization Day.*
- Living in the Desert: An Integrated Approach to Programming Environments, Microsoft.*
- 1997 *Living in the Desert: An Integrated Approach to Programming Environments, IBM Research.*
- 1998 *Software Understanding through Visualization, Sun Microsystems*
- Software Understanding through Visualization, IBM Research*
- Lessons in Messaging, UC Irvine workshop on Wide-area messaging.*
- 1999 *TEA: Changing the Way People Program, AT&T Research*
- TEA: Changing the Way People Program, Microsoft Research*
- 2001 *An Overview of Bloom, Sun Microsystems.*
- Analyzing Trace and Program Data, NEPLS at Williams College.*
- 2002 *Constraining Software Evolution, Microsoft Research.*
- 2003 *Constraining Software Evolution, UMass, Amherst*
- 2004 *The Brown Internet Computing Laboratory, NSF CISE/RI Workshop.*
- 2005 *The Brown Internet Computing Laboratory, NSF CISE/RI Workshop*
- Efficient Checking of Component Specifications in Java Systems, U. Arizona.*
- 2007 *The Impact of Research on Middleware Technology, Panel at ICSE 2007.*
- Experiences with Flow Analysis, Georgia Tech.*
- Future Directions in Programming Environments, Georgia Tech.*
- 2008 *Semantics-Based Search, IBM PL Day.*
- Dynamic Program Understanding, IBM Research.*
- 2009 *Visualizing the Java Heap, Harvey Mudd College*
- 2010 *Code Bubbles: Rethinking the Programming User Interface, Google*
- 2011 *Code Bubbles: Making the Vision Real, Microsoft Research*
- Code Bubbles: Making the Vision Real, CHOOSE symposium*
- 2014 *Seeking the User Interface, Microsoft Research*

Seeking the User Interface, Lehigh University
Seeking the User Interface, University of Pennsylvania
Seeking the User Interface, University of Eastern Finland

- 2015 Going Beyond Code Search, *Carnegie Mellon University*.
Going Beyond Code Search, *University of Texas at Dallas*.
Going Beyond Code Search, *University of California at San Diego*.
Going Beyond Code Search, *Georgia Tech*.

Research in Progress

Software development tools and environments; automatic bug repair; user interfaces for IoT

Service

To the University

1977 *Planning Committee of the Computer Advisory Committee*
1978-1980 *Faculty Policy Group*
1979-1985 *Computer Science Undergraduate Advisor*
1979-1985 *ACM Faculty Advisor*
1980-1982 *Freshman Advisor*
1985-2001 *CS Facilities Committee*
1986-1990 *Graduate Committee, Graduate Admissions*
1988-1992 *Freshman Advisor, Sophomore Advisor*
1990-1992 *Graduate Council*
1990-1993 *Computer Science Undergraduate Advisor*
1991-1993 *Chairman, CS Graduate Admissions*
1994-1997 *Computer Science Graduate Advisor*
1994-1999 *Freshman Advisor, Sophomore Advisor*
1996-1999 *Committee on Admissions and Financial Aid*
1998-2002 *Associate Chairman*
1999-2001 *Chairman, CS Junior Faculty Search Committee*
1999-2002 *Sophomore Advisor*
2000-2002 *Freshman Advisor*
2001-2004 *College Curriculum Committee*
2005-2006 *Graduate Council*
2007-2009 *Freshman Advisor*
2008-2010 *Sophomore Advisor*
2008-2009 *TPAC*
2011-2013 *TPAC*
2012-2013 *TPAC Chair*
2013-2014 *TPAC*
2017-2018 *University Committee to Review the Academic Code*
2017-2018 *Academic Code Committee*
2018-2021 *Academic Code Committee, Chair*
2022-2023 *Academic Code Committee, Chair*

To The Profession

- 1985 *Program Committee for ACM SIGPLAN '85 Symposium on Language Issues in Programming Environments*
- 1988 *Program Committee for 2nd IEEE International Conference on Computer Workstations*
Program Committee for ACM SIGSOFT '88: Third Symposium on Software Development Environments
- 1989 *Program Committee for 3rd IEEE Conference on Computer Languages*
- 1990 *Program Committee 1991 USENIX C++ Conference*
- 1993 *Program Committee, AADEBUG '93*
- 1994 *Program Committee, 3rd Workshop on Program Comprehension*
- 1995 *Program Committee, SIGSOFT Foundations of Software Engineering*
Program Committee, AADEBUG '95
- 1997 *ISAT working group on visualizing information spaces*
- 1999 *Program Committee, First Conf. on Software Engineering Tools.*
- 2000 *Program Committee, Second Conf. on Software Engineering Tools*
- 2001 *Organizer, IEEE Workshop on Software Visualization (in conjunction with ICSE)*
Organizer, ACM Workshop on Object-Oriented Software Visualization (in conjunction with OOPSLA)
- 2002 *Program Committee, First Conf. on Software Visualization.*
Member, Rhode Island College Computer Science External Advisory Board
- 2003 *Program Committee, 26th Intl. Conf. on Software Engineering*
Steering Committee, IEEE Conf. on Software Visualization
Program Committee, IEEE HCC '03 Symp. on Visual/Multimedia Software Engineering
Member, Rhode Island College Computer Science External Advisory Board
- 2004 *Program Committee, ICSE '04*
Program Committee, WoDiSEE '04
Program Committee, ICAC 2004
Member, Rhode Island College Computer Science External Advisory Board
- 2005 *Conference Chair, SoftVis '05*
Program Committee, AADebug '05
Program Committee, VL/HCC '05
Program Committee, ICSM '05
Member, Rhode Island College Computer Science External Advisory Board
- 2006 *Program Committee, SoftVis '06*
Program Committee, WODA 2006
Program Committee, VLC2006
Program Committee, ICSM 2006
Member, Rhode Island College Computer Science External Advisory Board
- 2007 *Program Committee, WODA 2007*
Program Committee ICSM 2007
- 2008 *Program Committee, VLC 2008*
Program Committee, SoftVis '08
Program Committee ICSM 2008
- 2009 *Program Committee, VLC 2009*
Program Committee, ICPC 2009

	<i>Program Committee, ICSM 2009</i>
	<i>Program Committee, VISSOFT 2009</i>
2010	<i>Program Chair, SoftVIS 2010</i>
	<i>Program Committee, ICSM 2010</i>
	<i>Program Committee, VLC 2010</i>
2011	<i>Program Committee, VisSoft 2011</i>
	<i>Program Committee, WODA 2011</i>
2012	<i>Program Committee, ICSE 2012 Mentor Program</i>
	<i>Program Committee, ICSM 2012</i>
	<i>Member, GRE Computer Science Committee</i>
2013	<i>Program Committee, ICSE 2013 Mentor Program</i>
	<i>Program Committee, ISCM 2013</i>
	<i>Program Committee, NEIR 2013</i>
	<i>Program Committee, VISSOFT 2013</i>
	<i>Member, GRE Computer Science Committee</i>
2014	<i>Program Committee, ICSE 2014 Mentor Program</i>
	<i>Program Committee, ICSE 2015</i>
	<i>Program Committee ICSME 2014</i>
	<i>Program Committee, VISSOFT 2014</i>
2015	<i>Program Committee, VISSOFT 2015</i>
	<i>Program Committee, ICSE 2016 V2025</i>
2016	<i>Program Committee, ICSE 2017</i>
	<i>Program Committee, VISSOFT 2016</i>
2018	<i>Program Committee, VISSOFT 2018</i>
2019	<i>Program Committee, VISSOFT 2019</i>
2020	<i>Program Committee, ASE 2020, Tool Demo</i>
	<i>Program Committee, VISSOFT 2020</i>
2021	<i>Program Committee, VISSOFT 2021</i>

Academic Honors, Fellowships, Honorary Societies

Honors

1969	<i>Thayer Mathematics Prize</i>
1970	<i>John G. Kemeny Prize in Computing, Second Prize, Applications Division</i>
1970	<i>John G. Kemeny Prize in Computing, Second Prize, Systems Division</i>
1972	<i>John G. Kemeny Prize in Computing, First Prize, Systems Division</i>
1972	<i>Phi Beta Kappa</i>
1974-1977	<i>National Science Foundation Graduate Fellowship</i>
1978	<i>Sigma Xi</i>
1984	<i>Best Paper Award for HICSS-17 Software Track</i>
1987-1989	<i>IBM Research Scholar</i>
1995	<i>ICSE-7 Retrospective Best Paper Award Honorable Mention</i>
2000	<i>Best Paper Award for HICSS-33 Software Track</i>
2002	<i>Best Paper Award for ICSM-02.</i>
2009	<i>ACM SigSoft Impact Paper Award</i>

2010 *VL/HCC Most Influential Paper Award from two decades ago.*
 2011 *Keynote Speaker, RV 2011.*
 2013 *Keynote Speaker, Modularity 2013.*
 2014 *ACM SigSoft Distinguished Paper Award, ASE 2014*
 2017 *Keynote Speaker, VISSOFT 2017*
 2018 *VISSOFT 2018 Most Influential Paper Award*
MOBILESoft 2018 Distinguished Paper Award

Research Grants

1979-1980 *National Science Foundation, "Computer Science and Computer Engineering Research Equipment" (with Eugene Charniak), \$55,096.*
 1979-1981 *National Science Foundation, "Automatic Compiler Construction," \$74,400.*
 1981-1983 *National Science Foundation, "Practical Data-Swapping," \$85,644.*
 1982-1987 *National Science Foundation, "An Integrated Experimental Environment for Research in Computer Science" (Co-principal investigator with A. van Dam, J.E. Savage, P. Wegner, R. Sedgewick, E. Charniak, T.W. Doeppner, J.S. Vitter and P.C. Kanellakis), \$2,771,311.*
 1982-1984 *National Science Foundation, "Automatic Production of Program Development Software," \$76,690.*
 1983-1987 *DARPA/ONR, "Ideographics" (Co-principal investigator with J.E. Savage, A. van Dam, G.M. Baudet, E. Charniak, B.M. Chazelle, T.W. Doeppner, P.C. Kanellakis, R. Sedgewick, P. Wegner, and J.S. Vitter), \$1,700,000.*
 1983-1985 *Semiconductor Research Corporation, "Hierarchical Silicon Compilation" (with G.M. Baudet and J.E. Savage), \$202,977.*
 1984-1985 *IBM, "Distributed Programming Languages – Implementation, Architecture, Methodology" (with P. Wegner and T.W. Doeppner), \$66,492.*
 1984-1985 *IBM, "Graphical Programming Environments" (with P. Wegner and T.W. Doeppner), \$66,492.*
 1985-1988 *IBM, "Environments and Languages for the 1990's" (with Peter Wegner), \$499,584.*
 1985-1988 *DEC, "Project Quartro," (with Andries van Dam, D.C.A. Bulterman and Thomas Doeppner), \$540,000.*
 1986-1987 *Sperry Corporation, "Software Configuration Advisor," \$34,580.*
 1987-1989 *DARPA, "Multiparadigm Design Environments" (with John Savage and Paris Kanellakis), \$1,776,120.*
 1987-1990 *DEC, "Practical Conceptual Programming," \$163,521 (and \$450,000 equipment*
 1988-1990 *U.S. West, "Object-Oriented Databases and Programming Languages and Multiparadigm Software Application Architectures" (with Stan Zdonik and Peter Wegner), \$277,804.*
 1988-1992 *National Science Foundation, "Multiparadigm Design Environments" co-principal investigator with J. Savage, E. Charniak, T. Doeppner, T. Dean, P. Kanellakis, D. Lopresti, L. Morgenstern, S. Reiss, A. Van Dam, J. Vitter, P. Wegner, K. Zadeck, and S. Zdonik, \$3,481,000.*
 1989-1991 *National Science Foundation, "GAMUT: A Laboratory for Computer Science Instruction," \$100,000.*
 1991-1994 *DARPA, "High-Performance Computing Environments" co-principal investigator with E. Charniak, T. Dean, T. Doeppner, J. Hughes, P. Kanellakis, P. Klein, Dan Lopresti, F.*

Preparata, S. Reiss, J. Savage, R. Tamassia, A. van Dam, P. Van Hentenryck, J. Vitter, P. Wegner, K. Zadeck, and S. Zdonik, \$2,697,175.

1991-1992 NSF, "Empirical Evaluation of Multiparadigm Software Development," \$50,000.

1991-1992 IBM, "Animated Program Visualization," \$31,414.

1992-1995 NSF, "Visualizing Abstractions," \$243, 596.

1992-1993 IBM, graduate fellowship for Y-J. Lin, \$15,000.

1992-1993 NSF, "Brown's Advanced Laboratory for Doing Rich Instructional Computing (BALDRIC)," \$100,000.

1993-1994 IBM, Graduate fellowship for Y-J. Lin, \$15,000.

1993-1994 NYNEX, "Tools for Object-Oriented Software Development," \$53,717

1994-1995 IBM, graduate fellowship for Y-J. Lin, \$15,000

1995-1997 NSF, "Landscapes: Visions of Programming Environments", \$225,000.

1995-1997 DARPA, "Integrated Environments for Distributed Computing", \$131,000

1996 Sun Microsystems, \$60,000 of equipment

1996-1997 DEC, \$200,000 of equipment credits

1996-1999 "Evolutionary Design of Software Systems", \$1,113,305

1997-1998 NSF, Educational supplement, \$40,000.

1999-2002 NSF, "Visualization for Software Understanding", (with David Laidlaw), \$559,213.

2000-2005 NSF, "Data Centers", (with Stan Zdonik), \$3,150,000

2000-2003 NSF, "Next Generation Programming Tools", \$297,205

2000-2001 Sun Microsystems, "Large-Scale Dynamic Visualization and Analysis of Java Programs", \$80,000

2001-2002 Sun Microsystems, "Large-Scale Dynamic Visualization and Analysis of Java Programs", \$50,000

2002-2005 NSF, "Constraining Software Evolution", (with Shriam Krishnamurthi) \$450,000.

2003-2005 NSF, "The Brown Internet Computing Laboratory", (with Tom Doeppner, Maurice Herlihy, Thomas Hoffman, Shriram Krishnamurthi, Roberto Tamassia, and Eli Upfal), \$600,000.

2005-2006 IBM SUR Grant, "The Brown Internet Computing Laboratory", (with Eli Upfal), about \$250,000 of equipment.

2005 IBM Eclipse Innovation Grant, "Myrrh: autonomic system for recovery with transactions", \$15,000.

2006 NSF, "Designing the Undesignable", \$224,093.

2007 IBM Jazz Faculty Grant, "JAZZing for Help and Review", \$25,000.

2008 NSF, "Applied Computer Science for the Humanities and Social Sciences", \$540,000.

2009 NSF, "Applied Computer Science for the Humanities and Social Sciences", REU Supplement, \$15,000.

2010 IBM X10 Innovation Grant, \$20,000

2010 Google Faculty Research Award, "Code Bubbles: Rethinking the Programming User Interface", (with Andy van Dam) \$45,000

2010 Microsoft, "Code Bubbles: Rethinking the Programming User Interface", (with Andy van Dam), \$90,000

2010-2015 NSF, "A Working Set Approach to Integrated Development Environments", (with Andy van Dam) \$1,132,450

2011 Curricular Development Grants: \$2,000 for CSCI13200, \$500 for CSCI1600.

2011 NSF, “A Working Set Approach to Integrated Development Environments,” REU Supplement, \$15,000.

Teaching (since 2010)

Courses

2010-2011 *Introduction to Computation for the Humanities and Social Sciences (CSCI0931, 8)*
 Programming Environments (CSCI2330, 4)
 Creating Modern Web Applications (CSCI1320, 67)
 Embedded and Real Time Programming (CSCI1600, 9)
2011-2012 *Introduction to Computation for the Humanities and Social Sciences (CSCI0931, 20)*
 Introduction to Compiler Construction (CSCI1260, 10?)
 Creating Modern Web Applications (CSCI1320, 77)
2012-2013 *Embedded and Real Time Programming (CSCI1600, 17)*
 Human Factors and User Interface Design (CSCI2310, 27)
 Creating Modern Web Applications (CSCI1320, 78)
2013-2014 *Designing, Evaluating and Creating User Interfaces (CSCI1950i, 50), co-taught with Jeff Huang (new course)*
 Creating Modern Web Applications (CSCI1320, 170)
2014-2015 *sabbatical*
2015-2016 *Compilers and Program Analysis (CSCI1260, 10)*
 Embedded and Real Time Programming (CSCI1600, 35)
 Creating Modern Web Applications (CSCI1320, 137)
2016-2017 *Embedded and Real Time Programming (CSCI1600, 37)*
 Topics in Software Engineering (CSCI2340, 10)
 Creating Modern Web Applications (CSCI1320, 130 est)
2017-2018 *Compilers and Program Analysis (CSCI1260, 11)*
 Embedded and Real Time Programming (CSCI1600, 15)
 Creating Modern Web Applications (CSCI1320, 150 est)
2018-2019 *Topics in Software Engineering (CSCI2340, 12)*
 Creating Modern Web Applications (CSCI1320, 120 est)
2019-2020 *Compilers and Program Analysis (CSCI1260, 8)*
 Embedded and Real Time Programming (CSCI1600, 13)
 Creating Modern Web and Mobile Applications (CSCI1320, 76)
2020-2021 *Software Engineering of Large Systems (CSCI1951U, 26)*
 Creating Modern Web and Mobile Applications (CSCI1320, 110 est)
2021-2022 *sabbatical*
2022-2023 *teaching relief*
2023-2024 *Graduate Software Engineering (CSCI2340, 31)*
2024-2025 *Graduate Software Engineering (CSCI2340, 51)*

Theses Directed

2014 Alexander Tarvo, “Automated Analysis of Multithreaded Programs for Performance Modeling”
2018 Qi Xin, “Towards Improving the Effectiveness of Automated Program Repair”

Date of Preparation: 12/31/2024