Jan Vitek

Curriculum Vitae

March 1, 2011

jv@cs.purdue.edu
Fax: (765) 494 0739
Phone: (765) 494 6531
Nationality: Swiss

Purdue University
250 N. University Street
West Lafayette IN, 47906 USA
Date of Birth: 6.9.1966

Research Interests

Software engineering: automated tools for assurance & reliability, static program analysis. **Real-time systems:** high-level real-time languages, virtual machines and real-time garbage collection. **Programming languages:** type systems, object-oriented programming. **Concurrency and Distribution:** transactional memory, models of concurrency, mobile programming languages. **Software security:** information flow, static security enforcement.

Employment

Professor 8/2010 –
Computer Science, Purdue University, West Lafayette, IN, USA

Chief Scientist 6/2009 –
Fiji Systems LLC, Indianapolis, IN, USA

Academic Visitor 9/2006 – 5/2011
IBM T.J Watson Research, Hawthorne, NY, USA

Associate Professor 8/2005 – 8/2010
Computer Science, Purdue University, West Lafayette, IN, USA

Visiting Professor 7/2008 – 9/2008
INRIA Rocquencourt, Paris, France

Visiting Professor 1/2006 – 7/2006
Computer Science, Ecole Polytechnique Fédérale de Lausanne, CH

Assistant Professor 8/1999 – 7/2005
Computer Science, Purdue University West Lafayette, IN, USA

Research Assistant 9/1994 – 7/1999
Centre Universitaire d'Informatique, University of Geneva, CH 9/1989 – 8/1990

Software Consultant 8/1992 – 8/1993
International Labor Organization, CH

Research Assistant 9/1990 – 7/1992
Computer Science, University of Victoria, CA

Education

University of Geneva — Geneva, Switzerland
Ph.D. in Information Systems, *1999*
Dissertation: *The Seal Calculus – A calculus of mobile computations*
Advisor: Dennis Tsichritzis

University of Victoria — Victoria, Canada
M.S. in Computer Science, *1995*
Dissertation: *Compact Dispatch Tables for Dynamically Typed Languages*
Advisor: R. Nigel Horspool

University of Geneva — Geneva, Switzerland
B.S. in Information Systems, *1989*

Awards and Honors

2011: *Purdue University Faculty Scholar (2011–2016).*

2011: *Microsoft Research Award, SEIF, 2011.*

2011: *Purdue College of Science Undergraduate Advising Award.*

2006: *IBM Faculty Award.*

2001: *National Science Foundation CAREER Award.*

1989: *Outstanding Bachelor's prize, University of Geneva.*

Professional and Scholarly Associations

1. Vice President of "Association Internationale pour les Téchnologies Objets", since 2010.
2. Elected member of the International Federation for Information Processing (IFIP) Working Group 2.4 "Software Implementation Technology".
3. Member of the Java Expert Group for JSR 302: Safety Critical Java Technology.
4. Member of "Association Internationale pour les Téchnologies Objets", Steering Committee of the European Conference on Object-Oriented Programming (ECOOP), since 2008.
5. Member of the Steering Committee of the International Conference on Coordination Models and Languages (COORDINATION), 2007–2010.
6. Member of the Steering and Organizing Committee for the International Summer School on Trends in Concurrency, since 2006.
7. Member of the Steering Committee of the ACM/USENIX Conference on Virtual Execution Environments, 2005–2010.
8. Founding Member of the Steering Committee of the ACM TRANSACT Worshop, since 2005.

9. Member of the Steering Committee of the Jave Technologies for Real-time and Embedded Systems (JTRes) workshop, since 2005.

10. Member, Association for Computer Machinery (ACM).

1 DISCOVERY

Dr. Vitek's research focuses on computer programming languages. The interface between user and machine. In his early work, he focused on implementation techniques for object-oriented languages and invented schemes for performing two of the most common operations in those languages, namely method invocation and subtype tests, more efficiently. Dr. Vitek is one of the inventors of the concept of ownership types, a key technique for controlling aliasing and sharing in pointer-based languages. This has applications in software engineering and information security. For instance, he showed how to use a lightweight ownership discipline, called confined types, to fix a security vulnerability in the Java programming environment. In recent work, Dr. Vitek has investigated high-level languages for embedded computing systems and in particular systems with hard real-time constraints. This led to the development of the first open source real-time Java virtual machine and the first flight test of Java on an unmanned aerial vehicle in collaboration with Boeing. Some of the technology developed in his group was adopted by IBM and also led to a spin-off company specializing in Java for mission-critical systems.

Confinement, Ownership and Heap Separation

Object identity is the foundation of object-oriented programming. Objects are used to model application domain entities. Even if the state or behavior of an object changes, the same object always represents the same phenomenon in the application domain. However, identity causes practical problems due to the presence of *aliasing*, when a particular object can be referred to by any number of others. A change to a specific object can affect all objects that refer to it, even though the object being updated may have no information about the referring objects themselves. In the presence of aliases, understanding the function of a program becomes more complex as runtime information about topology of the system is required. Working with James Noble and John Potter, Dr. Vitek laid the foundations of an approach that came to be known as *ownership types* [74]. The seemlingly simple idea of encoding constraints on the runtime structure of the heap into class definitions was highly influencial with hundreds of citations and many research offshoots. The 1998 ECOOP paper [74] work was formalized by Clarke, and used by Boyapati, Rinard and Liskov amongst others. While ownership types have proven to be a successful vehicle for experimenting with new language features, their practical adoption has been hampered by the extra overhead inherent to the fine-grained annotations needed to define an encapsulation policy. This observation led to the lighter weight notions of confinement presented in [20]. Ideally, confinement should simply codify software engineering best practice already familiar to programmers. [20] demonstrates a non-trivial encapsulation property which requires very few annotations. Confinement can be inferred as shown in [67]. Using inference on 100MB of Java code shows that over a third of the candidate classes are confined [8]. This is encouraging as it suggests that experienced programmers tend to write code that does not violate confinement. It is thus reasonable to expect that confinement can be widely adopted as a way to reinforce good programming style. Dr. Vitek proved soundness of confined types for a core object-oriented calculus [61], [11] and is developing a type system inspired by confined types for region allocation in Real-time Java programs in [4], [6], [46]. Vitek further demonstrated applications of the ideas in his work on StreamFlex [2], [35], [36], [39] and for enforcing thread locality [34]. A variant of ownership types is considered for adoption in the upcoming Safety Critical Java standard to statically prevent memory access errors.

Programming Language Implementation

Object-oriented programming has become the major paradigm for new software development. Many modern software systems are developed in Java, C++ or C#. Object-oriented languages present a number of practical challenges. Dr. Vitek's contributions to this area are novel algorithms for the two most frequent operations, method dispatch and subtype tests. Method dispatch is the operation which selects, each time a function is called, the instruction sequence to be executed. Early dispatch algorithms were an important source of inefficiency and, for many years, an impediment to adoption of object-oriented languages. Dr. Vitek proposed a compact solution to the dispatching problem in dynamically typed languages such as Smalltalk [80] and later further reduced space requirements [78]. [79] assessed the impact of modern pipelined architectures on the performance of several dispatch algorithms. This work was subsequently used by several authors, including, Driesen, Royer, Gil and Zvibin. In later work Dr. Vitek investigated flexible dispatching policies [50], [5]. The second most frequent object-oriented operation involves dynamically computing whether two elements are related by a partial order, known as a subtype test. Subtype tests are executed in a variety of contexts, such as checked casts, array updates, and exception handling. While subtype tests can be performed in linear time by traversing the type hierarchy, their performance is not acceptable in practice. The research question is thus to find data structures, optimized for space and time, for answering reachability queries on a lattice. [76] proposed a compact bitset encoding of the type hierarchy based on graph coloring. [75] presented a constant time algorithm that adapted Dr. Vitek's method dispatch work to the task of compacting a matrix encoding of the type hierarchy. [63] presented a constant time subtype test algorithm with a very low space footprint and fast lattice update times. This algorithm is suited to systems requiring predictable performance and runtime loading of new software modules.

Dr. Vitek led the Ovm project which delivered an open source framework for building language runtimes. Ovm is a toolkit that provides the basic components of a virtual machine capable of executing Java programs. By design, the components can be specialized and assembled into a configuration which is customized for a particular problem domain. For example, it can be used to develop a Real-Time Java configuration which yields a virtual machine implementing part of the Real-time Specification for Java. The system have a code base of over 150KLoc and involved approximately 20 man/years of effort in its development. It led to publications in virtual machines and memory managment [2], [3], [7], [14], [35], [37], [39], [40], [41], [42], [45], [46], [47], [48], [49], [64].

High-Assurance Real-Time Computing

Dr. Vitek has started a major research effort to raise the abstraction level for hard real-time systems. Ovm has been used to build a real-time Java virtual machine that targets avionics applications [7], [48] and aerospace [33]. Recently, Prof. Vitek's team has developed StreamFlex – a programming model inspired by real-time Java but targeting real-time stream processing systems. In [2] they have shown that periods of 50 μs can be reliably achieved without losing portability or the memory safety guarantees of Java. StreamFlex was further developed in [35], [39], [41] and offers a dataflow programming model with zero copy guarantees for streaming data and a novel use of software transactional memory [49] for communication with non-real-time tasks. The StreamFlex technology was transfered to an open source product from IBM called Flexotask [36].

Professors Vitek and Leavens have recently started a collaboration that aims to formalize the guarantees needed for the development of safety critical applications in Java [88], [92]. Part of this work is being done in the context of the JSR-302 Safety Critical Java expert group on which Prof. Vitek serves. The infrastructure goal of the proposal is to extend the work

done on Ovm and implement a Safety Critical subset of Java that will be open sourced. The scientific goal of the project is to extend the JML formal specification notation to support assertions about worst case execution time of Java components. The project is still in the early stages, but the current direction is to work at the specification level.

Vitek is also investigating real-time memory management techniques [57], [44], [42], [40], [37], [31],[3], [89], [26]. The main result has been the development of a novel defragmenting real-time concurrent garbage collection algorithm that does not unduly deteriorate the performance real-time code. Traditionally, real-time developers have been leery of using garbage collection (even though it has been shown to reduce development costs substantially). This because of the lack of predictability of traditional collector and/or of the massive throughput overhead of real-time collectors. Furthermore, to avoid the danger of fragmenting memory real-time programmers prefer pre-allocate all the system's data. This is neither efficient in terms of memory nor very reliable. Schism, a new collector developed at Fiji Systems (a startup on which Vitek is chief scientist) provably never preempts or interferes with user code and imposes a worst case run-time overhead of 40% over native C. This overhead is inclusive of all costs of running Java (array bounds checks, null checks, methods, barriers). This results places Java firmly within the same league as low-level system languages.

Information Assurance and Security

Software systems are evolving. Monolithic applications are being replaced by off-the-shelf components-based architectures. Instead of a small number of large programs from well-established suppliers, a user's desktop is now made up of many smaller applications and software modules that interact in intricate ways to carry out a variety of information processing tasks. Moreover, whereas a software base used to be fairly static, it is now easy to download code from the network and even extend application programs as they execute. In such fluid environments, traditional security mechanisms and policies appear almost irrelevant. While passwords and access control mechanisms are adequate to protect the integrity of the computer system as whole, they utterly fail to protecting the user from downloaded code being run from his account. Although it may not be feasible to analyze all third-party software packages, one can intercept the communications between a package and the other parts of the system by interposing code at the boundaries of the different components. Interposition techniques effectively encapsulate untrusted components in wrapper programs that have full control over the interactions between encapsulated components and the OS and over the interactions among components. Dr. Vitek looked at the formal foundation of secure composition using wrappers, focusing on the rigorous statement and proof of their security properties. To reason about wrappers he developed a small programming language that allows the composition of concurrently-executing software components and supports the enforcement of security policies. The essential aspects of the problem were abstracted in a process calculus: the box-π calculus, a minimal extension of the π-calculus with encapsulation [68], [18], [69]. Wrappers are only able to enforce local security policies. Working with Dominic Duggan, Dr. Vitek defined distributed access control as a weak form of information flow control. They associate access restrictions with data in perpetuity as the data propagates through the system. As data is produced based on the content of other data, the produced data inherits the access restrictions of the consumed data, and may have additional access restrictions. This is weaker than information flow control as it deliberately ignores "covert channels," focusing only on the enforcement of access control restrictions. Also, rather than enforcing mandatory access control, distributed access control is discretionary, in the sense that principals can collaborate to release previously classified data [62], [60]. Together with Spafford and Gapolakrishna, Dr. Vitek implemented an efficient intrusion detection technique

for based on a reference monitor at operating system call level [52]. Finally, Vitek worked on a secure coordination model for distributed systems [19]. The language, Secure Spaces, extends the coordination language Linda with fine-grained access control. Secure spaces answer to the difficulty of engineering a comprehensive security architecture that enforces the security requirements of a variety of applications without being overly restrictive.

Mobile Computation

Wide area networks (WAN) are fundamentally different from the centrally administered local area networks of the past. They are spread over administrative domains with different security policies, and they enjoy vfluctuations connectivity and bandwidth. To address the changes in distributed computing, Dr. Vitek devised new abstractions for programming WANs. In collaboration with G. Castagna, he defined the Seal calculus, a core model of mobile computational systems to reason about the behavior of distributed mobile programs. He designed and implemented a language called JavaSeal [21],[121], [123], [70], [72], [100], [102], [105] which endows program logic with location-awareness and with control over the logical and physical location of computation. Seal was among the first attempts to explore the design space of mobile programming languages from both theoretical and practical angles [15],[99],[104]. Seal is a contemporary of the Ambient Calculus. The two calculi differ most in their mobility models, and Seal's hierarchical control model turned out to be more appropriate to restrict behavior of untrusted code. Seal influenced the work of a number of researchers in the field. Location mobility as a result of process synchronization, first introduced in Seal as a natural extension of π-calculus communication primitives, was adopted by Levi and Sangiorgi, and in several other works. Seal influenced the design of Boxed Ambients, a variant of Ambients obtained by suppressing the open capability. The communication primitives of Seal were adapted to Boxed Ambients in order to ease static detection of insecure information flows. The difference in expressiveness of the communication primitives of variants of Seal is at the origin of the New Boxed Ambients calculus, which enriches the shared channel communication of Seal with name capturing receiving actions. Seal has inspired the Calculus of Mobile Resources which inherits the interaction pattern for exiting agents, and mobility on anonymous content of locations. It also played a role in the design of the Crypto-Loc calculus which has a similar communication model. Finally, Seal primitives are also at the basis of box-π, a language for securely integrating trusted and untrusted off-the-shelf components.

Computational Molecular Biology

Problems in computational molecular biology require novel algorithmic solutions and implementation techniques. Dr. Vitek contributed to the field of Nuclear Magnetic Resonance spectroscopy, a key experimental method used to study structure and interactions of the proteins. In particular, he worked on an inferential algorithm for backbone resonance assignment which defines the mapping between atoms in the protein and peaks in the resonance spectra. He developed a branch-and-bound exhaustive search algorithm based on a Bayesian statistical model that accounts for various sources of uncertainty and provides an automated framework for inference. The algorithm successfully handles search spaces much larger than the spaces searched by the existing algorithms of exhaustive search [9], [12], [16], [51].

2 Grants

1. Principal Investigator (100%) on *Automated Generation of JavaScript Workloads.*
   Mozilla Corporation. Amount: $75,000. Duration: February 2012.

2. Principal Investigator (50%) on *SI2-SSE: A Tracing Virtual Machine For Statistical Computing.*
   NSF. Amount: $489,084. Duration: September 2010 – August 2013.

3. PI (100%) on *EAGER: Foundations of Data-Centric Concurrency Control.*
   NSF. Amount: $110,000. Duration: August 2010 – September 2011.

4. PI (100%) on *VEESC: Virtual Execution Environments for Scientific Computing Workshop.*
   NSF. Amount: $45,000. Duration: August 2010 – September 2010.

5. PI (100%) on *Third International Summer School on Trends in Concurrency.*
   NSF. Amount: $12,000. Duration: May 2010 – September 2010.

6. Co-PI (with Suresh Jagannathan) (50%) on *II-New: A Computational Infrastructure for Scalable Transactional Memory Abstractions in Managed Languages.*
   NSF Amount: $536,000. July 2010 –.

7. Senior Personnel (with Fiji Systems) (0%) on *SBIR Phase I: Low-power Real-time Java for Mission-critical Systems.*
   NSF Amount: $100,000. June 2009 – June 2010.

8. Co-PI (with Gary Leavens) (50%) on *SHF: Specification and Verification of Safety Critical Java.*
   NSF Amount: $500,000. August 2009–September 2011.

9. Principal Investigator (100%) on *A Computational Model for High-Assurance Dynamic Information Systems.*
   ONR Amount: $200,000. March 2009–December 2009.

10. Principal Investigator (with Tony Hosking) (50%) on *CPA-CPL Certified Garbage Collection for Highly Responsive Systems.*
    NSF Amount: $498,952. August 2008–July 2011.

11. Co-PI (with Suresh Jagannathan) (50%) on *CSR/AES: Fault Determination and Recovery in Cycle Sharing Infrastructures (Supplement).*
    NSF Amount: $23,000. May 2008.

12. Principal Investigator (with Suresh Jagannathan, Dan Grossman and Maurice Herlihy) on *CPA-SEL-T: Collaborative Research: Unified Open Source Transactional Infrastructure.*
    National Science Foundation. Amount: $1,000,000. Duration: September 2008 – September 2011.

13. Co-Principal Investigator (with Ananth Grama, Tony Hosking, Suresh Jagannathan on *Language and Runtime Support for Safe and Scalable Programs.*
    Microsoft Research. Amount: $200,000. Duration: June 2008.

14. Co-Principal Investigator (with Rachid Guerraoui, EPFL) on *Soft Integration of Hard Real-Time Capabilities in C#.*
    Microsoft Research. Amount: $400,000. Duration: July 2008.

15. Co-Principal Investigator on *Second International Summer School: Trends in Concurrency.*
    National Science Foundation. Amount: $23,000. Duration: March 2008.

16. Principal Investigator on *Second International Summer School: Trends in Concurrency.*
    IBM Research. Amount: $1,000. Duration: January 2008.

17. Principal Investigator on *Second International Summer School: Trends in Concurrency.*
    Microsoft Research. Amount: $10,000. Duration: February 2008.

18. Principal Investigator on *Second International Summer School: Trends in Concurrency.*
    Intel Research. Amount: $5,000. Duration: December 2007.

19. Principal Investigator on *CSR-EHS: High-throughput Real-time Stream Processing in Java.*
    National Science Foundation. Amount: $210,000. Duration: September 2007 – August 2010.

20. Co-Principal Investigator (with: Suresh Jagannathan) on *CT-ER: Controlled Declassification with Software Transactional Memory.*
    National Science Foundation. Amount: $249,857. Duration: September 2007 – September 2009.

21. Principal Investigator on *IBM Faculty Award.*
    IBM. Amount: $30,000. Duration: September 2006.

22. Principal Investigator on *High-level Concurrency Control Abstractions Methodologies, Languages and Runtimes.*
    Microsoft Research Award. Amount: $50,000. Duration: October 2006.

23. Co-Principal Investigator (with: Suresh Jagannathan, Tony Hosking, and Ananth Y. Grama on *A Computational Infrastructure for Experimentation on Relaxed Concurrency Abstractions.*
    National Science Foundation. Amount: $99,979. Duration: March 2006 – February 2008.

24. Principal Investigator on *Bertinoro International Summer School: Trends in Concurrency.*
    Microsoft Research. Amount: $5,000. Duration: July 2006.

25. Principal Investigator on *Bertinoro International Summer School: Trends in Concurrency.*
    IBM Italy. Amount: $5,000. Duration: July 2006.

26. Co-Principal Investigator (with PI: Suresh Jagannathan, Purdue) on *Fault Determination and Recovery in Cycle-Sharing Infrastructures.*
    National Science Foundation CSR AES. Amount: $300,000. Duration: September 2005 – August 2008.

27. Principal Investigator on *Aspectual Configuration of Real-time Embedded Middleware.*
    National Science Foundation CSR EHS. Amount: $250,000. Duration: September 2005 – August 2008.

28. Co-Principal Investigator (with PI: Pascal Meunier, CERIAS) on *ReAssure: A logically destructive imaging computer security and forensics experimental facility.*
National Science Foundation CSR MRI. Amount: $800,000. Duration: September 2004 – August 2007.

29. Principal Investigator on *Assured Software Composition For Real-Time Systems.*
National Science Foundation/NASA HDCCSR. Amount: $500,000. Duration: September 2003 – August 2007.

30. Principal Investigator on *Language Abstractions for Parallel Computing.*
DARPA PERCS. Amount: $400,000. Duration: September 2003 – September 2006.

31. Co-Principal Investigator (with PI: Charlie Hu, Purdue) on *Partage: An Open Peer-to-Peer Infrastructure for Cycle-Sharing.*
National Science Foundation ITR. Amount: $498,945. Duration: September 2003 – September 2006.

32. Co-Principal Investigator (with PI Dominic Duggan, Stevens Institute of Technology) on *Distributed Access Control for Accountable Systems.*
National Science Foundation Cybertrust. Amount: $318,375. Duration: September 2002 – August 2006.

33. Principal Investigator on *Foundations and Implementation of Mobile Object Systems.*
National Science Foundation CAREER Award. Amount: $325,936. Duration: September 2001 – August 2006.

34. Principal Investigator (with Co-PIs: Tony Hosking, Purdue; Jens Palsberg, UCLA; Bill Pugh, University of Maryland College Park; Doug Lea, SUNY Oswego) on *DCMF/NES - Dynamic Compositional Middleware Frameworks for Networked Embedded Systems.*
DARPA PCES. Amount: $3,274,680. Duration: July 2001 – May 2005.

35. Principal Investigator on *Software Engineering: Research on Customizable Virtual Machines.*
Microsoft Research. Amount: $100,000. Duration: 2002.

36. Principal Investigator on *Trusted Software Composition.*
Eli Lilly Research Grant. Amount: $50,000. Duration: September 2001 – August 2002.

37. Principal Investigator on *ReAssure–Secure and Resilient Network Computing.*
Eli Lilly Research Grant. Amount: $90,000. Duration: September 1999 – August 2001.

38. Principal Investigator on *Resilient Mobile Agent Architecture.*
Motorola. Amount: $62,543. Duration: September 2000 – August 2005.

39. Principal Investigator on *Type confinement in Java.*
Eli Lilly Research Grant. Amount: $25,000. Duration: Septmeber 1999 – August 2000.

40. Investigator (with PI: Dennis Tsichritzis) on *Agent Systems, Architectures and Platforms.*
Swiss SPP-ICS 5003-45335. Amount: $360'000. Duration: September 1996 – August 1999.

41. Investigator (with PI: Dennis Tsichritzis on *Mobile Object Systems and Cooperative Learning.*
Swiss FNRS 20-40'592.94. Amount: $320'000. Duration: September 1996 – August 1998.

3 PUBLICATIONS

Journal Publications:

[1] T. Kalibera, J. Hagelberg, P. Maj, F. Pizlo, B. Titzer, and J. Vitek. A family of real-time Java benchmarks In *Concurrency and Computation: Practice and Experience*, To appear 2011.

[2] J. Spring, F. Pizlo, J. Privat, R. Guerraoui, J. Vitek. Reflexes: Abstractions for Integrating Highly Responsive Tasks into Java Applications. In *ACM Transactions in Embedded Computing Systems (TECS)*, 2010. 28 pages.

[3] J. Baker, A. Cunei, T. Kalibera, F. Pizlo, J. Vitek. Accurate Garbage Collection in Uncooperative Environments. In *Concurrency and Computation: Practice and Experience*, 21(12), pp. 1572–1606, 2009.

[4] T. Zhao, J. Baker, J. Hunt, J. Noble and J. Vitek. Implicit Ownership Types for Memory Management, In *Science of Computer Programming*, 71, pp. 213–241, 2008.

[5] A. Cunei, J. Vitek. An Efficient ad Flexible Toolkit for Composing Customized Method Dispatchers. In *Software Practice and Experience*, 38(1), pp. 33–73, 2008.

[6] C. Andrea, Y. Coady, C. Gibbs, J. Noble, T. Zhao, J. Vitek. Scoped Types and Aspects for Real-time Java Memory Management. In *Realtime Systems Journal*, pp. 1–44, October, 2007.

[7] A. Armbuster, J. Baker, A. Cunei, C. Flack, D. Holmes, F. Pizlo, E. Pla, M. Prochazka, J. Vitek. A Real-time Java Virtual Machine with Applications in Avionics. In *ACM Transactions in Embedded Computing Systems (TECS)*, 7(1), pp. 1–49 pages, 2007.

[8] C. Grothoff, J. Palsberg, J. Vitek. Encapsulating Objects with Confined Types. In *ACM Transactions on Programming Languages and Systems*, 29(6), 41 pages, 2007.

[9] O. Vitek, B. Craig, C. Bailey-Kellog, J. Vitek. Inferential backbone assignment for sparse data. In *Journal of Biomolecular NMR*, 35(3), pp. 187–208, Springer Verlag, 2006.

[10] B. Cărbunar, A. Grama, J. Vitek, O. Cărbunar. Redundancy and Coverage Detection in Sensor Networks. In *ACM Transaction on Sensor Networks*, pp. 94–128, 2(1), 2006.

[11] T. Zhao, J. Palsberg, J. Vitek. Type-based Confinement. In *The Journal of Functional Programming*, pp 83–128, 16(1), January 2006.

[12] O. Vitek, C. Bailey-Kellogg, B. Craig, P. Kuliniewicz, J. Vitek. Reconsidering Complete Search Algorithms for Protein Backbone NMR Assignment. In *Bioinformatics*, 21, pp. 230–236, September 2005.

[13] S. Jagannathan, J. Vitek, A. Welc, T. Hosking, A Transactional Object Calculus. In *The Science of Computer Programming*, Elsevier, pp. 164–186, 57(2), August 2005.

[14] K. Palacz, J. Baker, C. Flack, C. Grothoff, H. Yamauchi and J. Vitek. The OVM customizable intermediate representation. In *The Science of Computer Programming*, pp. 357–378, 57(3) Elsevier, September 2005.

[15] G. Castagna, J. Vitek and F. Zappa Nardeli. The Seal calculus. In *Information and Computation*, Elsevier, 201(1), pp. 1–54, August 2005.

[16] O. Vitek, J. Vitek, B. Craig and C. Bailey-Kellogg. Model-based assignment and inference of protein backbone nuclear magnetic resonances. In *Statistical Applications in Genetics and Molecular Biology*, Berkeley Electronic Press, Volume 1, Issue 1, 2004.

[17] B. Carbunar, M. T. Valente and J. Vitek. Lime revisited. In *Mathematical Structures in Computer Science*, Cambridge University Press, Volume 14, Issue 3, June 2004, pp. 397–419, 2004.

[18] P. Sewelland J. Vitek. Secure composition of untrusted code: box-π, wrappers and causality types. In *Journal of Computer Security*, IOS Press, 11, pp. 135–188, 2003.

[19] J. Vitek, C. Bryce and M. Oriol. Coordinating agents with secure spaces. In *Science of Computer Programming*, Elsevier, 46, pp. 163–193, 2002.

[20] J. Vitek and B. Bokowski. Confined types for Java. In *Software Practice and Experience*, Wiley, 31, pp. 507–532, 2001.

[21] C. Bryce and J. Vitek. The JavaSeal mobile agent kernel. In *Autonomous Agents and Multi-Agent Systems*, Kluwer, 4, pp. 359–384, 2001.

[22] R. N. Horspool and J. Vitek. Static analysis of PostScript code. In *Journal of Computer Languages*, Pergamon Press, 19, pp. 65-78, 1993.

[23] G. Kappel, J. Vitek, O. Nierstrasz, B. Junod and M. Stadelmann. Scripting applications in the public administration domain. In *SIGOIS Bulletin*, 10, pp. 21-32, 1992.

Articles in Refereed Conference Proceedings

[24] G. Richards, C. Hammer, B. Burg, J. Vitek. The Eval that Men Do: A Large-scale Study of the Use of Eval in JavaScript Applications. To Appear in *European Conference on Object-Oriented Programming (ECOOP)*, Lancaster UK, July 2011.

[25] A. Milanova, J. Vitek. Static Dominance Inference. To appear in *TOOLS Europe*, Zurich, CH, 2011.

[26] F. Pizlo, E. Blanton, P. Maj, J. Vitek, L. Ziareck. SCHISM: Fragmentation-Tolerant Real-Time Garbage Collection. In *Proceedings the ACM Programming Language Design and Implementation Conference (PLDI)*, Toronto, CA, June 2010.

[27] G. Richards, S. Lesbrene, B. Burg, J. Vitek. An Analysis of the Dynamic Behavior of JavaScript Programs. In *Proceedings the ACM Programming Language Design and Implementation Conference (PLDI)*, Toronto, CA, June 2010.

[28] F. Pizlo, L. Ziareck, E. Blanton, P. Maj, J. Vitek. High-level Programming of Embedded Hard Real-Time Devices. In *Proceedings the European Conference on Computer Systems (EUROSYS)*, Paris, France, April 2010.

[29] T. Wrigstad F. Zappa Nardelli, S. Lebresne, J. Ostlund, J. Vitek. Integrating of Typed and Untyped Code in a Scripting Language. In *Proceedings of the ACM Symposium on Principles of Programming Languages (POPL)*, Madrid, Spain January 2010.

[30] M. Schoeberl, F. Brandner, J. Vitek. RTTM: Real-Time Transactional Memory. In *ACM Symposium on Applied Computing, Real-Time Systems Track* (SAC), Sierre, Switzerland, March 2010. (Acceptance rate 26%).

[31] T. Kalibera, F. Pizlo, A. Hosking, J. Vitek. Scheduling Hard Real-time Garbage Collection. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Washington D.C., December 2009. (Acceptance rate 21%).

[32] B. Bloom, J. Field, N. Nystrom, J. Ostlund, G. Richards, R. Strnisa, J. Vitekand T. Wrigstad. Thorn—Robust, Concurrent, Extensible Scripting on the JVM. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, 20 p., Orlando, Florida, October 2009.

[33] T. Kalibera, M. Prochazka, F. Pizlo, J. Vitek, M. Zulianello, M. Decky. Real-time Java in Space: Potential Benefits and Open Challenges. In *Proceedings of DAta Systems In Aerospace (DASIA)*, 10 pp., Istanbul, Turkey, June 2009.

[34] T. Wrigstad, F. Pizlo, F. Meawad, L. Zhao, J. Vitek. Loci: Simple Thread-Locality for Java. In *Proceedings of the European Conference on Object Oriented Programming (ECOOP),* Genova, Italy, June 2009. (Acceptance rate 16%).

[35] A. Cunei, R. Guerraoui, J. Spring, J. Privat, J. Vitek. High-Performance Transactional Event Processing. In *the International Conference on Coordination Models and Languages (COORDINATION)*, pp. 27–46, Madrid, Spain, June 2009. (Acceptance rate 30%).

[36] J. Auerbach, J. H. Spring, D. Bacon, R. Guerraoui, J. Vitek. A Unified Restricted Thread Programming Model for Java. In *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, Tucson, AZ, pp. 1–11, June 2008. (Acceptance rate 25%).

[37] F. Pizlo, J. Vitek. Memory Management for Real-time Java: State of the Art. In *Proceedings of the IEEE International Symposium on Object-oriented Real-Time Distributed Computing (ISORC)*, pp. 248–254, Orlando, Florida, May 2008. (Acceptance rate 30%).

[38] M. Hirtzel, B. Bloom, N. Nystrom, J. Vitek. Matchete: Paths through the Pattern Matching Jungle. In *Symposium on Practical Aspects of Declarative Languages PADL*, San Francisco, CA, pp. 150–166, January 2008. (Acceptance rate 45%).

[39] J. H. Spring, J. Privat, R. Guerraoui, J. Vitek. StreamFlex – High performance stream programming in Java. *Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, pp. 211–228, Montreal, Quebec, Canada, October 2007. (Acceptance rate 19%).

[40] F. Pizlo, A. Hosking, J. Vitek. Hierarchical Real-time Garbage Collection. In *Proceedings of the ACM SIGPLAN/SIGBEDConference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, pp. 123–133, San Diego, California, June 2007. (Acceptance rate 27%).

[41] J.H. Spring, F. Pizlo, R. Guerraoui, J. Vitek. Reflexes: Abstractions for Highly Responsive Systems. In *Proceedings of the ACM/USENIX International Conference on Virtual Execution Environments (VEE)*, pp. 191–201, San Diego, California, June 2007. (Acceptance rate 26%).

[42] J. Baker, A. Cunei, F. Pizlo, J. Vitek. Accurate Garbage Collection in Uncooperative Environments with Lazy Pointer Stacks. In *Proceedings of the International Conference on Compiler Construction (CC)*, pp. 64–79, Braga, Portugal, March 2007. (Acceptance rate 23%).

[43] M. Kalpka, R. Guerraoui, J. Vitek. STMBench7: A Benchmark for Software Transactional Memory. In *Proceedings the European Conference on Computer Systems (EUROSYS)*, Lisbon, Portugal, pp. 315–324, March 2007. (Acceptance rate 20%).

[44] F. Pizlo, J. Vitek. An Empirical Evalutation of Memory Management Alternatives for Real-time Java. In *Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS)*, pp. 35–46, Rio de Janeiro, Brazil, December 2006. (Acceptance rate 23%).

[45] H. Yamauchi, J. Vitek. Combining Offline and Online Optimizations: Register Allocation and Method Inlining. In *Proceedings of The Fourth ASIAN Symposium on Programming Languages and Systems (APLAS)*, pp. 307–322, Sydney, Australia, November 2006.

[46] C. Andrea, Y. Coady, C. Gibbs, J. Noble, J. Vitek, T. Zhao. Scoped Types and Aspects for Real-Time Systems. In *Proceedings of the European Conference on Object Oriented Programming (ECOOP)*, pp. 124–147, Nantes, France, July 2006. (Acceptance rate 13%).

[47] A. Cunei, J. Vitek. A New Approach to Real-time Checkpointing. In *Proceedings of the Second ACM/USENIX International Conference on Virtual Execution Environments (VEE)*, pp. 68–77, Ottawa, Canada, June 2006. (Acceptance rate 35%).

[48] J. Baker, A. Cunei, C. Flack, F. Pizlo, M. Prochazka, J. Vitek, A. Armbuster, E. Pla, D. Holmes. Real-time Java in Avionics Applications. In *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 384–396, San Jose, CA, April 2006. (Acceptance rate 29%).

[49] J. Manson, J. Baker, A. Cunei, S. Jagannathan, M. Prochazka, B. Xin, J. Vitek. Preemptible Atomic Regions for Real-time Java. In *Proceedings of the 26th IEEE Real-Time Systems Symposium (RTSS)*, pp. 62–71, Miami, FL, December 2005. (Acceptance rate 21%).

[50] A. Cunei, J Vitek, PolyD: A Flexible Dispatching Framework, In *Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'05)*, pp. 487–504, San Diego, CA, October 2005. (Acceptance rate 18%).

[51] O. Vitek, C. Bailey-Kellogg, B. Craig, P. Kuliniewicz, J. Vitek, Reconsidering Complete Search Algorithms for Protein Backbone NMR Assignment. In *European Conference on Computational Biology (ECCB'05)*, pp. 236–245, Vienna, Austria, September 2005. (Acceptance rate 14%).

[52] R. Gopalakrishna, E. Spafford, J. Vitek. Efficient Intrusion Detection using Automaton Inlining. In *Proceedings of the IEEE Symposium on Security and Privacy*, (S&P'05), pp. 18–31, Oakland, CA, May 2005. (Acceptance rate 8%).

[53] T. Zhao, J. Noble, J. Vitek, Scoped Types for Real-time Java, In *Proceedings of the 25th IEEE International Real-Time Systems Symposium* (RTSS04), pp. 241–251, Lisbon, Portugal, December 5-8, 2004. (Acceptance rate 22%).

[54] B. Carbunar, I. Ioannidis, A. Grama, J. Vitek. A Secure Crediting Protocol for Hybrid Cellular and Ad-Hoc Networks. In *Proceedings of the 1st International Conference on E-Business and Telecommunication Networks* (ICETE). pp. 142–149, Setubal, Portugal, August 2004.

[55] B. Carbunar, A. Grama, J. Vitek. Coverage Preserving Redundancy Elimination in Sensor Networks. In *First IEEE International Conference on Sensor and Ad-Hoc Communications and Networks* (SECON). 2004. (Acceptance rate 25%).

[56] B. Carbunar, A. Grama and J. Vitek. Distributed and Dynamic Voronoi Overlays for Coverage Detection and Distributed Hash Tables in Ad-Hoc Networks. In *Proceedings of the Tenth International Conference on Parallel and Distributed Systems (ICPADS 2004)*, pp. 549–559, Newport Beach, July 2004. (Acceptance rate 20%).

[57] F. Pizlo, J. Fox, D. Holmes and J. Vitek. Real-Time Java scoped memory: design patterns and semantics. In *Proceedings of the IEEE International Symposium on Object-oriented Real-Time Distributed Computing (ISORC'04)*, pp. 101–112, Vienna, Austria, May 2004. (Acceptance rate 20%).

[58] J. Vitek, S. Jagannathan, A. Welc and A.L. Hosking. A semantic framework for designer transactions. In *Proceedings of the European Symposium on Programming (ESOP'04)*, pp. 249–263, Barcelona, Spain, April 2004. (Acceptance rate 21%).

[59] S. Jagannathanand J. Vitek. Optimistic concurrency semantics for transactions in coordination languages. In *Proceedings of the International Conference on Coordination Models and Languages (COORDINATION'04)*, pp. 183–198, Pisa, Italy, March 2004. (Acceptance rate 34%).

[60] T. Chothia, D. Dugganand J. Vitek, Principals, Policies and Keys in a Secure Distributed Programming Language. In *IEEE Computer Security Foundations (CSF'03)*, pp. 170–180 Turku, Finland, July, 2003.

[61] T. Zhao, J. Palsberg and J. Vitek. Lightweight confinement for featherweight Java. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems and Languages (OOPSLA'03)*, pp. 135–148, San Diego, California, October 2003. (Acceptance rate 18%).

[62] T. Chothia, D. Duggan and J. Vitek. Type-based distributed access control. In *Proceedings of the IEEE Computer Security Foundations Workshop (CSFW'03)*, pp. 170 –187, Pacific Grove, California, July 2003. (Acceptance rate 40%).

[63] K. Palacz and J. Vitek. Subtype tests in real time. In *Proceedings of the European Conference on Object Oriented Programming (ECOOP'03)*, pp. 378–404, Darmstadt, Germany, July 2003. (Acceptance rate 20%).

[64] K. Palacz, J. Baker, C. Flack, C. Grothoff, H. Yamauchi and J. Vitek. (Acceptance rate 19%). Engineering a customizable intermediate representation. In *Proceedings of the ACM SIGPLAN Workshop on Interpreters, Virtual Machines and Emulators, (IVME'03)*, pp. 1–12, San Diego, California, June 2003. (Acceptance rate 19%).

[65] K. Palacz, G. Czaikowski, L. Daynes and J. Vitek. Incommunicado: a communication substrate for Isolates. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems and Languages (OOPSLA'02)*, pp. 262–274, Seattle, Washington November 2002. (Acceptance rate 20%).

[66] B. Carbunar, M. T. Valente and J. Vitek. Lime revisited. In *Proceedings of the International Conference on Mobile Agents (MA'01)*, pp. 54–69, Atlanta, Georgia, December 2001. (Acceptance rate 24%).

[67] C. Grothoff, J. Palsberg and J. Vitek. Encapsulating objects with confined types. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems and Languages (OOPSLA'01)*, pp. 241–255, Tempa, Florida, October 2001. (Acceptance rate 19%).

[68] P. Sewelland J. Vitek. Secure composition of untrusted code: wrappers and causality types. In *Proceedings of the Computer Security Foundations Workshop (CSFW'00)*, pp. 269–284, Cambridge, England, July 2000. (Acceptance rate 48%).

[69] P. Sewell and J. Vitek. Secure composition of insecure components. In *Proceedings of the Computer Security Foundations Workshop* (CSFW'99), pp. 136–150, Mordano, Italy, June 1999. (Acceptance rate 38%).

[70] C. Bryce, M. Oriol and J. Vitek. Secure object spaces: a coordination model for agents. In *Proceedings of the International Conference on Coordination Models and Languages (COORDINATION'99)*, pp. 4–20, Amsterdam, Netherlands, April 1999. (Acceptance rate 38%).

[71] B. Bokowski and J. Vitek. Confined types. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'99)*, pp. 82–97, Denver, Colorado, October 1999. (Acceptance rate 20%).

[72] J. Vitek and C. Bryce. Security for mobile code: the JavaSeal experiment. In *Proceedings of the Agent Systems and Applications Mobile Agents (ASA/MA'99)*, pp. 103–118, Palm Springs, California, October 1999. (Acceptance rate 20%).

[73] Q. Bradley, R. N. Horspool and J. Vitek. JAZZ: An efficient compressed format for Java archive files. In *Proceedings of the IBM CASCON Conference (CASCON'98)*, pp. 294–302, Toronto, Canada, December 1998.

[74] J. Noble, J. Vitekand J. Potter. Flexible alias protection. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP'98)*, pp. 158–185, Brussels, Belgium, July 1998. (Acceptance rate 19%).

[75] J. Vitek, R.N. Horspool and A. Krall. Efficient type inclusion tests. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'97)*, pp. 142–157, San Jose, California, October 1997. (Acceptance rate 17%).

[76] A. Krall, J. Vitek and R.N. Horspool. Near optimal hierarchical encoding of types. In *In Proceedings of the European Conference on Object-Oriented Programming (ECOOP'97)*, pp. 128–146, Jyvaskyla, Finland, June 1997. (Acceptance rate 19%).

[77] A. Krall and J. Vitek. On extending Java. In *Proceedings of the Joint Modular Languages Conference (JMLC'97)*, pp. 321–325, Linz, Austria, March 1997.

[78] J. Vitek and R. N. Horspool. Compact dispatch tables for dynamically typed object oriented languages. In *Proceedings of the International Conference on Compiler Construction (CC'96)*, pp. 309–326, Linkoping, Sweden, April 1996. (Acceptance rate 40%).

[79] K. Driesen, U. Hölzle and J. Vitek. Message dispatch on pipelined processors. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP'95)*, pp. 253–283, Åarhus, Denmark, August 1995. (Acceptance rate 19%).

[80] J. Vitek and R. N. Horspool. Taming message passing: efficient method look-up for dynamically typed languages. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP'94)*, pp. 432–449, Bologna, Italy, July 1994. (Acceptance rate 15%).

[81] J. Vitek, R.N. Horspool and J. Uhl. Compile-time analysis of object-oriented programs. In *Proceedings of the Conference on Compiler Construction (CC'92)*, pp. 236–250, Paderborn, Germany, October 1992. (Acceptance rate 25%).

[82] R. N. Horspool and J. Vitek. Static analysis of PostScript code. In *Proceedings of the International Conference on Computer Languages (ICCL'92)*, pp. 14–23, Oakland, California, April 1992. (Acceptance rate 22%).

Refereed Workshop Publications

[83] N. Kidd, S. Jagannathan, J. Vitek. One Stack to Run Them All: Reducing Concurrent Analysis to Sequential Analysis Under Priority Scheduling. In *International SPIN Workshop on Model Checking of Software* (SPIN'10), Enschede, NL, September 2010.

[84] D. Tang, A. Plsek, J. Vitek. Static Checking of Safety Critical Java Annotations. In *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Prague, CZ, September 2009.

[85] A. Plsek, L. Zhao, V. Sahin, D. Tang, T. Kalibera, J. Vitek. Developing Safety Critical Java applications with oSCJ/L0. In *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Prague, CZ, September 2009.

[86] T. Kalibera, P. Parizek, G. Haddad, G. Leavens, J. Vitek. Challenge Benchmarks for Verification of Real-time Programs. In *ACM Workshop on Programming Languages meets Program Verification (PLPV)*, 6 pages, Madrid, Spain, January 2010.

[87] T. Kalibera, J. Hagelberg, F. Pizlo, A. Plsek, B. Titzer, J. Vitek. CDx: A Family of Real-time Java Benchmarks. In *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Madrid, SP, September 2009.

[88] L. Zhao, D. Tang, J. Vitek. A Technology Compatibility Kit for Safety Critical Java In *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Madrid, SP, September 2009.

[89] F. Pizlo, L. Ziarek, J. Vitek. Towards Java on Bare Metal with the Fiji VM. In *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Madrid, SP, September 2009.

[90] T. Wrigstad, P. Eugster, J. Field, N. Nystrom, J. Vitek. Software Hardening: A Research Agenda. In *International Workshop on Script to Program Evolution (STOP)*, Genoa, IT, July 2009.

[91] S. Lebresne, G. Richards, J. Östlund, T. Wrigstad, J. Vitek. Understanding the Dynamics of JavaScript. In *International Workshop on Script to Program Evolution (STOP)*, Genoa, IT, July 2009.

[92] J. Hunt, D. Locke, K. Nilsen, M. Schoeberl, J. Vitek. Java for Safety-Critical Applications. In *Certification of Safety-Critical Software Controlled Systems (SafeCert'09)*, York, UK, March 2009.

[93] M. Schoeberl, J. Vitek. Garbage Collection for Safety Critical Java. In *5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Vienna, Austria, September 2007.

[94] I. Dragos, A. Cunei, J. Vitek. Continuation in the Java Virtual Machine. In *Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2007)*, Berlin, Germany, July 2007.

[95] Y. Coady, C. Gibbs, M. Haupt, J. Vitek, H. Yamauchi. Towards a domain specific language for virtual machines. In *Domain-Specific Aspect Languages Workshop (DSAL)*, Portland Oregon, October 2006.

[96] J. Manson, S. Jagannathan, and J. Vitek. Dynamic Aspects for Runtime Fault Determination and Recovery. In *Dynamic Aspects Workshop (DAW05)*, Chicago, Ill., March 2005.

[97] F. Pizlo, M. Prochazka, S. Jaggannathan and J. Vitek. Transactional lock-free data structure for Real Time Java. In *Workshop on Concurrency and Synchronization in Java Programs*, St John's, Newfoundland, Canada, July 2004. (Acceptance rate 55%).

[98] B. Carbunar, M.T. Valente and J. Vitek. CoreLime: a coordination model for mobile agents. In *Proceedings of the International Workshop on Concurrency and Coordination*, Lipary, Italy, July 2001.

[99] J. Vitek and G. Castagna. Mobile computations and hostile hosts. In *Proceedings of the Journées Francophones des Langages Applicatifs (JFLA'99)*, pp. 113–132, Avoriaz, France, February 1999.

[100] J. Vitek. New Paradigms in distributed computing. In *Proceedings of the European Research Seminar in Advanced Distributed Systems (ERSADS'97)*, pp. 117–122, Zinal, Switzerland, March 1997.

[101] J. Vitek. Secure object spaces. In *ECOOP Workshop on Mobile Object Systems*, pp. 41-48, Linz, Austria, July 1996.

Book Chapters

[102] N. Suri, J. Vitek. "Mobile Agents" in *Encyclopedia of Complexity and Systems Science*, pp. 5604–5618, Springer Verlag, 2009.

[103] J. Vitek. Mobile Agents. In *Software Agents for the Warfighter*, J.M. Bradshaw, G. Boy, E. Durfee, M. Gruninger, H. Hexmoor, N. Suri, M. Tambe, M. Uschold and J. Vitek (Eds.), ITAC Report, 2000.

[104] J. Vitek and G. Castagna. Seal: a framework for secure mobile computations. In *Internet Programming Languages*, pp. 47–78, Springer-Verlag, 1998.

[105] J. Vitek, M. Serrano and D. Thanos. Security and communication in mobile object systems. In *Mobile Object Systems: Towards the Programmable Internet*, pp. 177–200, Springer-Verlag, 1997.

Books

[106] J. Vitek(Ed). Proceedings of the *48th Conference on Objects, Models, Components, Pattern* (TOOLS Europe), Springer Verlag, June 2010.

[107] J. Vitek(Ed). Proceedings of the *8th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, ACM Press, August 2010.

[108] J. Vitek(Ed). Proceedings of *22nd European Conference on Object-Oriented Programming*, Springer Verlag, July 2008.

[109] A. Murphy, J. Vitek (Eds). Proceedings of the *9th International Conference on Coordination Models and Languages (COORDINATION)*, Springer Verlag, June 2007.

[110] M. Hind, J. Vitek (Eds). Proceedings of the *First ACM/USENIX International Conference on Virtual Execution Environments VEE'05*, ACM Press, 2005.

[111] J. Vitek and C. Jensen (Eds.). *Secure Internet Programming: Security Issues for Mobile and Distributed Object Systems*. Springer-Verlag, 1999.

[112] J. Vitek and C. Tschudin (Eds.). *Mobile Object Systems: Towards the Programmable Internet*. Springer-Verlag, 1997.

Journal Special Issues

[113] Amy Murphy, Einar Broch Johnsen, Jan Vitek (Eds). Special issue DiScoTec 2007, In *Theoretical Computer Science*, vol 410(2-3): 113, 2009.

[114] F. Logozzo, J. Vitek (Eds). Proceedings of the 7th Workshop on Formal Techniques for Java-like Programs - FTfJP'2005 (Special issue). Journal of Object Technology, 2006.

[115] M. Noir, N. Shavit, J. Vitek (Eds). *Special issue on Concurrency and Synchronization in Java*, The Science of Computer Programming, 58(3), December 2005.

[116] P. Ciancarini, R. Tolskdorf and J. Vitek (Eds.). Special issue on Distributed World Wide Web Processing: Applications and Techniques. *WWW Journal*, 1998.

[117] P. Ciancarini, R. Tolskdorf and J. Vitek (Eds.). Workshop on Collaborative Agents in Distributed Web Applications. In *Proceedings of the 6th IEEE Workshops on Enabling Technologies for Collaborative Enterprises*, (WETICE). MIT, Boston, 1998.

[118] J. Vitek and C. Tschudin (Eds.). *Proceedings of the 1996 Mobile Object Systems Workshop*. Dpunkt Verlag, 1998.

Under review

[119] J. Spring, J. Auerbach, D. Bacon, R. Guerraoui, T. Zhao, J. Vitek, The Flexotask programming model.

Other Publications

[120] J. Vitek. Introduction to: The Myths of Object-Orientation. *Proceedings of the European Conference on Object Oriented Programming (ECOOP)*, July 2009.

[121] J. Vitek, C. Bryce and W. Binder. Designing JavaSeal, or how to make Java safe for agents. In *Electronic Commerce Objects*, D. Tsichritzis (Ed.), pp. 105-126, Centre Universitaire d'Informatique, University of Geneva, July 1998.

[122] J. Vitek. Compact dispatch tables for dynamically typed programming languages. In *Object Applications*, D. Tsichritzis (Ed.), pp. 81-138, Centre Universitaire d'Informatique, University of Geneva, August 1996.

[123] D. Konstantas, J.H. Morin and J. Vitek. MEDIA: A platform for the commercialization of electronic documents. In *Object Applications*, D. Tsichritzis (Ed.), pp. 7-18, Centre Universitaire d'Informatique, University of Geneva, August 1996.

[124] O. Nierstrasz, L. Dami, V. de Mey, M. Stadelmann, D. Tsichritzis and J. Vitek. Visual scripting – towards interactive construction of object-oriented applications. In *Object Management*, D. Tsichritzis (Ed.), pp. 315-331, Centre Universitaire d'Informatique, University of Geneva, July 1990.

[125] M. Stadelmann, G. Kappel and J. Vitek. VST: a scripting tool based on the UNIX shell. In *Object Management*, D. Tsichritzis (Ed.), pp. 333-344, Centre Universitaire d'Informatique, University of Geneva, July 1990.

[126] J. Vitek, B. Junod, O. Nierstrasz, S. Renfer and C. Werner. Events and sensors: enhancing the reusability of objects. In *Object Management*, D. Tsichritzis (Ed.), pp. 345-356, Centre Universitaire d'Informatique, University of Geneva, July 1990.

[127] G. Kappel, J. Vitek, O. Nierstrasz, S. Gibbs, B. Junod, M. Stadelmann and D. Tsichritzis. An object-based visual scripting environment. In *Object Oriented Development*, D. Tsichritzis (Ed.), pp. 123-142, Centre Universitaire d'Informatique, University of Geneva, July 1989.

4 LECTURES

Invited Lectures

1. Of Scripts and Programs: Tall tales, Urban Legends, and Future Prospects. Keynote talk at the *Analysis and Programming Languages for Web Applications and Cloud Applications*, Toronto, CA, June 2010.

2. Programming Models for Concurrency and Real-time. Keynote talk at the *47th International Conference on Objects, Models, Components, Patterns (TOOLS)*, Zurich, July 2009.

3. Memory Management for Hard Real-time Systems. Invited talk at the *Workshop on Virtual Machines and Intermediate Languages for emerging modularization mechanisms* (VMIL), Nashville, Tennessee on October 19, 2008.

4. Programming models for Concurrency and Real-time. Invited talk at *XII Brazilian Symposium on Programming Languages*, Fortaleza, Brazil, on August 27-29, 2008.

5. Programming models for Concurrency and Real-time. Invited talk at *Programming Language Approaches to Concurrency and Communication-cEntric Software*, June 7, 2008, Oslo, Norway.

6. Semantics-based Intrusion Detection, Invited Talk at the *Foundations of Computer Security*, Chicago, June 29, 2005.

7. Java for Hard Real-Time, Invited Talk at the *Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2006)*, Nantes, France, July 2006.

8. Advances in Intrusion Detection, Keynote talk at the *Program Analysis for Security and Safety Workshop (PASSWORD)*, Nantes, France, July 2006.

National and International Meetings.

1. CDx: A Family of Real-time Java Benchmarks. *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Madrid, SP, September 2009.

2. A Technology Compatibility Kit for Safety Critical Java. *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Madrid, SP, September 2009.

3. Software Hardening: A Research Agenda. *International Workshop on Script to Program Evolution (STOP)*, Genoa, IT, July 2009.

4. Programming Real-time Embedded Systems in Java. Summer school part of the *Wroclaw Information Technology Initiative*, Wroclaw, PL, 18-20 May 2009.

5. Java for Safety-Critical Applications, *Certification of Safety-Critical Software Controlled Systems (SafeCert'09)*, York, UK, March 2009.

6. Large-Scale Embedded Programming, *Software Quality Symposium*, Swiss Federal Institute of Technology, Zurich, CH, 2007.

7. Programming Highly Responsive Systems, *International Federation for Information Processing (IFIP) Working Group 2.4 "Software Implementation Technology"*, Lake Arrowhead, CA, 2007.

8. Transactions and Composability: Transactions Considered Harmful? *IBM Workshop on Transactional Memory and Programming Technologies*, Armonk, NY, March 5-6, 2007.

9. Data-centric Synchronization, *IBM Workshop on Transactional Memory and Programming Technologies*, Armonk, NY, March 5-6, 2007.

10. How not to get a job in research, *Summer School on Trends in Concurrency,* Bertinoro, Italy, July 2006.

11. Scoped Types and Aspects for Real-Time Systems, *European Conference on Object Oriented Programming (ECOOP)*, Nantes, France, July 2006.

12. Real-time Java in Avionics Applications. *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2006.

13. Preemptible Atomics, *International Federation for Information Processing (IFIP) Working Group 2.4 "Software Implementation Technology"*, Jackson's Mill, West Virginia, October, 2005.

14. Memory Safe RTSJ Programming, *Safety & Mission Critical Workshop (JAWS05)*, Palo Alto, CA, September 2005.

15. Preemptible Atomic Regions, SUN Microsystems, CA, August 2005.

16. Adopting Ownership Types, *Dagstuhl Tool for Types Workshop*, Dagstuhl, DE, June, 2005.

17. Stealth Types, *Foundations of Object-Oriented Languages, (FOOL'05)* panel on Extreme Typing, Long Beach, CA, January 11, 2005.

18. The Real-time Specification for Java: issues and opportunities, *International Federation for Information Processing (IFIP) Working Group 2.4 "Software Implementation Technology"*, Baden, Austria, January 5, 2005.

19. Scoped Types for Real-time Java, *25th IEEE International Real-Time Systems Symposium* (RTSS04), Lisbon, December 5-8, 2004.

20. A semantic framework for designer transactions, *European Symposium on Programming (ESOP'04)*, Barcelona, April 2004.

21. Transactional Facilities for Java. *International Conference on Object Oriented Programming Systems, Languages and Applications*, Vancouver BC, 2004.

22. Security and Coordination. *International School on Foundations of Security Analysis and Design* (FOSAD), Italy, September 2004.

23. Real-time Java with the Ovm virtual machine. *Real-time Java Symposium*, Defense Advance Research Programs Agency, Arlington, VA, July 2004.

24. Engineering Intermediate Representations, *IFIP Working Group 2.4*, Santa Cruz, CA, July 2003.

25. Lightweight confinement for featherweight Java, *Conference on Object-Oriented Programming Systems and Languages (OOPSLA'03)*, San Diego, October 2003.

26. Subtype tests in real time. In *European Conference on Object Oriented Programming (ECOOP'03)*, Darmstadt, July 2003.

27. Engineering a customizable intermediate representation, *Workshop on Interpreters, Virtual Machines and Emulators, (IVME'03)*, San Diego, June 2003.

28. Encapsulating objects with confined types, *Conference on Object-Oriented Programming Systems and Languages (OOPSLA'01)*, Tempa, October 2001.

29. Confined Types, *IFIP Working Group 2.4*, Italy, July 2001.

30. Confined types, *Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'99)*, Denver, October 1999.

31. Efficient type inclusion tests, *Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'97)*, San Jose, October 1997.

32. Near optimal hierarchical encoding of types, *European Conference on Object-Oriented Programming (ECOOP'97)*, Jyvaskyla, June 1997.

33. Compact dispatch tables for dynamically typed object oriented languages, *International Conference on Compiler Construction (CC'96)*, Linkoping, Sweden, April 1996.

34. Taming message passing: efficient method look-up for dynamically typed languages, *European Conference on Object-Oriented Programming (ECOOP'94)*, Bologna, Italy, July 1994.

35. Compile-time analysis of object-oriented programs, *Conference on Compiler Construction (CC'92)*, Paderborn, Germany, October 1992.

Talks at Universities and Other Institutions.

1. Microsoft Research, Redmond (2011), ETHZ, Zurich (2011).

2. INRIA-Rennes (2010).

3. Imperial College (2009), Microsoft Research (2009), Brown University (2009), EPFL (2009), University of Central Florida (2009).

4. University of Lugano, CH (2008), INRIA Rocquencourt, FR (2008), INRIA Rennes, FR (2008), Ecole Polytechnique Fédérale, Lausanne, CH (2008), Imperial College, UK (2008), University of California, Los Angeles, CA (2008),

5. Edinburgh University, UK (2007), IBM T.J. Watson, NY (2007), Charles University, Prague, CZ (2007), Microsoft Research, WA (2007),

6. IBM T.J. Watson, NY (2006), Swiss Federal Institute of Technology, Zurich, CH(2006), University of Bern (2006), Ecole Polytechnique Fédérale, Lausanne (2006), Portland State University (2006), Microsoft Research (2006), University of Utah (2006),

7. University of Washington (2005), Carnegie Mellon University (2005), University of Victoria (2005), University of Alberta (2005),

8. University of Nice (2003),

9. Tokyo University (2001),

10. University of Waterloo (1999), University of Syracuse (1999), University of Pennsylvania (1999), University of Toronto (1999), University of Victoria (1999), University of Rennes (1999).

5 SOFTWARE ARTIFACTS

Vitek has acted as team leader and main developer on several software systems. This work has been carried in close collaboration with industrial partners including Boeing, Jet Propulsion Laboratories, Motorola, Lockheed Martin and IBM. All software listed below were released in open source form and are publicly available.

1. **Thorn:** Thorn is new concurrent and distributed programming language being developed at Purdue and IBM Research. Thorn support rapid software development in the style of dynamic scripting languages as well as gradual hardening of scripts into robust programs with a novel type system based on Like Types. Publications: [90, 91, 38, 32]
   Development: 2008 – present. Open source.
   Web site: `http://www.thorn-lang.org`
   Support: ONR, IBM.

2. **PJAz:** The Purdue JavaScript Analyzer package is the first open source trace-based analysis engine for JavaScript. Our tool allows developers, compiler writers, security experts to understand the dynamic behavior of JavaScript programs. Results obtained with PJAz have shown that common benchmark used in the industry to measure the performance of JavaScript implementations are not representative of real-world programs and has been helpful in invalidating many widely held misconceptions about how the language is being used.
   Publications: [27, 91]
   Development: 2009 – present. Open source.
   Support: NSF.

3. **CDx:** The CDx (Collision Detector) benchmark suite is an open source application benchmark suite that targets dierent hard and soft real-time virtual machines. CDx is, at its core, a real-time benchmark with a single periodic task, which implements aircraft collision detection based on simulated radar frames. The benchmark can be congured to use dierent sets of real-time features and comes with a number of workloads running in C, Java or RTSJ. It is currently the only publicly available real-time Java workload.
   Publications: [31, 33, 87, 86]
   Development: 2004 – present. Open source.
   Web site: `http://www.cs.purdue.edu/ tkaliber/rcd`
   Support: NSF.

4. **Flexotasks:** Achieving sub-millisecond response times in a managed language environment such as Java requires overcoming significant challenges. Flexotasks are a programming model and runtime system infrastructure developped jointly with EPFL and IBM Research lets developers mix highly responsive tasks and timing-oblivious Java applications. We have shown flexotasks to be suitable for the hardest of real-time environments on single as well as multi-core architectures.
   Publications: [41, 39, 36, 35, 2]
   Development: 2007 – 2009. Open source.
   Web site: `http://flexotask.sourceforge.net`
   Support: NSF, IBM.

5. **StmBench7:** Software transactional memory (STM) is a promising technique for controlling concurrency in modern multi-processor architectures. STM aims to be more scalable than coarse-grained locking and easier to use than fine-grained locks. However, STM implementations have yet to demonstrate that their runtime overheads are

acceptable. To date, empiric evaluations of these implementations have suered from the lack of realistic benchmarks. STMBench7 is a benchmark for evaluating STM implementations.
Publications: [43]
Development: 2007 – present. Open source.
Web site: `http://lpd.epfl.ch/kapalka/oo7.php`
Support: NSF

6. **Ovm:** An open source Java virtual machine framework with support for the Real-time Specification for Java. The Ovm system was developed by a team of nine students, two post-doctoral researchers and one independent software consultants led by Vitek. The entire system is over 100,000 lines of code. Ovm is currently the highest performing real-time Java virtual machine.
   Users: *Boeing, Lockheed Martin, University of California Irvine, Purdue and Kansas State University.*
   Publications: [7, 14, 40, 42, 44, 45, 47, 48, 49, 57, 64, 6, 5, 41, 46, 50]
   Web site: `http://ovmj.net`
   Development: 2000 – present. Open source.
   Support: DARPA, NSF, Boeing, IBM.

7. **MBA:** A tool for *M*odel-*B*ased protein backbone nuclear magnetic resonance *A*ssignments. MBA was developed in collaboration with the Purdue Statistics department. It is currently being evaluated by researchers in several research laboratories.
   Publications: [9, 12, 51, 16]
   Web site: `www.stat.purdue.edu/ ovitek/mba/mba.html`
   Development: 2003 – 2005. Open source.
   Support: NSF.

8. **Kacheck:** A tool for analyzing Java programs for detecting confinement violations developed by Vitek and one graduate student. Kacheck has been used to analyze over 100MB of Java code. Publications: [8,20, 21, 67, 71]
   Development: 2000 – 2002. Open source.
   Support: DARPA, Lockeed Martin.

9. **JavaSeal:** A mobile agent middleware system based on the Java programming language. This system was developed at the University of Geneva by Vitek in collaboration with four graduate students. JavaSeal has been used in the implementation of HyperNews, a commercial electronic publishing application. Publications: [15, 19, 72, 104, 99, 121, 100, 123]
   Development: 1996 – 1999.
   Support: Swiss National Fund, European Union ESPRIT.

10. **Jazz:** A tool for compression of Java class files. This was developed by Vitek and a graduate student at the University of Victoria.
    Publication: [73].
    Development: 1998.

6 Advised Students

Graduated Students

1. Johan Östlund. MSc, 2010.
2. Jesper H. Spring, (Co-advised with Rachid Guerraoui, EPFL).
   PhD, Thesis Title: "Reflexes: Programming Abstractions for Highly Responsive Computing in Java", September 2008.
3. Rajeev Gopalakrishna (in co-direction with Prof. Spafford).
   PhD, Thesis Title: "Metric-driven feedback mechanism for secure software development", May 2006. (First Position: Intel Research Labs).
4. Bogdan Carbunar.
   Phd, Thesis Title: "Coverage Problems in Wireless Sensor Networks", May 2005 (First position: Motorola Research Labs.)
5. Krzyzstof Palacz,
   PhD, Thesis Title: "Crusoe–Towards a Multicomputer Execution Environment for Java", December 2004. (First position: Sun Research Labs).
6. Jason Baker. MSc, 2007, (First position: Google).
7. Hiroshi Yamauchi, MSc, 2007, (First position: Google).
8. Christian Grothoff, MSc, 2005 (First Position: University of Denver).
9. Andrey Madan, MSc, 2004, (First position: Medtronics).
10. Gergana Markova, MSc Thesis Title: "Analyzing the Visitor Design Pattern", 2003 (First position: IBM).
11. Jason M. Fox, MSc, 2003 (First position: Jet Propulsion Laboratories).
12. James Liang, MSc, 2002 (First position: Sandia Labs).

Doctoral Students

1. Jacques Thomas. Past Prelim.
   Expected graduation January 2011.
2. Filip Pizlo, Past Qual II. Thesis: "Real-time Memory Managment Techniques", Expected graduation date: Spring 2011.
3. Daniel Tang. Pre-Qual. (Start Fall 2008)
4. Gregor Richards. Pre-Qual. (Start Fall 2008)
5. Lei Zhao. Pre-Qual. (Start Fall 2008)
6. Fadi Meawad, Pre-Qual. (Start Fall 2008)

Current Undergraduate students

1. Ryan Macnak.

Past Undergraduate students

1. Brian Burg, BSc 2010. (Next: University of Washington graduate program)
2. Brett Mravec, Jason Ward, Chris Abernathy, BSc 2010.
3. Rob Gevers, BSc 2009. (Next position: Purdue graduate program)
4. Daniel Tang. Project: Static analysis of Real-time Java. BSc 2008. (Next position: Purdue Graduate Program)
5. William Harris. Project: Pattern Matching in Java.
6. Andrew McClure, Research project: Cluster Java Virtual Machines. Main achievement: Implementation of Cluster support in a Java VM while working as an intern at SUN Microsystem Research Laboratories.
7. Zacchary Wiggins, Research project: Algorithms for SpinSystem Computation in NMR Data.
8. Paul Kuliniewicz, Research project: Protein Backbone Assignment. 2004.
9. Wenchang Liu, Research project: JavaBench–benchmarking real-time Java applications. Main achievement: JavaBench has been released as open source software. BSc 2004 (Next position: Purdue graduate program).
10. Filip Pizlo, Research project: Real-time Java Programming Model. Main achievements: Two published papers in major real-time forums. BSc 2004 (Next position: Purdue CS graduate student).
11. Chris Willmore, Research project: Java just-in-time compilation for the PPC.
12. Andrey Madan, Research project: Generics Graph Classes in Java. BSc 2002 (Next position: Medtronics, after a MSc at Purdue).
13. Ben Titzer, Research project: TGen – Generic Macro Processor for Java. BSc 2003 (Next position: UCLA after spending a year in the CS graduate program at Purdue).
14. Adam Lugowski, Research project: Applications of Real-time Java to Computer Games.
15. Josh Moore, Research project: A Software Radio in Real-time Java.
16. Gergana Markova, BSc 2001 (Next position: at IBM after graduating from CS at Purdue).
17. Theodore Witkamp, BSc 2003 (Next position: Idealab, Pasadena(CA).
18. Javed Siddique, Alen Montz, BSc 2004 (Purdue).

Post-doctoral Researchers

1. Nicholas Kidd, 2009 – current.
2. Christian Hammer, 2009 – current.
3. Ales Plsek, 2009 – current.
4. Sylvain Lebresne, 2008 – 2009. (First Position: yakaz.com)

5. Tomas Kalibera, 2007 – 2009. (First Position: Charles University)
6. Tobias Wrigstad, 2007 – 2009. (First Position: University of Stockholm)
7. Antonio Cunei, 2003 – 2008. (First Position: EPFL post-doc).
8. Jean Privat, 2006 – 2007, (First position: Université du Québec).
9. Marek Prochazka, 2003 – 2005, (First Position: SciSys).
10. Jeremy Manson, 2003 – 2005, (First Position: Google).
11. Michael Richmond, 2002 – 2003, (First Position: IBM Research).

Internships

Vitek has been instrumental in placing his students in highly-competitive research internships. The success rate for IBM internship is between 5-10% (last year there were 400 internship candidates and fewer than 20 positions).

1. Fadi Meawad (2010), Microsoft Research.
2. Lei Zhao (2010), Oracle.
3. Gregor Richards (2010), Microsoft Research.
4. Daniel Tang (2010), Google.
5. Fadi Meawad (2009), Google.
6. Gregor Richards (2009), IBM Research.
7. Armand Navabi (2009), Microsoft.
8. Johan Östlund (2009), Adobe.
9. Jesper Spring, IBM Research.
10. Filip Pizlo, Microsoft Resarch.
11. Jacques Thomas, Google.
12. Jacques Thomas, Microsoft.
13. Krzystof Palacz, SUN Labs.
14. Adam Welc, SUN Labs.
15. Hiroshi Yamauchi, SUN Labs.
16. Gergana Markova, IBM Research.
17. Filip Pizlo, IBM Research.
18. Christian Grothof, IBM Research.
19. Andrew McClure, SUN Labs.
20. Ben Titzer, SUN Labs.
21. Andrei Madan, Medtronics.

7 SERVICE

Dr. Vitek is has a rich community service record. He has served on over 50 conference program committees as a PC member or a chair, and about as many workshops. He initiated several successful Workshop series, starting with Mobile Object Systems (MOS) (1995–2005), the International Workshop on Aliasing, Confinement and Ownership (IWACO) (1999–present), the ACM SIGPLAN Workshop on Lanugages, Compilers, and Hardware Support for Transactional Computing (TRANSACT) (2006–present), and was the first program chair of the ACM/USENIX Conference on Virtual Excetion Environments (2005–present). Together with his colleagues Jagannathan and Grama, he founded and ran the first two instances of the Summer School on Trends in Concurrency (TIC) which were held in Bertinoro and Prague and attracted over 50 PhD students each time.

International Meetings and Schools

1. Organizer of the Third International Summer School on Trends in Concurrency, Bangalore, India, *June 2010*.
2. Organizer of the IFIP WG 2.4 Working group meeting. Bormio, Italy.
3. Organizer of the Second International Summer School on Trends in Concurrency, Prague, Czech Repbulic, *June 2008*.
4. Organizer of the Dagstuhl Seminar on *Types for Tools: Applications of Type Theoretic Techniques. June 2005*.
5. Organizer of the First International Summer School on Trends in Concurrency, Bertinoro, Italy, *July 2006*.

Steering Committees

1. Chair of the Steering Committee of the International Memory Management Symposium (ISMM), since 2010.
2. Member of the Steering Committee of the International Conference on Objects, Models, Components, Patterns (TOOLS Europe), since 2010.
3. Member of "Association Internationale pour les Téchnologies Objets", Steering Committee of the European Conference on Object-Oriented Programming (ECOOP), since 2008.
4. Member of the Steering Committee of the International Conference on Coordination Models and Languages (COORDINATION), 2007 – 2010.
5. Member of the Steering and Organizing Committee for the International Summer School on Trends in Concurrency, since 2006.
6. Member of the Steering Committee of the ACM/USENIX Conference on Virtual Execution Environments, 2005 – 2010.
7. Founding Member of the Steering Committee of the ACM TRANSACT Worshop, since 2005.
8. Member of the Steering Committee of the Java Technologies for Real-time and Embedded Systems (JTRes) workshop, since 2005.

Expert Committees

1. Member of the Java Expert Group for JSR 302: Safety Critical Java Technology.

General Chair

1. General Chair of the ACM SIGPLAN Conference on Languages, Programming Language Design and Implementation (PLDI), To be held in 2012.
2. General Chair of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems (LCTES), To be held in April 2011.
3. General Chair of the ACM SIGPLAN International Memory Management Symposium (ISMM), June 2010.
4. General Chair of the First ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing (TRANSACT), June 2006.

Program Chair

1. Program Chair of the *48th International Conference on Objects, Models, Components, Patterns (TOOLS Europe)*. Madrid Spain, 2010.
2. Program Chair of the *Java Technologies for Real-time and Embedded Systems* (JTRes) syposium. Prague, CZ, 2010.
3. Program Chair of the *European Conference on Object Oriented Programming (ECOOP)*, 2008.
4. Program Chair of the *9th International Conference on Coordination Models and Languages (COORDINATION)*, 2007.
5. Program Chair of the *First ACM/USENIX Conference on Virtual Execution Environments (VEE'05)*, 2005.
6. Program Chair of the *Formal Techniques for Java-like Programs* (FTfJP) workshop, 2005.
7. Program Chair of the *Jave Technologies for Real-time and Embedded Systems* (JTRes) workshop, 2005.

Conference Program Committees

1. **ESOP**: European Symposium on Programming, *2011*.
2. **POPL**: ACM SIGPLAN Conference on Principles of Programming Languages, *2011*.
3. **EUC**: IEEE/IFIP International Conference On Embedded and Ubiquitous Computing, *2010*.
4. **DLS**: Dynamic Language Symposium Conference, *2010*.
5. **RTSS**: IEEE International Real-Time Systems Symposium, *2010*.
6. **PODC**: Symposium on Principles of Distributed Computing, *2010*.
7. **ECOOP**: European Conference on Object Oriented Programming, *2010*.

8. **PLDI**: Conference on Programming Language Design and Implementation, *2010*.
9. **DATE**: Design, Automation & Test in Europe, Conference, *2010*.
10. **CATS**: Computing: The Australasian Theory Symposium, *2010*.
11. **RTSS**: IEEE International Real-Time Systems Symposium, *2009*.
12. **ECOOP**: European Conference on Object Oriented Programming, *2009*.
13. **ESOP**: European Symposium on Programming, *2009*.
14. **EUC**: IEEE/IFIP International Conference On Embedded and Ubiquitous Computing, *2008*.
15. **OOPSLA**: ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, *2008*.
16. **COORD**: International Conference on Coordination Models and Languages, *2008*.
17. **CSF**: IEEE Computer Security Foundations Symposium, *2008*.
18. **CC**: International Conference on Compiler Construction, *2008*.
19. **POPL**: ACM SIGPLAN Conference on Principles of Programming Languages, *2007*.
20. **OOPSLA**: ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, *2007*.
21. **ECOOP**: European Conference on Object Oriented Programming, *2007*.
22. **ESOP**: European Symposium on Programming, *2007*.
23. **PPPJ**: International conference on Principles and Practice of Programming in Java, *2006*.
24. **ICFP**: ACM SIGPLAN International Conference on Functional Programming, *2005*.
25. **OOPS**: Object Oriented Programming Languages and Systems *2005*.
26. **COORD**: International Conference on Coordination Models and Languages, *2005*.
27. **MASS**: Symposium on Multi-Agent Security and Survivability, *2005*.
28. **CD**: Component Deployment, *2004*.
29. **OOPSLA**: ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, *2004*.
30. **OOPS**: Object Oriented Programming Languages and Systems *2004*.
31. **CC**: International Conference on Compiler Construction, *2003*.
32. **MASS**: Symposium on Multi-Agent Security and Survivability, *2004*.
33. **ECOOP**: European Conference on Object Oriented Programming, *2003*.
34. **PLDI**: ACM SIGPLAN Conference on Programming Language Design and Implementation, *2002*.

35. **CD**: Component Deployment, *2002*.
36. **ECOOP**: European Conference on Object Oriented Programming, *2002*.
37. **ESOP**: European Symposium on Programming, *2002*.
38. **AISB**: Symposium on Software Mobility and Adaptive Behavior, *2001*.
39. **ASA/MA**: Agent Systems and Applications / Mobile Agents, *2001*.
40. **POPL**: ACM SIGPLAN Conference on Principles of Programming Languages, *2001*.
41. **SACMAT**: Symposium on Access Control Models and Technologies, *2001*.
42. **ECOOP**: European Conference on Object Oriented Programming, *2001*.
43. **OOPSLA**: ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, *2000*.
44. **ICALP**: International Conference on Automata, Languages and Programming, *2000*.
45. **JFLA**: Journées Francophones des Langages Applicatifs, *2000*.
46. **ECOOP**: European Conference on Object Oriented Programming, *2000*.
47. **JFLA**: Journées Francophones des Langages Applicatifs, *1998*.
48. **ECOOP**: European Conference on Object Oriented Programming, *1998*.
49. **JFLA**: Journées Francophones des Langages Applicatifs, *1995*.

Workshop Program Committees

1. **PLACES**: Programming Language Approaches to Concurrency and Communication-cEntric Software, *2010*.
2. **IWMSE**: Third International Workshop on Multicore Software Engineering, *2010*.
3. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2009*.
4. **STOP**: Script to Program Evolution, *2009*.
5. **PLACES**: Programming Language Approaches to Concurrency and Communication-cEntric Software, *2009*.
6. **VMIL.09**: Workshop on Virtual Machines and Intermediate Languages, *2009*.
7. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2008*.
8. **CPS**: International Workshop on Cyber-Physical Systems, *2008*.
9. **Bytecode**: Workshop on Bytecode Semantics, Verification Analysis and Transformation, *2007*.
10. **FOCLASA**: International Workshop on the Foundations of Coordination Languages and Software Architectures, *2007*.
11. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2007*.

12. **IWACO**: International Workshop on Aliasing, Confinement and Ownership, *2007*.
13. **PLAS**: Workshop on Programming Languages Analysis for Security, *2007*
14. **SecCo**: International Workshop on Security Issues in Coordination Models, Languages and Systems, *2007*.
15. **PWD**: Workshop on Program Analysis for Security and Safety (PASSWORD), *2006*
16. **ICOOO**: Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems. (ICOOOLPS), *2006*.
17. **CORD**: Workshop on Concurrency, Real-Time and Distribution in Eiffel (CORDIE), *2006*.
18. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2006*.
19. **FOCLASA**: International Workshop on the Foundations of Coordination Languages and Software Architectures, *2005*.
20. **AIOOL**: International Workshop on Abstract Interpretation of Object-Oriented Languages, *2005*.
21. **SecCo**: International Workshop on Security Issues in Coordination Models, Languages and Systems, *2005*.
22. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2005*.
23. **MOS**: Mobile Object Systems Workshop, *2005*.
24. **ACP4IS**: Workshop on Aspects, Components, and Patterns for Infrastructure Software, *2004*.
25. **FOCLASA**: International Workshop on the Foundations of Coordination Languages and Software Architectures, *2004*.
26. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2004*.
27. **SecCo**: International Workshop on Security Issues in Coordination Models, Languages and Systems, *2004*.
28. **MOS**: Mobile Object Systems Workshop, *2004*.
29. **CSJP**: Workshop on Concurrency and Synchronization in Java Programs, *2004*.
30. **JTRes**: Workshop on Java Technologies for Real-Time and Embedded Systems, *2003*.
31. **SecCo**: International Workshop on Security Issues in Coordination Models, Languages and Systems, *2003*.
32. **IWACO**: International Workshop on Aliasing, Confinement and Ownership, *2003*.
33. **MOS**: Mobile Object Systems Workshop, *2003*.
34. **ACP4IS**: Workshop on Aspects, Components, and Patterns for Infrastructure Software, *2003*.
35. **MOS**: Mobile Object Systems Workshop, *2002*.

36. **MOS**: Mobile Object Systems Workshop, *2001*.
37. **MOS**: Mobile Object Systems Workshop, *2000*.
38. **DOSW**: Distributed Object Security Workshop. *1999*.
39. **IWAOOS**: Intercontinental Workshop on Aliasing in Object-Oriented Systems. *1999*.
40. **MOS**: Mobile Object Systems Workshop, *1999*.
41. **WSIC**: Workshop on Secure Internet Computations. Organizer *1999*.
42. **MOS**: Mobile Object Systems Workshop, *1998*.
43. **MOS**: Mobile Object Systems Workshop, *1997*.
44. **MOS**: Mobile Object Systems Workshop, *1996*.
45. **MOS**: Mobile Object Systems Workshop, *1995*.

Reviewer for journals

1. ACM Computing Surveys
2. ACM TOPLAS
3. Science of Computer Programming
4. Computer Security
5. Theoretical Computer Science
6. Theory and Practice of Object Systems
7. Software Practice And Experience
8. Autonomous Agents and Multi-Agent Systems
9. Micro magazine
10. Journal of the ACM
11. International Journal of Information Security

Reviewer for grant applications

1. **CISE**: The Directorate for Computer and Information Science and Engineering. Panel on Cyber Phyiscal Systems, *2009*.
2. **CISE**: The Directorate for Computer and Information Science and Engineering. Panel on Cybertrust, *2005*.
3. **CISE**: The Directorate for Computer and Information Science and Engineering. Panel on Embedded and Hybrid Systems, *2004*.
4. **EPSRC**: The Engineering and Physical Sciences Research Council, UK. Grant applications, *2004*.

5. **NSERC**: The Natural Sciences and Engineering Research Council, Canada. Grant applications, *2004*.
6. **CISE**: The Directorate for Computer and Information Science and Engineering. Panel on Software Engineering and Languages, *2003*.
7. **NSERC**: The Natural Sciences and Engineering Research Council, Canada. Grant applications, *2003*.
8. **INRIA**: The French National Institute For Research in Computer Science and Control, France. Grant applications, *2003*.
9. **CISE**: The Directorate for Computer and Information Science and Engineering. Panel on Software Engineering and Languages, *2001*.

Department

1. Graduate admissions: *2009*.
2. Graduate admissions: *2009*.
3. Hiring committee: *2009*.
4. Graduate admissions: *2008*.
5. Graduate admissions: *2007*.
6. Colloquium chair: *2003, 2004*.
7. Graduate committee: *1999 – 2002*.
8. Volunteer on Graduate admission's day: *2000–2004*.
9. Graduate admissions: *2004 – 2005*.

School

1. Student appeal committee: *2004-05*.