

John Hatcliff

Curriculum Vitae
October 2010

Professor	Phone: (785) 532-6350
Department of Computing and Information Sciences	FAX: (785) 532-7353
Kansas State University	email: hatcliff@cis.ksu.edu
Manhattan, KS 66506	WWW: www.cis.ksu.edu/~hatcliff
USA	

Education

Ph.D.: Computer Science, *Kansas State University*, Manhattan, Kansas, USA (1991-1994)

M.Sc.: Computer Science, *Queen's University*, Kingston, Ontario, Canada (1989-1991)

B.A.: Computer Science/Mathematics, *Mount Vernon Nazarene College*, Mount Vernon, Ohio, USA (1984-88)

Interests

Research Interests

Formal methods in software engineering, software verification, model-checking, static analyses of programs, partial evaluation and program transformations, concurrent and distributed systems, middleware, model-integrated computing, semantics of programming languages, compiler construction, logics and type theory.

Teaching Interests

Foundations of programming languages, software specification and verification, logic and set theory, construction of concurrent systems, compiler construction, formal language theory, software engineering, functional programming, logic programming.

Employment

2005–present: Professor, Department of Computing and Information Science, Kansas State University.

2001–2005: Associate Professor, Department of Computing and Information Science, Kansas State University.

1998–2000: Assistant Professor, Department of Computing and Information Science, Kansas State University.

1996–1998: Assistant Professor, Computer Science Department, Oklahoma State University.

1994–1996: Visiting Assistant Research Professor, Computer Science Department, Copenhagen University (DIKU).

1991–1994: Research assistant, Department of Computing and Information Science, Kansas State University.

1990–1991: Research assistant, Computer and Information Sciences Department, Queen’s University, Kingston, Ontario.

1989–1990: Teaching assistant, Computer and Information Sciences Department, Queen’s University, Kingston, Ontario.

Awards and Honors

2010: International Conference on Software Engineering (ICSE) ten-year retrospective *Most Influential Paper Award* for the paper “Bandera: extracting finite-state models from Java source code” published in ICSE 2000. The Most Influential Paper Award is given jointly by ACM & IEEE and the world’s premier software engineering conference (ICSE) to the paper that is judged to have had the most influence on the theory and practice of software engineering during the ten years since its original publication. According to the ACM Digital Library statistics, this paper is the most highly cited paper from among all papers that have appeared in the world’s largest and most prestigious software engineering conference series since its inception in 1975.

2010: Association for Computing Machinery (ACM) Special Interest Group on Software Engineering (SIGSOFT) *Impact Paper Award* for “Bandera: extracting finite-state models from Java source code”. ACM is the world’s largest educational and scientific computing society. The ACM SIGSOFT award is presented annually by the world’s primary professional organization of software engineers, to the author(s) of a paper presented at a SIGSOFT sponsored or co-sponsored conference that is judged to have had the most influence on the theory and practice of software engineering during the 10 years since its original publication. According to Google Scholar, this paper currently has more than 790 citations. Based on statistics from the ACM Digital Library, it is the second most highly cited paper out of the 10,600+ papers published throughout the 32-year history of ACM SIGSOFT-sponsored conferences and workshops.

2004: Kansas State University’s College of Engineering Researcher of the Year award given annually to the College of Engineering faculty with the most outstanding research contributions during the past five years.

2003: NASA Turning Goals Into Reality Award (one of 15 such awards given annually by NASA). Member of Java Path Finder research team which received the award for its ground-breaking work on software verification using model checking technology.

1997: NSF Early Career Award

Grants

1. Team Member (with PI Julian Goldman (Mass General Hospital, Harvard Medical School, Partners HealthCareOrg.) and others from industry and academia) Development of a Prototype Healthcare Intranet for Improved Health Outcomes. (US National Institutes of Health (NIH) / National Institute of Biomedical Imaging and Bioengineering (NIBIB) – Quantum Grant Program). *The prestigious NIH Quantum program funds projects that are viewed as “medical moonshots” that can potentially achieve a profound (quantum) improvement in health care. Prof. Hatcliff will lead*

project efforts related to system architecture, software validation, and certification. Total Amount: \$9.8million, KSU Portion: \$375,000 Duration: October 2010 – October 2015.

2. Principal Investigator (with co-PIs Robby, Gurdip Singh, Virg Wallentine, Steve Warren) Robby, Steve Warren). An Integrated Development and Certification Environment for a Medical Device Coordination Framework. (US National Science Foundation – NSF FDA Scholar in Residence Postdoctoral Funding – CNS 1065887). Amount: \$80,000. Duration: October 2010 – October 2011.
3. Principal Investigator (with co-PIs Daniel Andresen, Robby, Steve Warren). Infrastructure and Technology Innovations for Medical Device Coordination. (US National Science Foundation – CNS 0932289). NSF Collaborative Grant with the University of Pennsylvania. Total Amount: \$1,500,000, KSU Portion: \$839,548. Duration: September 2009 – August 2012.
4. Principal Investigator (with co-PIs Torben Amtoft, Anindya Banerjee, Robby, Simon Ou, Andrew Appel (Princeton University)). Evidence-based Trust in Large-scale MLS Systems (US Air Force Office of Scientific Research – Contract Number FA9550-09-1- 0138). Amount: \$3,000,000. Duration: May 2009 – August 2014.
5. Principal Investigator (with co-PIs Torben Amtoft, Simon Ou, and Robby) A Domain Specific Language for Defining High-Assurance Secure-Network Guards (Phase II) (Rockwell Collins Advanced Technology Center). Amount: \$85,000. Duration: September 2009 – August 2010.
6. Principal Investigator. Automatic Analysis Techniques for Discovering Information Flow Properties of Cryptographic Controllers (Rockwell Collins Advanced Technology Center). Amount: \$85,000. Duration: October 2008 – August 2009.
7. Principal Investigator (with co-PIs Torben Amtoft, Simon Ou, and Robby) A Domain Specific Language for Defining High-Assurance Secure-Network Guards (Rockwell Collins Advanced Technology Center). Amount: \$85,000. Duration: September 2008 – August 2009.
8. Principal Investigator. Conditional Information Flow Modeling for High-assurance Systems (Rockwell Collins Advanced Technology Center). Amount: \$25,000. Duration: October 2008 – November 2008.
9. Co-Principal Investigator (with Scott DeLoach (PI), Gurdip Singh, David Gustafson) A Testbed for Intelligent, Mobile Sensor Experiments. Air Force Office of Scientific Research (AFOSR/NM). Amount: \$219,140. Duration: 2007 – 2008.
10. Principal Investigator (with co-PIs Dan Andresen, Robby, Steve Warren). Development of an Open Test-bed for Application of Formal Methods to Plug and Play Medical Devices. (National Science Foundation CNS-0734204). Amount: \$55,000. Duration: September 2007 – August 2008.
11. Principal Investigator on Radical Innovations in Testing. (Lockheed Martin Advanced Technology Labs). Amount: \$60,000. Duration: Jan 2007 – December 2007
12. Principal Investigator (with co-PI Torben Amtoft) on Information Flow Modeling Analysis (Formalization and Supporting Tools for Secure Information Flow Certification of Industrial Applications). (Rockwell Collins Advanced Technology Center). Amount: \$50,000. Duration: Jan 2007 – December 2007

13. Principal Investigator on Principles and Tools for Rigorous Development and Integration of Component-based Systems. (Lockheed Martin Advanced Technology Laboratory). Amount: \$125,000. Duration: Jan 2006 – November 2006
14. Principal Investigator (with co-PIs Torben Amtoft and Anindya Banerjee) on An Integrated Specification and Verification Environment for Component-based Architectures of Large-scale Distributed Systems (Air Force Office of Scientific Research (AFOSR)). Amount: \$448,530. Duration: April 2006 – March 2009
15. co-Principal Investigator (with PI Gregg Rothermel, and co-PIs Matthew Dwyer, Sebastian Elbaum, Greg Rothermel) on CRI: Collaborative Research : A Community Resource to Support Controlled Experimentation with Program Analysis and Software Testing Techniques (NSF CNS 0454203). Amount: \$1,106,576. Duration: August 2005 – July 2009
16. Principal Investigator (with PI Matthew Dwyer, and co-PIs Sebastian Elbaum, Greg Rothermel) on Collaborative Research: Program Analysis Techniques to Support Dependable RTSJ Applications. NSF/NASA CCF-0429141. Amount: \$320,000. Duration: September 2004 – August 2007
17. Co-Principal Investigator (with co-PIs Matthew Dwyer, Dan Andresen, and Virgil Wallentine) on Parallel Analysis of Models for Distributed Real-Time Embedded Systems. Department of Defense (DURIP) Amount: \$394,991. Equipment Grant.
18. Principal Investigator (with co-PIs Matthew Dwyer and Gurdip Singh) on Integration of Tools for Model-driven Embedded System Designs with Next-Generation Real-Time Middleware. Subcontract to Lockheed-Martin on DARPA PCES. Amount: \$180,000. Duration: May 2004 – December 2004.
19. Principal Investigator (with co-PIs Matthew Dwyer and Gurdip Singh) on “Technologies, Development Tools, and Patterns for Automatic Generation and Customization of Adaptable DRE Middleware. DARPA - Program Composition for Embedded Systems.” Amount: \$1,161,215. Duration: two years, beginning May 2003.
20. Principal Investigator (with co-PIs Matthew Dwyer and Gurdip Singh) on Verification Tools for Model-driven Embedded System Designs. Subcontract to Lockheed-Martin on DARPA PCES. Amount: \$180,000. Duration: June 2003 – December 2003.
21. Co-Principal Investigator (with Matthew Dwyer (PI)) on “An Extensible Software Model-checking Framework”. NSF CCR-0306607. Amount: \$179,999. Duration: two years, beginning in June 1, 2003
22. Co-Principal Investigator (with Matthew Dwyer (PI) and George Avrunin) on “Integrated Software Model-checking”, Army Research Office, Amount: \$3,000,000. Duration: Five years, beginning Summer 2001.
23. Principal Investigator “Automatic Customization of Avionics Software”, Rockwell-Collins Advanced Technology Center. Amount: \$25,000. Duration: 1.25 years, beginning Fall 2001.
24. Principal Investigator “Educational Environment for Software Model-checking”, Rockwell-Collins University Grant. Amount: \$15,000. Duration: one year, beginning Fall 2002.

25. Co-Principal Investigator (with Matthew Dwyer (PI), Masaaki Mizuno, Mitch Neilson, Gur-dip Singh) on “Automatic Derivation, Integration and Verification of Synchronization Aspects in Object-Oriented Design Methods” DARPA Order K203/AFRL Contract F33615-00-C-3044 Amount: \$1,097,093. Duration: Four years, beginning Summer 2000.
26. Co-Principal Investigator (with Matthew Dwyer and Dave Schmidt (PI)) on “Automatic Model Construction for Finite-state Verification Applying Abstract Interpretation and Partial Evaluation Techniques”. DARPA/NASA Award NAG-02-1209 Amount: \$450,000. Duration: Two years, beginning Fall 1998.
27. Principal Investigator (with co-PIs Matthew Dwyer and Dave Schmidt) on “Automatic Model Construction for Finite-state Verification Applying Abstract Interpretation and Partial Evaluation Techniques”. NSF Post-doctoral Fellowship Amount: \$66,000. Duration: Two years, beginning Fall 1999.
28. Co-Principal Investigator (with Matthew Dwyer and Dave Schmidt (PI)) on “Integrating Platforms for Finite-State Verification”. NSF International Travel Award. Amount: \$15,467. Duration: Two years, beginning Fall 1999.
29. Principal Investigator on “A Partial Evaluation Tool Set for Automatically Customizing Adaptable Software”, *National Science Foundation Early Career Award*. Amount: \$200,000. Duration: Four years beginning Summer, 1997.

Publications

Edited Books/Proceedings

1. *Formal Techniques for Distributed Systems*, Proceedings of the Joint 12th IFIP WG6.1 International Conference, FMOODS 2010 and 30th IFIP WG6.1 International Conference, FORTE 2010, Amsterdam, The Netherlands, June 2010. John Hatcliff and Elena Zucca (editors) Lecture Notes in Computer Science (LNCS) 2619, Springer, 2010.
2. *Proceedings of the Ninth Annual Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2003)*, Warsaw, Poland. April 2003. Hubert Garavel and John Hatcliff (editors) Lecture Notes in Computer Science (LNCS) 2619, Springer-Verlag, 2003.
3. *Proceedings of the DIKU 1998 International Summerschool on Partial Evaluation*, Copenhagen, Denmark. John Hatcliff, Torben Mogensen, Peter Thiemann (editors). Lecture Notes in Computer Science (LNCS) 1706, Springer-Verlag, 1998.

Edited Journal Issues

1. Special Issue of *International Journal for Software Tools for Technology Transfer (STTT)* (Springer) (Volume 8, Number 1 / February, 2006) dedicated to selected papers from Ninth Annual Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2003), Hubert Garavel and John Hatcliff (editors)
2. Special Issue of *Theoretical Computer Science* (Volume 354, Issue 2) dedicated to selected papers from Ninth Annual Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2003), Hubert Garavel and John Hatcliff (editors)

Invited book chapters

1. “Specification and Checking of Software Contracts for Conditional Information Flow (extended version)”, Torben Amtoft, John Hatcliff, Edwin Rodriguez, Robby, Jonathan Hoag, and David Greve. Invited book chapter in *Design and Verification of Microprocessor Systems for High-Assurance Applications*, pp. 341–380. Springer, 2010. ISBN 978-1-4419-1538-2.
2. “Foundations of the Bandera Abstraction Tools”, John Hatcliff, Matthew B. Dwyer, Corina S.Puasquareanu, Robby. pp. 172 – 203. In ”The Essence of Computation – Essays dedicated to Neil Jones”. Lecture Notes in Computer Science 2566. Editors: Torben Mogensen, Hal Sudborough, Dave Schmidt
3. “Partial Evaluation”, Olivier Danvy, John Hatcliff, *Encyclopedia of Computer Science* Nature Publishing Group, UK, 2000.
4. “An Integrated Model-Driven Development Environment for Composing and Validating Distributed Real-Time and Embedded Systems”, Gabriele Trombetti, Aniruddha Gokhale, Douglas C. Schmidt, Jesse Greenwald, John Hatcliff, Georg Jung, Gurdip Singh, In *Model-Driven Software Development*, Beydeda, Sami; Book, Matthias; Gruhn, Volker (Eds.) 2005, ISBN: 3-540-25613-X, pp. 329–362.

Refereed Publications in Journals

1. Georg Jung and John Hatcliff. “A Type-centric Framework for Specifying Heterogeneous, Large-scale, Component-oriented, Architectures”, *Science of Computer Programming*, 75 (1) July 2010, pp. 615-637.
2. “A New Foundation For Control-Dependence and Slicing for Modern Program Structures”, Venkatesh Prasad Ranganath, Torben Amtoft, Anindya Banerjee, Matthew B. Dwyer, and John Hatcliff. *ACM Transactions of Programming Languages and Systems*. Volume 29, Issue 5, August 2007.
3. “An Event Correlation Framework for the CORBA Component Model”, Georg Jung, and John Hatcliff, *Journal of Software Tools for Technology Transfer* (Springer), 9 (5) October 2007, pp. 417-427.
4. “Slicing Concurrent Java Programs using Indus and Kaveri”, *Journal of Software Tools for Technology Transfer* (Springer), 9 (5) October 2007, pp. 489-504.
5. “High-Confidence Medical Device Software and Systems”, Insup Lee, George J. Pappas, Rance Cleaveland, John Hatcliff, Bruce Krogh, Peter Lee, Harvey Rubin, and Lui Sha. *IEEE Computer*, Vol. 39, No. 4, April, 2006, pp. 33-39.
6. “CALM and Cadena: Meta-modeling for Component-Based Product-Line Development”, Adam Childs, Jesse Greenwald, Georg Jung, Matthew Hoosier, and John Hatcliff, *IEEE Computer*, Vol. 39, No. 2, February, 2006, pp. 42-50.
7. “Exploiting Object Escape and Locking Information in Partial Order Reductions for Concurrent Object-Oriented Programs”, Matthew Dwyer, John Hatcliff, Venkatesh Ranganath, and Robby, *Journal of Formal Methods in System Design*, 25(2), Sep 2004, pp. 199-240.
8. “Translating Java for Multiple Model Checkers: the Bandera Back-End”, Radu Iosif, Matthew Dwyer, and John Hatcliff, *Journal of Formal Methods in System Design* (Kluwer), Volume 26, Issue 2 (March 2005), pp. 137–180.

9. “Expressing Checkable Properties of Dynamic Systems: The Bandera Specification Language”, James Corbett, Matthew Dwyer, John Hatcliff, Robby. *Journal of Software Tools for Technology Transfer*, 4(1):34-56, 2002. Springer-Verlag
10. “Weak Normalization Implies Strong Normalization in Generalized Non-dependent Pure Type Systems”, Gilles Barthe, John Hatcliff, Morten Heine Soerensen, *Journal of Theoretical Computer Science*, 269(1-2): 317–361, 2001.
11. “An Induction Principle for Pure Type Systems”, Gilles Barthe, John Hatcliff, Morten Heine Soerensen, *Journal of Theoretical Computer Science*, 266(1-2): 773-818, 2001.
12. “Slicing Software for Model Construction”, John Hatcliff, Matthew B. Dwyer, Hongjun Zheng, *Journal of Higher-order and Symbolic Computation*, 13(4), pp. 315–353, June, 2000. A special issue containing selected papers from the 1999 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation.
13. “Foundations for Partial Evaluation of Functional Languages with Computational Effects”, John Hatcliff, *1998 Symposium on Partial Evaluation, ACM Computing Surveys 30(3es)*, Sept, 1998.
14. “Using Partial Evaluation to Enable Verification of Concurrent Software”, Matthew B. Dwyer, John Hatcliff, and Muhammad Nanda, *1998 Symposium on Partial Evaluation, ACM Computing Surveys 30(3es)*, Sept, 1998.
15. “CPS Translations and Applications: the Cube and Beyond”, Gilles Barthe, John Hatcliff, Morten Heine Soerensen, *Journal of Higher-order and Symbolic Computation* 12(2), pp. 129 – 170, September, 1999.
16. “A Computational Formalization for Partial Evaluation”, John Hatcliff and Olivier Danvy, *Journal of Mathematical Structures in Computer Science*, vol 7, 1997, pp. 507 – 541. Special issue dedicated to the workshop on *Logic, Domains, and Programming Languages*, Darmstadt, Germany, May 1995.
17. “CPS Transformation after Strictness Analysis,” Olivier Danvy and John Hatcliff, *ACM Letters on Programming Languages and Systems*, 1(3):195-212, 1993.
18. “Thunks (continued),” Olivier Danvy and John Hatcliff, *Proceedings of the Workshop on Static Analysis WSA '92*. Bigre Journal, 81-82:3-11, 1992.
19. “Thunks and the λ -calculus,” John Hatcliff and Olivier Danvy, *Journal of Functional Programming*, 7(3), 1997, pp. 303 – 319.

Refereed Publications in Conference Proceedings

1. Torben Amtoft, John Hatcliff, and Edwin Rodriguez, “Precise and Automated Contract-based Reasoning for Verification and Certification of Information Flow Properties of Programs with Arrays”, Proceedings of the 2010 European Symposium on Programming (ESOP 2010), LNCS 6012, pp. 43–63, March 2010.
2. Andrew King, Sam Procter, Dan Andresen, John Hatcliff, Steve Warren, William Spees, Raoul Jetley, Paul Jones, Sandy Weininger. “A Publish-Subscribe Architecture and Component-based Programming Model for Medical Device Coordination and Integration”, SIGBED Review, Volume 6, Number 2, July 2009 Special Issue on the 2nd Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability.

3. Andrew King, Sam Procter, Dan Andresen, John Hatcliff, Steve Warren, William Spees, Raoul Jetley, Paul Jones, Sandy Weininger. “An Open Test Bed for Medical Device Integration and Coordination”, *Proceedings of International Conference on Software Engineering (ICSE 2009)*, (Software Engineering in Practice Track). ICSE-Companion 2009. 31st International Conference on Software Engineering, May 2009, pp. 141 - 151. IEEE Press, May 2009.
4. Torben Amtoft, John Hatcliff, Edwin Rodriguez, Robby, Jonathan Hoag, and David Greve, “Specification and Checking of Software Contracts for Conditional Information Flow”, *Proceedings of the 2008 International Conference on Formal Methods (FM '08)*, LNCS 5014, May 2008.
5. Georg Jung and John Hatcliff. “A Type-centric Framework for Specifying Heterogeneous, Large-scale, Component-oriented, Architectures”, *Proceedings of the 6th International Conference on Generative Programming and Component Engineering*, Salzburg, Austria. October, 2007. pp. 33–42.
6. “Kiasan/KUnit: Automatic Test Case Generation and Analysis Feedback for Open Object-oriented Systems”, Xianghua (William) Deng, Robby, John Hatcliff. *Proceedings of the 2007 International Conference on Testing: Academic and Industrial Conference: Practice and Research Techniques*, Windsor, England, pp. 3–12, IEEE Press, September 2007.
7. “Towards A Case-Optimal Symbolic Execution Algorithm for Analyzing Strong Properties of Object-Oriented Programs”, Xianghua Deng, Robby, and John Hatcliff. *Proceedings of the 2007 International Conference on Software Engineering and Formal Methods*, London, England, pp. 273–282, September 2007.
8. “Formal Software Analysis : Emerging Trends in Software Model Checking”, Matthew B. Dwyer, John Hatcliff, Robby, Corina S. Psreanu, Willem Visser. *Future of Software Engineering Track. Proceedings of the 2007 International Conference on Software Engineering (ICSE 2007)*. May, 2007.
9. “Bogor: A Flexible Framework for Creating Software Model Checkers”, Robby, Matthew Dwyer, and John Hatcliff, *Proceedings of the International Conference on Testing: Academic and Industrial Conference: Practice and Research Techniques*, IEEE Press, August 2006.
10. “An Overview of the Indus Framework for Analysis and Slicing of Concurrent Java Software”, Venkatesh Ranganath, John Hatcliff, *Proceedings of the 2006 IEEE Workshop on Source Code Analysis and Manipulation (SCAM)*. IEEE Press, September 2006.
11. “Context-Specific Middleware Specialization Techniques for Optimizing Software Product-line Architectures”, Arvind S. Krishna, Aniruddha Gokhale, Douglas C. Schmidt, John Hatcliff, and Venkatesh Prasad Ranganath, *Proceedings of EuroSys 2006*, Leuven, Belgium, April 18-21, 2006.
12. “A Case Study in Domain-Customized Model Checking for Real-Time Component Software”, Matthew Hoosier, Matthew B. Dwyer, Robby, and John Hatcliff, *Proceedings of the 2004 International Symposium on Leveraging Applications in Formal Methods (ISOLA 2004)*, Paphos, Cyprus, *Lecture Notes in Computer Science (4313)*, pp. 161–180.
13. “Evaluating the Effectiveness of Slicing for Model Reduction of Concurrent Object-Oriented Programs”, Matthew B. Dwyer, John Hatcliff, Matthew Hoosier, Venkatesh Ranganath, Robby, and Todd Wallentine, *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2006)*, Vienna, Austria, March, 2006, *Lecture Notes in Computer Science (3920)*, pp. 73 – 89.

14. “A New Foundation For Control-Dependence and Slicing for Modern Program Structures”, Venkatesh Prasad Ranganath, Torben Amtoft, Anindya Banerjee, Matthew B. Dwyer, and John Hatcliff. *Proceedings of the European Symposium On Programming (ESOP’05)*, Edinburg, Scotland, April 2005, Lecture Notes in Computer Science (3444), pp. 77–93.
15. “Kaveri: Delivering Indus Java Program Slicer to Eclipse Ganeshan Jayaraman, Venkatesh Prasad Ranganath, and John Hatcliff”. *Proceedings of the International Symposium on Fundamental Approaches to Software Engineering (FASE’05)* Edinburg, Scotland, April 2005, Lecture Notes in Computer Science (3442), pp. 269–272.
16. “Building Your Own Software Model Checker Using The Bogor Extensible Model Checking Framework”, Matthew B. Dwyer, John Hatcliff, Matthew Hoosier, Robby. *Proceedings of 17th Conference on Computer-Aided Verification (CAV 2005)*, Edinburgh, Scotland, July 2005, Lecture Notes in Computer Science (3576), pp. 148–152.
17. “Extending JML for Modular Specification and Verification of Multi-Threaded Programs”, Edwin Rodriguez, Matthew B. Dwyer, Cormac Flanagan, John Hatcliff, Gary T. Leavens, Robby. *Proceedings of 19th European Conference on Object-Oriented Programming (ECOOP 2005)*, Glasgow, Scotland, July 2005, Lecture Notes in Computer Science (3586), pp. 148–152.
18. “Checking Strong Specifications Using an Extensible Software Model-checking Framework”, Robby, Edwin Rodriguez, Matthew Dwyer, and John Hatcliff, *Proceedings of the International Conference on Tools and Algorithms for Construction and Analysis of Systems*, April, 2004. Lecture Notes in Computer Science (2988), pp. 404–420.
19. “SyncGen : An Aspect-oriented Framework for Automatically Generating Synchronization Implementations from High-level Specifications” John Hatcliff, William Deng, Matthew Dwyer, Masaaki Mizuno, *Proceedings of the International Conference on Tool and Algorithms for Construction and Analysis of Systems*, April, 2004. Lecture Notes in Computer Science (2988), pp. 158–162.
20. “An Event Correlation Framework for the CORBA Component Model”, Georg Jung, John Hatcliff, and Venkatesh Ranganath, *Proceedings of the International Conference on Fundamental Aspects of Software Engineering*, April, 2004. Lecture Notes in Computer Science (2984), pp. 144–159.
21. “Cadena : An Integrated Development Environment for Analysis, Synthesis, and Verification of Component-based Systems”, Adam Childs, Jesse Greenwald, Venkatesh Ranganath, Xinhua Deng, Matthew Dwyer, John Hatcliff, Georg Jung, Prashant Shanti Kumar, Gurdip Singh, *Proceedings of the International Conference on Fundamental Aspects of Software Engineering*, April, 2004. Lecture Notes in Computer Science (2984), pp. 160–164.
22. “Pruning Interference and Ready Dependence the Slicing Concurrent Java Programs”, Venkatesh Ranganath and John Hatcliff, *Proceedings of the International Conference on Compiler Construction*, April, 2004. Lecture Notes in Computer Science (2985), pp. 39–56.
23. “A Flexible Framework for the Estimation of Coverage Metrics in Explicit State Software Model Checking”, Edwin Rodriguez, Matthew B. Dwyer, John Hatcliff, Robby. *Proceedings of the 2004 International Workshop on Construction and Analysis of Safe, Secure, and Interoperable Smart Devices (CASSIS 2004)*, June 2004. Lecture Notes in Computer Science.
24. “Verifying Atomicity Specifications for Concurrent Object-Oriented Software using Model-Checking”, John Hatcliff, Robby, and Matthew Dwyer, *Proceedings of the 5th International Conference on Verification, Model Checking and Abstract Interpretation*, Venice, Italy. January, 2004.

25. "Space Reductions for Model Checking Quasi-cyclic Systems", Matthew Dwyer, Robby, John Hatcliff, and Xinhua Deng, *Proceedings of the 3rd International Conference on Embedded Software (EMSOFT 2003)*, Lecture Notes in Computer Science, Springer-Verlag, Oct., 2003.
26. "Bogor: An Extensible and Highly-Modular Model Checking Framework", Robby, Matthew B. Dwyer, John Hatcliff. *Proceedings of the 2003 ACM Conference on Foundations of Software Engineering*. Helsinki, Finland, September 2003.
27. "Space-Reduction Strategies for Model Checking Dynamic Software", Robby, Matthew Dwyer, John Hatcliff, and Radu Iosif in *Proceedings of the 2nd Workshop on Software Model Checking*, Electronic Notes in Computer Science, 89.3, June, 2003
28. "Model-checking Middleware-based Event-driven Real-time Embedded Software", William Deng, Matthew B. Dwyer, John Hatcliff, Georg Jung, Robby, Gurdip Singh. *Proceedings of the 2002 International Conference on Formal Methods for Components and Objects (FMCO 2002)*. Leiden, The Netherlands, November 2002 (invited paper).
29. "Slicing and Partial Evaluation of CORBA Component Model Designs for Avionics Systems", John Hatcliff, William Deng, Matthew Dwyer, Georg Jung, Venkatesh Ranganath, and Robby in *Proceedings of the 2003 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation*, June, 2003.
30. "Cadena: An Integrated Development, Analysis, and Verification Environment for Component-based Systems", John Hatcliff, William Deng, Matthew Dwyer, Georg Jung, Venkatesh Prasad. *Proceedings of the International Conference on Software Engineering (ICSE 2003)*. IEEE Press. Portland, Oregon, May 2003.
31. "Invariant-based Specification, Synthesis, and Verification of Synchronization in Concurrent Programs", Xianghua Deng, Matthew B. Dwyer, John Hatcliff, and Masaaki Mizuno. *Proceedings of International Conference on Software Engineering (ICSE 2002)*, IEEE Press, May 2002.
32. "Using the Bandera Tool Set to Model-check Properties of Concurrent Java Software", John Hatcliff and Matthew Dwyer. *Proceedings of 12th International Conference on Concurrency Theory (CONCUR 2001)*, August 2001, LNCS 2154, Springer-Verlag, pp. 39 – 58 (invited paper).
33. "Tool-supported Program Abstraction for Finite-state Verification", Matthew Dwyer, John Hatcliff, Corina Pasareanu, Robby, Willem Visser, Hongjun Zheng. *Proceedings of the International Conference on Software Engineering (ICSE 2001)*, May 2001, IEEE Press.
34. "Bandera : Extracting Finite-state Models from Java Source Code", James Corbett, Matthew Dwyer, John Hatcliff, Corina Pasareanu, Robby, Shawn Laubach, Hongjun Zheng. *Proceedings of the International Conference on Software Engineering (ICSE 2000)*, May 2000, IEEE Press.
35. "A Language Framework For Expressing Checkable Properties of Dynamic Software", James Corbett, Matthew Dwyer, John Hatcliff, Robby, *Proceedings of the 2000 Spin Workshop*. August, 2000. Stanford, CA, USA. LNCS 1885, pp. 205 – 223.
36. "Slicing Software for Model Construction", John Hatcliff, Matthew B. Dwyer. *Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*.

37. “A Formal Study of Slicing for Multi-threaded Programs with JVM Concurrency Primitives”, John Hatcliff, James C. Corbett, Matthew B. Dwyer, Stefan Sokolowski, and Hognjun Zheng, *Proceedings of the International Symposium on Static Analysis (SAS’99)*. Venice, Italy, September, 1999, LNCS 1694, pp. 1– 18.
38. “Staging Static Analyses Using Abstraction-based Program Specialization”, John Hatcliff, Matthew Dwyer, Shawn Laubach. *Proceedings of the 1998 Symposium on Programming Languages, Implementations, Logics and Programs (PLILP’98)*. Pisa, Italy, September, 1998. LNCS 1490, pp. 134 – 151.
39. “Introduction to Partial Evaluation Using a Simple Flowchart Language”, John Hatcliff, *Proceedings of the DIKU 1998 International Summerschool on Partial Evaluation*. LNCS 1706.
40. “Monadic Type Systems: Pure Type Systems for Impure Settings”, Gilles Barthe, John Hatcliff, and Peter Thiemann, *Proceedings of the Second HOOTS Workshop* (editors A. Gordon and A. Pitts and C. Talcott), Stanford University, Palo Alto, CA. December, 1997. Electronic Notes in Computer Science, Volume 10.
41. “Reflections on Reflections”, Gilles Barthe, John Hatcliff, Morten Heine Soerensen, *Proceedings of the Ninth International Symposium on Programming Languages, Implementations, Logics and Programs (PLILP’97)*, (editors H. Glaser, P. Hartel, and H. Kuchen) Pisa, Italy, September, 1997. LNCS 1292, pp. 241 – 258.
42. “An Approach to Classical Pure Type Systems”, Gilles Barthe, John Hatcliff, Morten Heine Soerensen, *Proceedings of The Thirteenth Annual Conference on Mathematical Foundations of Programming Language Semantics (MFPS XIII)* (S. Brookes and M. Mislove, editors), Pittsburgh, PA, March, 1997. Electronic Notes and Theoretical Computer Science, Volume 6.
43. “Reasoning about Hierarchies of Online Specialization Systems”, John Hatcliff and Robert Glück, *Proceedings of the Dagstuhl Seminar on Partial Evaluation*, February 1996, pp. 161–182. LNCS 1110.
44. “Generalization in Hierarchies of Online Program Specialization Systems”, Robert Glück, John Hatcliff, and Jesper Joergensen, *Proceedings of 1999 Conference on Logic-Based Program Synthesis and Transformation* (P. Flener, editor), pp. 197–198. LNCS 1559.
45. “Mechanically Verifying the Correctness of an Off-line Partial Evaluator,” John Hatcliff, *Proceedings of the Seventh International Symposium on Programming Languages, Implementations, Logics and Programs*, Utrecht, The Netherlands. September, 1995. Lecture Notes in Computer Science, Number 982.
46. “A Generic Account of Continuation-Passing Styles,” John Hatcliff and Olivier Danvy, *Proceedings of the 21st annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, Portland, Oregon, 1994.
47. “On the Transformation between Direct and Continuation Semantics,” Olivier Danvy and John Hatcliff, *Proceedings of the 9th Conference on Mathematical Foundations of Programming Semantics*, New Orleans, April 1993. Lecture Notes in Computer Science, Number 802.

Teaching Experience

1998–present: (Kansas State)

- Survey of Programming Languages (undergraduate) (F98,S99,F99,S00,F00)
- Foundations of Partial Evaluation and Program Specialization (graduate) (S00,S01)
- Specification and Verification of Reactive Systems (graduate) (F01,F02,S04,F04)
- Specification and Verification of Safety-Critical Systems (graduate) (F07)
- Software Specifications (graduate) (S01,S02,S03,S04,S06,S07,S08)

1996–1998: (Oklahoma State) Taught several graduate and undergraduate courses including *Organization of programming languages* (undergraduate), *Formal language theory* (graduate), *Advanced compiler writing* (graduate), *Foundations of adaptable software construction* (graduate).

1994–1996: (University of Copenhagen) Taught two graduate courses *Computation and deduction* and *Control, continuations, and extracting constructive content from classical proofs*.

Course Material Distributed

- *Specification and Verification of Reactive Systems* (with Matthew Dwyer and Robby)
 - **topic:** software model checking with an approach similar to that used in compiler courses – theory and practice of basic topics are taught by having students extend and adapt the Bogor model-checker developed by our research group.
 - **materials:** draft of course text book 1000+ lecture slides, course projects, homeworks, week quizzes and lab exercises.
 - **used by others:** Worcester Polytechnic, Oregon Graduate Institute, Dong Yang University (Korea), Queen’s University (Kingston, Ontario), University of Koblenz (Germany)
- *Software Specifications* (with Matthew Dwyer)
 - **materials:** 600+ lecture slides, course projects, model homework assignments, and exams.
 - **used by others:** University of Iowa (Spring 2003), Depaul University (Spring 2003), Emory University (Spring 2004), King Fahd University (Saudi Arabia – Fall 2003), Norht South University Dhaka, Bangladesh (Fall 2003), NED University of Engineering and Technology, Karachi-75270 Pakistan (Fall 2003), University of Montreal (Fall 2007)
- *An Introduction to Online and Offline Partial Evaluation Using a Simple Flowchart Language*
 - **materials:** 120 pages of type-set lecture notes, 101 lecture slides, course project in Scheme and Java
 - **used by others:** University of Copenhagen, University of Vienna (Austria), Waseda University (Japan), Hanyang University (Korea), and Ochanomizu University (Japan).
- *Programming Languages*
 - **materials:** 50 pages of type-set lecture notes (these supplement a text by Friedman, Haynes, and Wand), 500+ lecture slides, course project in Scheme and SML
 - **used by others:** Stevens Institute of Technology (New Jersey) – three semesters

Software Distributed

In the field of contributor science, distribution of software is an important method of research dissemination and impact, along with conference and journal publications. My research group has produced multiple software tools that have been used extensively in the world-wide research community in software engineering and program analysis.

- *Indus Java Slicing and Analysis Framework*
 - <http://indus.projects.cis.ksu.edu>
 - I supervised my PhD student Venkatesh Ranganath who has designed and developed this software. Indus is the only publicly available slicing framework for full Java.
 - **downloads:** 50,000+ since 2006, by individuals in over 30 countries. Indus is averaging around 20,000 downloads per year outside of KSU and is being used in a number of academic research projects and by two industrial research projects at Fujitsu Labs, Lockheed Martin (Indus development was funded by Lockheed Martin for one year), and Microsoft Research.
- *Bogor Software Model-checker*
 - <http://bogor.projects.cis.ksu.edu>
 - Bogor is one the premier model checkers available for object-oriented software. It has been used in numerous academic projects around the world as well as in industrial research projects at Lockheed Martin (Bogor development was funded by Lockheed Martin for two years) and Fujitsu. Matt Dwyer and I have supervised our PhD student Robby who has designed and developed this software. Robby is now a faculty member at Kansas State University.
 - **downloads:** Bogor has been downloaded to 7000+ individuals in over 25 different countries since initial release in September, 2003.
- *Bandera Verification Tool Set*
 - <http://bandera.projects.cis.ksu.edu>
 - Bandera was one of the first available model checkers. Matt Dwyer and I have jointly managed the development and helped with implementation of this software. Over 16 man-years have been invested in the effort with close to 300,000 lines of code.
 - **downloads:** 11000+ downloads outside of KSU to individuals in over 25 different countries, and to individuals at companies including Air France, Boeing, France Telecom, Gem Plus, Grammatech, Honeywell, IBM, Intel, Lucent, Marconi, SmartTrust, Sun, Synopsys, Toshiba, Rational (averaging 75 downloads a month since October 2003).
- *Cadena: An environment for Specification, Analysis, and Verification of Component-based Distributed Systems*
 - <http://cadena.projects.cis.ksu.edu>
 - I (along with Matt Dwyer and Gurdip Singh) have designed and managed the development of this software. Currently over 8 man-years have been invested in the effort with close to 50,000 lines of code.
 - **downloads:** 2400+ external downloads to individuals in over 20 countries since initial release in August 2003, and to individuals in companies including Lockheed Martin, Rockwell-Collins, BAE Systems, Barco Orthogon, and Boeing.

Ph.D. Students Supervised (degree completed)

1. Robby (2004), Current Employment: Assistant Professor, Kansas State University
2. Venkatesh Ranganath (2006), Current Employment: Research Engineer, Microsoft Research (Bangalore, India)
3. Georg Jung (2007), Current Employment: Lecturer, University of Potsdam (Germany)
4. William Deng (2007), Current Employment: Google (Mountain View, California)

Professional Activities

Conference Steering Committees

1. IFIP WG 6.1 International Conference on Formal Techniques for Distributed System (2010-current)
2. ACM Symposium on Partial Evaluation and Program Manipulation (PEPM) (2005-current)
3. Joint European Conference on Theory and Practice of Software (ETAPS) (2002)
4. Conference on Tools and Algorithms for the Construction and Analysis of Systems (2004)

Conference/Event Program Chairs/Organizer

1. 2010 IFIP WG 6.1 International Conference on Formal Techniques for Distributed Systems (FORTE/FMOODS) (Amsterdam, The Netherlands) – **co-chair** with Elana Zucca (University of Genoa, Italy)
2. 2008 ACM Workshop on Partial Evaluation and Program Manipulation (General Chair)
3. 2006 ACM Workshop on Partial Evaluation and Program Manipulation (PEPM 2006) (Charleston, SC) – **co-chair** with Frank Tip (IBM USA).
4. 2006 International School on Tools for Rigorous Engineering of Software Systems (STRESS), Dortmund Germany – **co-organizer** with Bernhard Steffen (Univ. Dortmund)
5. 2006 RTAS Workshop on Innovative Techniques for Certification of Embedded Systems (ITCES '06) San Jose, California, April 4, 2006.
6. 2003 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2003) (Warsaw, Poland) – **co-chair** with Hubert Garavel (INRIA).

Program Committees

1. 2011 ACM/IEEE International Conference on Software Engineering (ICSE) May 2011, Waikiki, Hawaii
2. 2010 International Conference on Run-time Verification (RV), Malta
3. 2010 International Symposium on Verified Software: Theories, Tools, Experiments (VSTTE), Edinburgh, Scotland.
4. 2010 Workshop on Software Engineering in Health Care (SEHC 2010), Cape Town, South Africa, May 2010

5. 2009 - 7th International Conference on integrated Formal Methods (iFM 2009), Duesseldorf, Germany.
6. 2008 - 15th International SPIN Workshop on Model Checking of Software (SPIN 2008), August 10-12, 2008, Los Angeles, USA.
7. 2008 - 10th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'08), June 4-6, 2008, Oslo, Norway.
8. 2007 ACM Symposium on Principles of Programming Languages (POPL 2007)
9. 2007 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2007) – part of the ETAPS 2007 Joint Conference, Braga, Portugal
10. 2006 International Summer School on Tool for Rigorous Engineering of Software Systems (STRESS), Dortmund, Germany, May 18-24, 2006. (co-chair with Bernhard Steffen, University of Dortmund)
11. 2006 RTAS Workshop on Innovative Techniques for Certification of Embedded Systems (ITCES '06) San Jose, California, April 4, 2006 (co-chair (with Oleg Sokolsky and Insup Lee from U Penn)
12. 2006 ACM SIGPLAN 2006 Workshop on Partial Evaluation and Program Manipulation (PEPM '06) Charleston, South Carolina, January 9-10, 2006. (co-chair with Frank Tip from IBM T.J. Watson Research Center)
13. 2006 TAIC PART – Testing: Academia and Industrial Conference - Practice and Research Techniques. Cumberland Lodge, Windsor, UK, 29th-31st August, 2006
14. 2005 Workshop on High Confidence Medical Device Software and Systems (HCMDSS), June 2-3, Philadelphia, PA.
15. 2005 International Workshop on Leveraging Applications of Formal Methods (ISoLA 2005), Sept. 24-25, Columbia, Maryland, USA.
16. 2005 International Workshop on Formal Techniques for Java-like Programs (FTfJP'2005). July 26, 2005. Glasgow, Scotland.
17. 2005 Workshop on Software Model Checking (SMC'05)
18. 2005 International Conference on Computer-Aided Verification (CAV'05)
19. 2005 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2005) (Edinburgh, Scotland)
20. Model Driven and Real-Time Embedded Systems track at 11th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2005), San Francisco, March 7-10, 2005.
21. International Symposium on Leveraging Applications of Formal Methods (ISoLA 2004)
22. 2004 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2004)
23. 2004 ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation (PEPM 2004)
24. 2004 ACM Conference on Foundations of Software Engineering (FSE)

25. 2004 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2004) (Barcelona, Spain)
26. SPIN'2004 Workshop (Barcelona, Spain)
27. 2003 ACM Symposium on Programming Languages Design and Implementation (PLDI'2003) (San Diego, CA)
28. 2003 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2003) – **co-chair** (Warsaw, Poland)
29. 2002 International Conference on Computer-Aided Verification (CAV'02) (Copenhagen, Denmark)
30. SPIN'2002 Workshop (Grenoble, France)
31. 2001 Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2001) (Genova, Italy)
32. SPIN'2000 Workshop (Palo Alto, California)
33. 2000 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM 2000) (Boston, Mass)
34. Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE 2000)
35. ICSE 2000 Workshop on Program Analysis, Testing and Verification (Limerick, Ireland)
36. Third ACM SIGPLAN Workshop on Continuations (CW'01) (Paris, France)

Invited Talks and Tutorials

1. ACM SIGSOFT Impact Paper Award Lecture, ACM Conference on Foundations of Software Engineering, Sante Fe, New Mexico, November, 2010.
2. Most Influential Paper Award Lecture, International Conference on Software Engineering (ICSE), Cape Town, South Africa, 2010.
3. Tenth International Conference on Formal Engineering Methods (ICFEM 2008) (one of three keynote speakers), Kitakyushu, Japan, October 27-31, 2008.
4. 2006 TAIC PART – Testing: Academia and Industrial Conference - Practice and Research Techniques. Cumberland Lodge, Windsor, UK, 29th-31st August, 2006. Talk topic: Model-checking concurrent software systems with Bogor. (Keynote Talk)
5. Sixth IEEE International Workshop on Source Code Analysis and Manipulation, Philadelphia, PA, USA, 27th-29th September 2006. Talk topic: Slicing concurrent Java programs with Indus. (Keynote Talk)
6. Programming Language Design and Implementation (PLDI 2005). Half-day tutorial on “Domain-specific model-checking with Bogor.” (with Matt Dwyer and Robby). Chicago, USA, June 2005.
7. Estonian Summer School in Computer and System Science (ESSCaSS), August 2004 (one of four invited lectures – 6 hours of lectures on Analysis and Verification of Embedded Software).

8. European Joint Conferences on Theory and Practice of Software (ETAPS 2004). Half-day tutorial on “Model-checking Software Systems with Bogor.” (with Matt Dwyer and Robby). Barcelona, Spain, April 2004.
9. ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation (PEPM 2003) (one of two keynote speakers).
10. International Symposium on Formal Methods for Components, Objects, and their Implementation (FMCO 2002). Leiden, The Netherlands, November, 2002. (one of fifteen invited research talks)
11. Schools on Formal Methods (SFM). September, 2002, Bertino, Italy. ”Software Model-checking”. (invited three hour lecture – one of eleven invited lecturers for a one-week international Ph.D. school)
12. European Joint Conferences on Theory and Practice of Software (ETAPS 2002). Full-day tutorial on “The Bandera Tool Set for Model-checking Concurrent Java Programs”. (with Matt Dwyer and Willem Visser)
13. International Conference on Mathematical Foundations of Programming Language Semantics (MFPS’02). March, 2002, New Orleans, LA. ”Model-checking Concurrent Java Software Using the Bandera Tool Set” (one of six key-note talks)
14. University of California at Berkeley. ”Model-checking Concurrent Java Software Using the Bandera Tool Set”. November, 2001.
15. CONCUR 2001: 12th International Conference on Concurrency Theory, Aalborg, Denmark, August 2001. (one of two invited 1.5 hour tutorials)
16. JavaCard Verification Project Meeting, August, 2001. INRIA, France. ”Model-checking Concurrent Java Software Using the Bandera Tool Set” (invited talk)
17. Workshop on Software Model-checking. July, 2001, Paris, France. ”Model-checking Concurrent Java Software Using the Bandera Tool Set” (one of two key-note talks)
18. BRICS, Aarhus University, Denmark, BRICS mini-course (six lectures) on software model-checking, October, 2000.
19. Danish Information and Technical University, Copenhagen, Denmark, “Model-checking Java Software with the Bandera Tool Set”, October, 2000.
20. University of Copenhagen, Copenhagen, Denmark, “Model-checking Java Software with the Bandera Tool Set”, October, 2000.
21. Carnegie-Mellon University, Pittsburgh, PA. “Model-checking Java Software with the Bandera Tool Set”, September, 2000.
22. Microsoft Research, Redmond, Washington. “Slicing and Abstraction in the Bandera Tool Set”, March, 2000.
23. Max Planck Institute Ringberg Workshop on Model-checking and Static Analysis, “Model-checking Java Software with the Bandera Tool Set”, Schloss Ringberg, Germany. February, 2000.
24. University of Massachussetts, “Object Flow Analysis for Java”, June, 1999.

25. INRIA, Sophia Antipolis, France, "A Formal Presentation of Slicing for a Language with Java Concurrency Primitives", September, 1999.
26. NASA Ames Research Lab, "Slicing and Abstraction in the Bandera Tool Set", November, 1999.
27. Invited Speaker for European Research Consortium for Informatics and Mathematics (ERCIM) working group on "Programming Language Technologies" Kickoff Meeting. Pisa, Italy. September, 1998.
28. Invited Lecturer for 1998 DIKU International Summer School on Partial Evaluation, University of Copenhagen, Denmark. July, 1998.
29. University of Copenhagen, "Reflections on Reflections", June 1997.
30. University of Kansas, "Automatically Customizing Adaptable Software Using Partial Evaluation", January, 1997.
31. Kansas State University, "Automatically Customizing Adaptable Software Using Partial Evaluation", March, 1997.
32. On-Line Library Corporation, Columbus, Ohio, "Partial Evaluation and Program Specialization", March, 1996.
33. Invited participant 1996 Dagstuhl Seminar on Partial Evaluation, Dagstuhl, Germany, February, 1996, "Reasoning about Hierarchies of Online Specialization Systems".
34. BRICS, Aarhus University, Denmark. "The Structure of Continuation-Passing Styles," October, 1994.
35. Indiana University, "Aspects of Evaluation Orders in Continuation-Passing Style," November, 1993.
36. Carnegie-Mellon University, "On the Transformation between Direct and Continuation Semantics," June, 1993.