

John Regehr

School of Computing
50 South Central Campus Drive, Rm 3190
University of Utah
Salt Lake City, UT 84112-9205

regehr@cs.utah.edu
<http://www.cs.utah.edu/~regehr>
office +1 801 581 4280
fax +1 801 585 3743
home +1 801 220 0652

Education

- PhD, Computer Science, University of Virginia. Charlottesville, VA. Advisor: Prof. John A. Stankovic. Thesis title: “Using Hierarchical Scheduling to Support Soft Real-Time Applications on General-Purpose Operating Systems.” May 2001.
- Masters of Computer Science, University of Virginia. Charlottesville, VA. Advisor: Prof. Paul F. Reynolds. Project title: “An Isotach Implementation for Myrinet.” May 1997.
- BS, Computer Science, Kansas State University. Manhattan, KS. Honors advisor: Prof. Masaaki Mizuno. Honors thesis title: “Operating System Support for the Subsumption Architecture for Mobile Robot Control.” May 1995.
- BS, Mathematics, Kansas State University. Manhattan, KS. May 1995.

Employment

- Assistant Professor, School of Computing, University of Utah, August 2003–present.
- Adjunct Assistant Professor, School of Computing, University of Utah. September 2002–May 2003.
- Postdoctoral Fellow, School of Computing, University of Utah. Supervisor: Prof. Jay Lepreau. April 2001–July 2003.
- Research Intern, Systems and Networking Group, Microsoft Research, Microsoft Corporation, Redmond Washington. Mentor: Dr. Michael B. Jones. May–August 1998 and May–August 1999.
- Intern, Myricom Inc., Arcadia California. Supervisor: Dr. Bob Felderman. May–August 1997.
- Programmer, Department of Agronomy, Kansas State University. Supervisor: Prof. Stephen M. Welch. May 1992–May 1995.

Grants

1. DARPA Computer Science Study Group program. PI John Regehr. March 17 2008–March 16 2009. \$95,989.
2. National Science Foundation Embedded and Hybrid Systems Program. “Improving Sensor Network Software Reliability through Language, Tool, and OS Co-Design.” Award CNS-0615367. PIs John Regehr and Philip Levis (Stanford), co-PI Dawson Engler (Stanford). September 2006–August 2009. Utah award: \$210,000, Stanford award \$360,000.

3. National Science Foundation Parallel and Distributed Operation Systems Program. “Experimenting with Garbage Collection in an Otherwise Conventional OS.” Award CNS–0509526. PI Matthew Flatt, co-PI John Regehr. June 2005–April 2008. \$380,000.
4. National Science Foundation CAREER Program. “Vertically Integrated Analysis for Embedded Software.” Award CNS–0448047. May 2005–April 2010. \$400,000.
5. University of Utah Individual Teaching Grant. “Supporting Projects in Developing Embedded Software.” Awarded December 2004. \$2,700.
6. National Science Foundation Embedded and Hybrid Systems Program. “Components and Aspects for Embedded Middleware.” Award CNS–0410285. PIs Matthew Flatt and Raymond Klefstad (UC Irvine), co-PIs Eric Eide and John Regehr. September 2004–August 2007. Utah award: \$360,002, UC Irvine award \$330,000.
7. Hardware donation from Intel, May 2004. StrongARM-based development boards valued at \$16,000.
8. National Science Foundation Embedded and Hybrid Systems Program. “Composable Execution Environments: A Foundation for Building Robust Embedded Systems.” Award CCR–0209185. PI Jay Lepreau, co-PI John Regehr. July 2002–June 2005. \$310,000.

Research Participation

- Participated as a postdoctoral researcher from 2001–2003 in research performed under the Program Composition for Embedded Systems (PCES) DARPA IXO program. PI Jay Lepreau, co-PI Matthew Flatt. Program managers: Dr. Helen Gill, Dr. Douglas Schmidt, and Dr. Joseph Cross.

Courses

- Instructor, CS/ECE 5785/6785, Advanced Embedded Systems, University of Utah, Fall 2008. 20 students enrolled. 3 credit hours.
- Instructor, CS 6470, Advanced Compilers, University of Utah, Spring 2008. 7 students enrolled. 3 credit hours.
- Instructor, CS/ECE 5785/6785, Advanced Embedded Systems, University of Utah, Fall 2007. 26 students enrolled. 3 credit hours.
- Instructor, CS 7933, Seminar on Ultra Large Scale Systems, University of Utah, Fall 2007. 5 students enrolled. 1 credit hour.
- Instructor, CS/ECE 5785/6785, Advanced Embedded Systems, University of Utah, Fall 2006. 27 students enrolled. 3 credit hours.
- Instructor, CS 7962, Embedded Systems, University of Utah, Spring 2006. 17 students enrolled. 3 credit hours. *Developed a new set of laboratory assignments for ARM processors.*

- Instructor, CS 3400, Computer Systems (using the Bryant and O'Hallaron book), University of Utah, Fall 2005. 90 students enrolled. 4 credit hours.
- Instructor, CS 7938, Seminar on Program Analysis, University of Utah, Fall 2005. 11 students enrolled. 1 credit hour.
- Instructor, CS 7962, Embedded Systems, University of Utah, Spring 2005. 15 students enrolled. 3 credit hours. *This is a new course I developed, including a complete set of laboratory assignments based on sensor network nodes.*
- Instructor, CS 3400, Computer Systems (using the Bryant and O'Hallaron book), University of Utah, Fall 2004. 72 students enrolled. 4 credit hours.
- Co-instructor (with Sneha Kasera), CS 7940, Seminar on Sensor Networks, Fall 2004. 1–3 credit hours.
- Instructor, CS 4400, Computer Systems (using the Bryant and O'Hallaron book), University of Utah, Spring 2004. 107 students enrolled. 3 credit hours.
- Co-instructor (with Sneha Kasera), CS 6935, Seminar on Embedded and Networked Systems, Fall 2003. 1–3 credit hours.
- Instructor, CS 5460, Operating Systems, University of Utah, Fall 2002. 75 students enrolled. 3 credit hours. *Developed a new set of programming assignments for this course.*

(student authors' names are underlined)

In submission

1. Nathan Coopriider and John Regehr. Optimizing Interrupt-Driven Embedded Software. Submitted to ACM Transactions on Embedded Computing Systems, September 2008.

Journal Publication

1. John Regehr, Alastair Reid, and Kirk Webb. Eliminating stack overflow by abstract interpretation. ACM Transactions on Embedded Computing Systems, 4(4):751–778, November 2005.
<http://portal.acm.org/citation.cfm?id=1113830.1113833>

Conference Publications

1. Eric Eide and John Regehr. Volatiles are miscompiled, and what to do about it. To appear in *Proceedings of the ACM Conference on Embedded Software (EMSOFT)*, Atlanta, GA, October 2008. Acceptance rate: 25% (28/110).
<http://www.cs.utah.edu/~regehr/papers/emsoft08-preprint.pdf>
2. Venkat Chakravarthy, John Regehr, and Eric Eide. Edicts: Implementing Features with Flexible Binding Times. In *Proceedings of the 7th International Conference on Aspect-Oriented Software Development (AOSD)*, Brussels, Belgium, April 2008. Acceptance rate: 20% (17/79).
<http://www.cs.utah.edu/~regehr/papers/aosd08-preprint.pdf>

3. Nathan Coopriders, William Archer, Eric Eide, David Gay, and John Regehr. Efficient Memory Safety for TinyOS. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007)*, 14 pages, Sydney, Australia, November 2007. Acceptance rate: 17% (25/146).
<http://www.cs.utah.edu/~regehr/papers/coop-sensys07.pdf>
4. Nathan Coopriders and John Regehr. Offline Compression for On-Chip RAM. In *Proceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007)*, pages 363–372, San Diego, CA, June 2007. Acceptance rate: 25% (45/178).
<http://www.cs.utah.edu/~regehr/papers/pldi075-coopriders.pdf>
5. Will Archer, Philip Levis, and John Regehr. Interface Contracts for TinyOS. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN) 2007, SPOTS track*, pages 158–165, Cambridge, MA, April 2007. Acceptance rate: 26% (12/46) for presentation at the conference, 37% (17/46) for inclusion in the proceedings.
<http://www.cs.utah.edu/~regehr/papers/spots07.pdf>
6. Nathan Coopriders and John Regehr. Pluggable Abstract Domains for Analyzing Embedded Software. In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2006)*, pages 44–53, Ottawa, Canada, June 2006. Acceptance rate: 25% (21/83).
http://www.cs.utah.edu/~regehr/papers/lctes06_1
7. John Regehr and Usit Duongsaa. Deriving Abstract Transfer Functions for Analyzing Embedded Software. In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2006)*, pages 34–43, Ottawa, Canada, June 2006. Acceptance rate: 25% (21/83).
http://www.cs.utah.edu/~regehr/papers/lctes06_2
8. John Regehr. Random testing of interrupt-driven software. In *Proceedings of the ACM Conference on Embedded Software (EMSOFT)*, pages 290–298, Jersey City, NJ, September 2005.
<http://www.cs.utah.edu/~regehr/papers/emsoft05>
9. John Regehr and Usit Duongsaa. Preventing interrupt overload. In *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES 2005)*, pages 50–58, Chicago, IL, June 2005. Acceptance rate: 26% (25/95).
<http://www.cs.utah.edu/~regehr/papers/lctes05/>
10. John Regehr and Alastair Reid. HOIST: A system for automatically deriving static analyzers for embedded systems. In *Proceedings of the Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 133–143, Boston, MA, October 9–13 2004. Acceptance rate: 14% (24/169).
<http://www.cs.utah.edu/~regehr/papers/asplos04/>
11. Eric Eide, Tim Stack, John Regehr, and Jay Lepreau. Dynamic CPU Management for Real-Time, Middleware-Based Systems. In *Proceedings of the Real-Time Technology and Applications Symposium (RTAS)*, pages 286–295, Toronto, Canada, May 25–28 2004. Acceptance rate: 28% (23/81).
<http://www.cs.utah.edu/flux/papers/cpubroker-rtas04-base.html>

12. John Regehr, Alastair Reid, Kirk Webb, Michael Parker, and Jay Lepreau. Evolving real-time systems using hierarchical scheduling and concurrency analysis. In *Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS)*, pages 25–36, Cancun, Mexico, December 3–5 2003. Acceptance rate: 17% (24/145).
<http://www.cs.utah.edu/flux/papers/cee-rtss03/>
13. John Regehr, Alastair Reid, and Kirk Webb. Eliminating stack overflow by abstract interpretation. In *Proceedings of the Third International Conference on Embedded Software (EMSOFT)*, pages 751–778, Philadelphia, PA, October 15–17 2003. Acceptance rate: 33% (20/60).
<http://www.cs.utah.edu/flux/papers/emsoft03>
14. John Regehr. Scheduling Tasks with Mixed Preemption Relations for Robustness to Timing Faults. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, pages 315–326, Austin, TX, December 3–5 2002.
<http://www.cs.utah.edu/flux/papers/spak-flux-tn-02-01/>
15. John Regehr. Inferring Scheduling Behavior with Hourglass. In *Proceedings of the 2002 USENIX Annual Technical Conference FREENIX track*, pages 143–156, Monterey, CA, June 10–15 2002.
<http://www.cs.utah.edu/flux/papers/hourglass-usenix02/>
16. Eric Eide, Alastair Reid, John Regehr, and Jay Lepreau. Static and Dynamic Structure in Design Patterns. In *Proceedings of the 2002 International Conference on Software Engineering (ICSE)*, pages 208–218, Orlando, FL, May 19–25 2002.
<http://www.cs.utah.edu/flux/papers/knit-icse02-base.html>
17. John Regehr and John A. Stankovic. HLS: A Framework for Composing Soft Real-Time Schedulers. In *Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS)*, pages 3–14, London, UK, December 3–6 2001.
<http://www.cs.utah.edu/flux/papers/hls-rtss01/>
18. John Regehr and John A. Stankovic. Augmented CPU Reservations: Towards Predictable Execution on General-Purpose Operating Systems. In *Proceedings of the 7th Real-Time Technology and Applications Symposium (RTAS)*, pages 141–148, Taipei, Taiwan, May 30–June 1 2001.
<http://www.cs.utah.edu/~regehr/papers/augmented/>
19. Michael B. Jones, John Regehr, and Stefan Saroiu. Two Case Studies in Predictable Application Scheduling Using Rialto/NT. In *Proceedings of the 7th Real-Time Technology and Applications Symposium (RTAS)*, pages 157–164, Taipei, Taiwan, May 30–June 1 2001.
http://www.cs.utah.edu/~regehr/papers/rialtont_apps/
20. Michael B. Jones and John Regehr. CPU Reservations and Time Constraints: Implementation Experience on Windows NT. In *Proceedings of the 3rd USENIX Windows NT Symposium*, 10 pages, Seattle, WA, July 1999.
<http://www.cs.utah.edu/~regehr/papers/usenixnt99/>

Workshop Publications

1. Usa Sammapun, John Regehr, Insup Lee, and Oleg Sokolsky. Runtime Verification for Wireless Sensor Network Applications. In *Proceedings of the Dagstuhl Seminar 07011 on Runtime Verification*.
<http://drops.dagstuhl.de/portals/index.php?semnr=07011>
2. John Regehr and Phil Levis. High Confidence TinyOS. In *Proceedings of the Composable and Systems Technology for High Confidence Cyber-Physical Systems Workshop*, 3 pages, Arlington, VA, July 2007.
<http://www.cs.utah.edu/~regehr/papers/hccps07.pdf>
3. John Regehr, Nathan Coopriider, and David Gay. Atomicity and Visibility in Tiny Embedded Systems. In *Proceedings of the PLOS 2006 Workshop on Linguistic Support for Modern Operating Systems*, 4 pages, San Jose, CA, October 2006.
<http://www.cs.utah.edu/~regehr/papers/plos06b.pdf>
4. John Regehr, Nathan Coopriider, Will Archer, and Eric Eide. Efficient Type and Memory Safety for Tiny Embedded Systems. In *Proceedings of the PLOS 2006 Workshop on Linguistic Support for Modern Operating Systems*, 5 pages, San Jose, CA, October 2006.
<http://www.cs.utah.edu/~regehr/papers/plos06a.pdf>
5. John Regehr. Thread Verification vs. Interrupt Verification. In *Proceedings of the Workshop on Multithreading in Hardware and Software: Formal Approaches to Design and Verification (TV06)*, 5 pages, Seattle, WA, August 2006.
<http://www.cs.utah.edu/~regehr/papers/tv06.pdf>
6. John Regehr, Konrad Slind, and Elsa Gunter. Proofs as a substrate for tool integration supporting high-confidence embedded software. In *Proceedings of the High Confidence Medical Device Software and Systems (HCMDSS) Workshop*, 2 pages, Philadelphia, PA, June 2005.
<http://www.cs.utah.edu/~regehr/papers/hcmdss05.pdf>
7. John Regehr. Opportunities and Challenges for the Real-Time Community. Invited paper at the 2003 Workshop on Challenges in Embedded Real-Time Systems, Cancun, Mexico, December 2 2003.
8. John Regehr. Vertically Integrated Analysis and Transformation for Embedded Software. In *Proceedings of the 2003 Workshop on Compilers and Tools for Constrained Embedded Systems (CTCES)*, 8 pages, San Jose, CA, October 29 2003.
<http://www.cs.utah.edu/flux/papers/ctces03/>
9. John Regehr and Alastair Reid. Lock Inference for Systems Software. In *Proceedings of the Second AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, 6 pages, Boston, MA, March 17 2003.
<http://www.cs.utah.edu/flux/papers/lock-inference-03>

10. John Regehr and Jay Lepreau. The Case for Using Middleware to Manage Diverse Soft Real-Time Schedulers. In *Proceedings of the International Workshop on Multimedia Middleware (M3W)*, 5 pages, Ottawa, Canada, October 2001.
<http://www.cs.utah.edu/flux/papers/crm-m3w01/>
11. Michael B. Jones and John Regehr. The Problems You're Having May Not Be the Problems You Think You're Having: Results from a Latency Study of Windows NT. In *Proceedings of the 7th Workshop on Hot Topics in Operating Systems (HotOS)*, pages 96–101, Rio Rico, AZ, March 1999.
<http://www.cs.utah.edu/~regehr/papers/hotos7/>
12. Michael B. Jones and John Regehr. Issues in Using Commodity Operating Systems for Time-Dependent Tasks: Experiences from a Study of Windows NT. In *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 4 pages, Cambridge, England, July 1998.

Book Chapter

1. John Regehr. Safe and Structured Use of Interrupts in Real-Time and Embedded Software. Chapter in *Handbook of Real-Time and Embedded Systems*, CRC Press, 2007.
http://www.cs.utah.edu/~regehr/papers/interrupt_chapter.pdf

Other Papers

1. John Regehr and Nathan Cooprider. Interrupt Verification via Thread Verification. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 174(9):139–150, June 2007.
2. John Regehr. Teaching Reliability. *IEEE Distributed Systems Online*, vol. 7, no. 5, 2006, art. no. 0605-o5002.
3. John Regehr, Nathan Cooprider, Will Archer, and Eric Eide. Memory Safety and Untrusted Extensions for TinyOS. Technical Report number UUCS-06-007, School of Computing, University of Utah, June 2006.
4. John Regehr. Say No to Stack Overflow. *Embedded Systems Programming*, 17(10), October 2004.
<http://www.embedded.com/showArticle.jhtml?articleID=47101892>
5. John Regehr, Alastair Reid, Kirk Webb, and Jay Lepreau. Composable Execution Environments. Flux Group Technical Note 2002–02, May 2002.
6. Michael B. Jones and John Regehr. Predictable Scheduling for Digital Audio. Microsoft Research Technical Report MSR-TR-2000-87, December 2000.
7. John Regehr, John A. Stankovic, and Marty Humphrey. The Case for Hierarchical Loadable Schedulers With Performance Guarantees. Department of Computer Science, University of Virginia Technical Report CS-2000-07, March 2000.
8. John Regehr. An Isotach Implementation for Myrinet. Department of Computer Science, University of Virginia Technical Report CS-97-12, May 1997.

Poster

1. TinyOS 2.1: Adding Threads and Memory Protection to TinyOS. The TinyOS Alliance. Poster to be presented at The 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08). Raleigh, NC, November 2008

Patent

1. Providing predictable scheduling of programs using repeating precomputed schedules on discretely scheduled and/or multiprocessor operating systems. US patent 7,000,232. Michael B. Jones and John Regehr. Awarded February 2006.

Talks

1. "Static Analysis of Interrupt-Driven Embedded C." Invited talk at the Department of Computer Science, University of Virginia, April 7 2008.
2. "Static Analysis of Interrupt-Driven Embedded C." Invited talk at the Department of Computer Science, Washington University, January 18 2008.
3. "Static Analysis of Interrupt-Driven Embedded C." Invited talk at Australia's Information and Communications Technology Centre of Excellence (NICTA), November 7 2007.
4. "Using Application- and System-Specific Heuristics to Compile Embedded Software for Real-Time and Other Resource Constraints." Invited talk at the Department of Computer Science, University of Pittsburgh, November 5 2004.
5. "Say No to Stack Overflow." Invited talk at the Department of Computer Science, University of Maryland, June 8 2004.
6. "Say No to Stack Overflow." Invited talk at the Department of Computer Science, Brigham Young University, March 4 2004.
7. "Vertically Integrated Analysis and Transformation for Embedded Software." Invited talk at the 2003 Workshop on Constraint-Aware Embedded Software, Cancun, Mexico, December 2 2003.
8. "High-Level Optimizations for Low-Level Software." Presented as a work in progress at the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, New York, October 21 2003.
9. "High-Level Optimizations for Low-Level Software." Invited talk at the Department of Computer and Information Science, University of Pennsylvania, October 17 2003.
10. "Creating Embedded Software." Talk at the School of Computing, University of Utah, October 12 2003.
11. "Principles and Pragmatics for Embedded Systems." Job interview talk at Northwestern University, Indiana University, The University of Iowa, Purdue University, The Pennsylvania State University, North Carolina State University, The University of Delaware, The University of Utah, The University of Massachusetts at Amherst, and The University of Minnesota, February–April 2003.

12. “How to Rapidly Prototype a Real-Time Scheduler,” with Luca Abeni. Presented as a work in progress at the 23rd IEEE Real-Time Systems Symposium (RTSS), Austin, TX, December 4 2002.
13. “Some Guidelines for Proportional Share CPU Scheduling in General-Purpose Operating Systems.” Presented as a work in progress at the 22nd IEEE Real-Time Systems Symposium (RTSS), London, UK, December 5 2001.
14. “Hierarchical Schedulers, Performance Guarantees, and Resource Management,” with John A. Stankovic. Presented as a work in progress at the 17th ACM Symposium on Operating Systems Principles, Kiawah Island, South Carolina, December 1999.
15. “The Problems You’re Having May Not Be the Problems You Think You’re Having: Results from a Latency Study of Windows NT,” with Michael B. Jones. Invited talk at the Real-Time Applications and Systems Symposium, Vancouver, Canada, June 1999.
16. “Myricom and Linux,” with Bob Felderman. Invited talk at the First Extreme Linux Workshop, Santa Fe, New Mexico, February 1998.

Students Supervised

- Current PhD advisees
 - Will Archer, May 2005–present, graduation expected in August 2009
 - Yang Chen, August 2007–present
 - Nathan Coopriider, August 2004–present, successfully defended in August 2008 and is finalizing his thesis
 - Jianjun Duan, August 2007–present (co-advised with Prof. Konrad Slind)
 - Peng Li, May 2008–present
 - Xuejun Yang, January 2007–present
 - Lu Zhao, August 2007–present
- Current MS advisees
 - Rohit Pagariya, August 2008–present
- Past MS advisees
 - Venkat Chakravarthy — successfully defended August 2007 and is still working on finalizing the thesis
 - Usit Duongsaa — graduated May 2006
- Past non-thesis MS advisees
 - Jong Chun Park — graduated May 2004
 - Zhongxi Shen — graduated May 2005

- Current PhD committees
 - Anton Burtsev
 - Jessica Croft (ECE)
 - Eric Eide
 - Sachin Goyal
 - Guodong Li
 - Scott Little
 - Niti Madan
 - Matt Probst
 - Karthik Ramani
 - Subodh Sharma
 - Robert Thacker (ECE)
 - Sean Walton
 - Yu Yang
 - Chongkai Zhu
- Current MS committees
 - Ali Moinzaviri (CES)
- Past PhD committees
 - Kostadin Damevski — graduated August 2006
 - Scott Owens — graduated August 2006
 - Adam Wick — graduated August 2006
- Past MS committees
 - Siddharth Aggarwal — graduated December 2005
 - Abhijeet Joglekar — graduated January 2004
 - Parveen Patel — graduated December 2003
 - Prashanth Radhakrishnan – graduated December 2007

Publicly Released Software

- Safe TinyOS, as part of TinyOS 2.1.0
<http://tinyos.net>
- Hourglass: A Tool for Inferring Scheduling Behavior.
<http://www.cs.utah.edu/~regehr/hourglass>
2253 downloads between January 2005 and October 2007.

- Stacktool: A Static Analyzer for Measuring Stack Depth.
<http://www.cs.utah.edu/~regehr/stacktool>
1134 downloads between January 2005 and October 2007.
- SPAK: A Static Priority Analysis Kit.
<http://www.cs.utah.edu/~regehr/spak>
370 downloads between January 2005 and October 2007.
- cXprop: Static analysis, function inlining, and whole-program optimization for embedded C programs.
<http://www.cs.utah.edu/~coop/research/cxprop>
310 downloads between February 2006 and October 2007.

External Service

- Program Committee Chair
 1. ACM SIGPLAN/SIGBED 2008 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)
 2. IEEE 2006 Real-Time Technology and Applications Symposium (RTAS), track on “Development, Verification, and Debugging Tools for Real-Time and Embedded Systems”
- Program Committee Member
 1. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2009
 2. Real-Time Systems Symposium (RTSS) track on Wireless Sensor Networks 2008
 3. IEEE 17th International Conference on Computer Communications and Networks (ICCCN) 2008
 4. IEEE/IFIP International Conference On Embedded and Ubiquitous Computing (EUC) 2008
 5. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2008
 6. International Conference on Embedded Software (EMSOFT) 2007
 7. Workshop on the Interaction between Compilers and Computer Architecture (INTERACT) 2007
 8. International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) 2007
 9. Workshop on Embedded Sensor Networks (EmNets) 2007
 10. ACM SIGPLAN/SIGBED 2007 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)
 11. International Conference on High Performance Embedded Architectures and Compilers (HiPEAC) 2007
 12. Real-Time Systems Symposium (RTSS) 2007

13. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2007
 14. International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) 2006
 15. International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS) 2006
 16. Workshop on Operating Systems Platforms for Embedded Real-Time applications (2006)
 17. Workshop on Multithreading in Hardware and Software: Formal Approaches to Design and Verification (TV) 2006
 18. Real-Time Systems Symposium (RTSS) 2006
 19. Conference on Distributed Computing in Sensor Systems (DCOSS) 2006
 20. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS) 2006
 21. Java Technologies for Real-Time and Embedded Systems (JTRES) 2005
 22. Real-Time Systems Symposium (RTSS) 2005
 23. ACM SIGPLAN/SIGBED 2005 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)
 24. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2005
 25. Workshop on High Performance, Fault Adaptive, Large Scale Embedded Real-Time Systems (FALSE-II) 2005
 26. Workshop on Compilers and Tools for Constrained Embedded Systems (CTCES) 2004
 27. Real-Time Systems Symposium (RTSS) Track on Real-Time Middleware and Software Engineering 2004
 28. Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES) 2004
 29. Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS) 2004
 30. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2004
 31. Workshop on Java Technologies for Real-Time and Embedded Systems (JTRES) 2003
 32. Real-Time Systems Symposium (RTSS) 2003
 33. Real-Time Systems Symposium (RTSS) Work in Progress Session 2003
 34. International Conference on Distributed Computing Systems (ICDCS) 2003
 35. Real-Time and Embedded Technology and Applications Symposium (RTAS) 2003
- Other
 - Associate Editor, IEEE Distributed Systems Online, Real-Time and Embedded Systems Area, February 2006–December 2007
 - NSF Panelist: one panel in 2008, one panel in 2007, two panels in 2003
 - Work in Progress Co-chair, Real-Time Systems Symposium (RTSS) 2004
 - Student Travel Awards Chair, Conference on Embedded Networked Sensor Systems (SenSys) 2004

- Web Chair, ACM 2004 Conference on Languages, Compilers, and Tools for Embedded Systems
- Scribe, Symposium on Operating Systems Principles (SOSP) 1999

Internal Service

- Director, BS/MS program, 2005–2008
 - Gave 2–3 internal talks per year publicizing the program
 - Redesigned the admission process
 - Supervised the admission of new students in Spring 2006, 2007, and 2008
 - Spent approximately 10 hours per month advising students
 - Created a new web page for the program: <http://www.cs.utah.edu/bsms/>
- Facility Liaison, 2007–2008
 - Met with facility staff every 2–3 weeks to discuss pending issues and help them prioritize
 - Interviewed hourly employees to support the full-time staff, this resulted in hiring Adam Twede
- Member, robotics track committee, 2006–2008
- Member, computer engineering track committee, 2006–2008
- Member, curriculum committee, 2005–2008
- Member, graduate admissions committee, 2004–2005, 2005–2006, 2007–2008
- Organizer, Distinguished lecture series, 2004–2005, 2006–2008
- Member, systems faculty recruiting committee, 2007–2008
- Member, graphics faculty recruiting committee, 2007–2008
- Member, facility committee, 2006–2007
- Member, external relations committee, 2006–2007
- Member, undergraduate studies committee, 2003–2004