Jan Vitek

Curriculum Vitae

j.vitek@neu.edu
Phone: (765) 494 6531
Nationality: USA
Date of Birth: 6.9.1966

March 22, 2015

Northeastern University
360 Huntington Ave.
Boston, MA 02115

Research Interests

Programming languages, Compilers, Software engineering, Embedded and real-time systems, Data analytics, Concurrency, Software security.

Employment

Professor 7/14 –
Northeastern University.

Professor 8/10 – 7/14
Purdue University.

Visiting Professor 6/12 – 6/13
Stanford University.

Visiting Researcher 9/12 – 7/13
Oracle Labs, Redwood shores.

Scientific Advisor 1/12 – 6/13
0xData, Mountainview.

Academic Visitor 9/06 – 5/11
IBM T.J Watson.

Associate Professor 8/05 – 8/10
Purdue University.

Visiting Professor 7/08 – 9/08
INRIA, Rocquencourt.

Visiting Professor 1/06 – 7/06
EPFL.

Assistant Professor 8/99 – 7/05
Purdue University.

Research Assistant 9/94 – 7/99, 9/89 – 8/90
University of Geneva,

Software Consultant 8/92 – 8/93
International Labor Organization, Geneva.

Research Assistant 9/90 – 7/92
University of Victoria, CA

Education

University of Geneva 99
PhD in Information Systems
"The Seal Calculus – A calculus of mobile computations" Advisor: D. Tsichritzis

University of Victoria 95
MSc in Computer Science
"Compact Dispatch Tables for Dynamically Typed Languages" Advisor: R. N. Horspool

Awards and Honors

11 **Purdue University Faculty Scholar**
11 **Microsoft SEIF Research Award**
11 **Purdue Undergraduate Advising Award**
06 **IBM Faculty Award**
01 **NSF CAREER Award**

Positions in Scholarly Associations

Chair of the ACM Special Interest Group on Programming Languages, elected July 12.

Vice President of IFIP Working Group 2.4, elected July 11.

Vice President of the Association Internationale pour les Technologies Objets, elected June 10.

Member of JSR 302 expert group Safety Critical Java since 07.

Member Scientific Committee of Ecole des Mines de Nantes since 12.

Member CominLabs International Advisory Committee since July 11.

Steering Committees

SC Member of PLDI since 11.

SC Member of OOPSLA since 12.

SC Member of IFCP since 12.

SC Member of SPLASH since 12.

SC Member of COORDINATION 07 – 10.

Founding SC Member of International Summer School on Trends in Concurrency since 06.

SC Member VEE 05 – 10.

Founding SC Member of the TRANSACT workshop since 05.

Member of the Java Technologies for Real-time and Embedded Systems workshop since 05.

SC Chair of ISMM 10–11.

Member of TOOLS Europe, 10 – 11.

Research highlights

Dr. Vitek's research focuses on the programmatic interface between man and machine with the overarching goal to devise higher-level abstraction for specifying computational tasks. His research touches on programming languages in all of their forms and on tools to assist their use and efficient implementation.

Confinement and Ownership. In [90] the foundations of what was to be called *ownership types* were defined. The seemlingly simple idea of encoding constraints on the runtime structure of the heap into object class definitions was influencial with hundreds of citations and many research offshoots. In follow up, Vitek looked at how to reduce the annotation burden, proposing the notions of confined types [87], [24]. Confinement aims to codify software engineering best practice already familiar to programmers. Confinement can be inferred [83], [12] and is sound [77], [15], it also inspired the region types used for Real-time Java programs in [8], [69], [10], [62] and was adopted in the StreamFlex stream programming system [6], [51], [52], [55], [57] as well as for enforcing thread locality [50]. Another variant of ownership types is part of the Safety Critical Java standard [103], [111].

Dynamic languages. The goal this work is to evolve scripts, rapidly constructed dynamic code fragments, into programs, robust well-engineered software components [109]. One side of the research focuses on JavaScript [110] and includes the first large scale study, ranging of over thousands of programs, of the dynamic behavior of real JavaScript applications [43]. This provided unique insights into programmers' use reflection [41]. JSBench [40] is a tool for transforming web sites into benchmark built in an unusual collaboration between Mozilla and Microsoft and eventually adopted by Apple as their standard browser performance benchmark. This was follwed by [36], which presented the first tool for automatically inferring the behavior of reflective `eval` calls from dynamic profiles. The other side of Dr. Vitek's research, in collaboration with IBM, involves the design of a next generation scripting language code named Thorn [48], [54]. A noteable features of Thorn is its novel type system which allows user to smoothly move from dynamic to static type with the addition of *like types* [45] which provide local soundness but do not make guarantees past function boundaries.

Scalable Data Analytics. One of the practical challenges that comes with ever increasing data set sizes is how to scale up computations from subsets of large data set to the whole thing. With colleagues at Stanford, Dr. Vitek investigated domain specific languages for high-performance computing [31]. Dr. Vitek's other major projects are related to the R programming language. R is a widely used vehicle for expressing statistical computations but has serious limitations in its ability to handle larger data sets. Dr. Vitek looked at the semantics of the language and analyzed close to 4 million lines of code [34]. Using the insights gainned in the corpus study and in collaboration with Oracle Labs, Dr. Vitek is leading an effort to provide a new implementation of the R language that will allow to scale R computations in time and space.

Language Implementation. Vitek's earliest lasting contributions were novel algorithms for the two most frequent operations in object oriented programs, namely, method dispatch and type tests. He proposed a compact solution to the dispatching problem in dynamically typed languages [96] and later reduced space requirements [94] and looked architectural impacts [95]. Dr. Vitek also investigated flexible dispatching policies [66], [9]. For type tests the challenge was to find data structures, optimized for space and time, for answering reachability queries on a lattice. He proposed a compact bitset encoding based on graph coloring [92], presented a constant time algorithm [91] and devised an encoding that could be easily recomputed on loading of new clases [79]. These techniques were used in the Fiji Java virtual machine [108], [44]. Dr. Vitek led the Ovm project which delivered an open source framework for building language runtimes. By design, Ovm can be specialized and assembled into a configuration which is customized for a particular problem domain. Ovm was used in the first Real-Time Java virtual machine to be deployed and flight tested on a Boeing-built UAV. The project was documented with a number of papers [7], [11], [18], [53], [56], [58], [61], [62], [63], [64], [80].

Real-Time and Embedded Computing. Schism [42] is the most efficient algorithm real-time garbage collection algortihm [47] and is used in production [44]. Vitek investigated real-time memory management techniques in [73], [60], [58], [56], [53], [7], [108]. For real-time stream processing systems, Vitek showed [6] that periods of 50 μs can be reliably achieved without losing the portability or the memory safety of Java. StreamFlex was further developed in [51], [55], [57], it offers a dataflow programming model with zero copy guarantees for streaming data and a novel use of software transactional memory [65] for communication with non-real-time tasks. The technology was transfered to IBM [52]. An ongoing project aims to formalize the guarantees needed for safety critical applications in Java [107], [111]. Part of this work is being done in the context of the JSR-302 Safety Critical Java expert group. One early result is the development of new buffered memory model for the Java language that can be used for proving the correctness of compiler optimizations [32].

Concurrency control. A study of the DACAPO benchmark suite [35] demonstrated the potential for parallel execution in Java and the challenges to achieving it. The concept of *atomic sets* extends Java with a high-level data-centric concurrency control abstraction. Atomic set leverage ideas from ownership types and confinement to decrease annotation burden, most use-cases require only a handful of annotations. Dr. Vitek investigated transactional abstractions, giving semantics to software transactional memory in [74], [17] and [75], tuning the garbage collector to be transaction-aware in [101], and providing the first non-trivial benchmark suite [59] for transactional memory. Looking at the problem of ensuring predictable performance in hard real-time environment, Dr. Vitek proposed *preemptible atomic regions* [65], [116] which were later adopted in StreamFlex [51]. In another project, he looked at architectural support for real-time transactions [3], [100], [46].

Software Security. In recent work Dr. Vitek combined ideas from software transactions with dynamic ownership tracking to define a new security enforcement infrastructure for JavaScript programs based on delimited histories with revocation [29] and applied it web mashups. Earlier, Vitek proposed an efficient intrusion detection technique for C programs based on inlining automata [68]. In more formal investigations Vitek defined the box-π calculus, a minimal extension of the π-calculus with encapsulation [84], [22], [85]. Distributed access control was viewed as a weak form of information flow control in [78], [76] and in [23] fine-grained access control was also to a key-value store.

Mobile Computations. Dr. Vitek devised new abstractions for programming WANs. The Seal calculus is a core model of mobile computational systems with a Java implementation [25],[149], [86], [88], [119], [123], [126]. The Seal calculus was among the first attempts to explore the design space of mobile programming languages from both theoretical and practical angles [19], [118], [125]. Location mobility as a result of process synchronization, first introduced in Seal as a natural extension of π-calculus communication primitives, was adopted in several other works.

Towards Rigor in Experimental Computer Science. Dr. Vitek has long be interested in experimental methodologies. He worked on creating and understanding benchmarks for software transaction [59], real-time computing [5], [106], web applications [40], concurrent programming [35]. Recently, he advocated for supporting repeatability of all published experimental works [37], [147]. Science advance faster when one can build on existing results, and when new ideas can easily be measured against the state of the art. This is exceedingly difficult in an environment that does not reward the production of reusable software artifacts. Dr. Vitek's goal is to get to the point where any published idea that has been evaluated, measured, or benchmarked is accompanied by the artifact that embodies it. *Repeatability* can be summed up as the act of checking the claims made in a paper by re-running a bundled software artifact. Repeatability is a cheap and easy test which clarifies the scientific contribution of a paper. It should become a standard feature of the dissemination of research results [145]. Together with Shriram Krishnamurthi, Dr. Vitek has started to organize an artifact evaluation process in some of the main programming language conferences (OOPSLA, ECOOP, SAS, PLDI). Early feedback from authors who submitted artifacts is encouraging, it is conceivable that artifacts may become a standard feature of these conferences.

Publications

Journals

[1] S. Jagannathan, V. Laporte, G. Petri, D. Pichardie, J. Vitek. Atomicity Refinement for Verified Compilation. *Accepted for Publication in ACM Transaction on Programming Languages and Systems*, 26 pages, 2014.

[2] J. Dolby, C. Hammer, D. Marino, F. Tip, M. Vaziri, J. Vitek. A Data-Centric Approach to Synchronization. *ACM Transaction on Programming Languages and Systems*, (**TOPLAS**) 48 pages, 34(1):4, 2012.

[3] F. Meawad, K. Iyer, M. Schoeberl and J. Vitek. Micro-transactions for concurrent data structures. In *Concurrency and Computation: Practice and Experience* (**CCPE**), 2012. `doi:10.1002/cpe.2985`

[4] T. Kalibera, F. Pizlo, T. Hosking, J. Vitek. Scheduling real-time garbage collection on uniprocessors. In *ACM Transaction on Computer Systems*, (**TOCS**), 29:3, pp. 8:1–8:29, 2011.

[5] 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*, (**CS:PE**), 23(14), pp. 1679–1700, 2011. `doi:10.1002/cpe.1677`

[6] 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.

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

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

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

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

[11] 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)*, (**TECS**) 7(1), pp. 1–49 pages, 2007.

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

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

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

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

[16] 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.

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

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

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

[20] 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*, (**SAGMB**), Berkeley Electronic Press, Volume 1, Issue 1, 2004.

[21] B. Carbunar, M. T. Valente and J. Vitek. Lime revisited. In *Mathematical Structures in Computer Science*, (**MSCS**) Cambridge University Press, 14(3), pp. 397–419, 2004.

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

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

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

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

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

[27] 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.

Refereed Conference Proceedings

[28] T. Kalibera, P. Maj, F. Morandat, J. Vitek. A Fast Abstract Syntax Tree Interpreter for R. In *Conference on Virtual Execution Environments* (**VEE**), Salt Lake City, March 2014.

[29] G. Richards, C. Hammer, S. Jagannathan, F. Zappa Nardelli and J. Vitek. Flexible Access Control Policies with Delimited Histories and Revocation In *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), Indianapolis, October 2013.

[30] D. Marino, C. Hammer, J. Dolby, M. Vaziri, F. Tip, and J. Vitek, Detecting deadlock in programs with data-centric synchronization. In *International Conference on Software Engineering* (**ICSE**), pp. 322-331, San Francisco, May 2013.

[31] Z. DeVito, J. Hegarty, A. Aiken, P. Hanrahan, and J. Vitek. Terra: a multi-stage language for high-performance computing. In *Programming Language Design and Implementation Conference* (**PLDI**) pp. 105-116, Seattle, June 2013.

[32] D. Demange, V. Laporte, L. Zhao, S. Jagannathan, D. Pichardie, and J. Vitek. Plan B: a buffered memory model for Java. In *ACM Symposium on Principles of Programming Languages* (**POPL**), pp. 329-342, Rome, January 2013.

[33] A. Bouakaz, J.-P. Talpin and J. Vitek. Affine Data-Flow Graphs for the Synthesis of Hard Real-Time Applications. In *Conference on Application of Concurrency to System Design* (**ACSD**) pp 183-192, Hamburg, July 2012.

[34] F. Morandat, B. Hill, L. Osvald and J. Vitek, Evaluating the Design of the R Language. In *European Conference on Object-Oriented Programming* (**ECOOP**) Beijing, June 2012.

[35] T. Kalibera, M. Mole, R. E. Jones and J. Vitek. A black-box approach to understanding concurrency in DaCapo. In *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), pp. 335-354, Tucson, October 2012.

[36] F. Meawad, G. Richards, F. Morandat and J. Vitek. Eval begone!: semi-automated removal of eval from JavaScript programs. In *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), pp. 607-620, Tucson, October 2012.

[37] J. Vitek, T. Kalibera. Repeatability, reproducibility, and rigor in systems research. In *International Conference on Embedded Software* (**EMSOFT**), Taipei, October, 2011.

[38] J. Vitek. Virtualizing real-time embedded systems with Java. In *Design Automation Conference* (**DAC**), San Diego, June, 2011.

[39] A. Milanova, J. Vitek. Static Dominance Inference. In *Conference on Objects, Models, Components, Patterns* (**TOOLS**), Zurich, June, 2011.

[40] G. Richards, A. Gal, B. Eich, J. Vitek. Automated Construction of JavaScript Benchmarks. In *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), pages 677–694, 2011.

[41] 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. In *European Conference on Object-Oriented Programming* (**ECOOP**), pages 52–78, Lancaster July 2011.

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

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

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

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

[46] M. Schoeberl, F. Brandner, J. Vitek. RTTM: Real-Time Transactional Memory. In *Symposium on Applied Computing, Real-Time Systems Track* (**SAC**), Sierre, March 2010.

[47] T. Kalibera, F. Pizlo, A. Hosking, J. Vitek. Scheduling Hard Real-time Garbage Collection. In *Real-Time Systems Symposium* (**RTSS**), Washington D.C., December 2009.

[48] 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 *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), 20 p., Orlando, October 2009.

[49] T. Kalibera, M. Prochazka, F. Pizlo, J. Vitek, M. Zulianello, M. Decky. Real-time Java in Space: Potential Benefits and Open Challenges.

In *DAta Systems In Aerospace* (**DASIA**), 10 pp., Istanbul, June 2009.

[50] T. Wrigstad, F. Pizlo, F. Meawad, L. Zhao, J. Vitek. Loci: Simple Thread-Locality for Java. In *European Conference on Object Oriented Programming* (**ECOOP**), Genova, June 2009.

[51] 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, June 2009.

[52] J. Auerbach, J. H. Spring, D. Bacon, R. Guerraoui, J. Vitek. A Unified Restricted Thread Programming Model for Java. In *Conference on Languages, Compilers, and Tools for Embedded Systems* (**LCTES**), Tucson, pp. 1–11, June 2008.

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

[54] 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, pp. 150–166, January 2008.

[55] 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, October 2007.

[56] F. Pizlo, A. Hosking, J. Vitek. Hierarchical Real-time Garbage Collection. In *Conference on Languages, Compilers, and Tools for Embedded Systems* (**LCTES**), pp. 123–133, San Diego, June 2007.

[57] J.H. Spring, F. Pizlo, R. Guerraoui, J. Vitek. Reflexes: Abstractions for Highly Responsive Systems. In *Conference on Virtual Execution Environments* (**VEE**), pp. 191–201, San Diego, June 2007.

[58] J. Baker, A. Cunei, F. Pizlo, J. Vitek. Accurate Garbage Collection in Uncooperative Environments with Lazy Pointer Stacks. In *Conference on Compiler Construction* (**CC**), pp. 64–79, Braga,q March 2007.

[59] M. Kalpka, R. Guerraoui, J. Vitek. STMBench7: A Benchmark for Software Transactional Memory. In *European Conference on Computer Systems* (**EUROSYS**), Lisbon, pp. 315–324, March 2007.

[60] F. Pizlo, J. Vitek. An Empirical Evalutation of Memory Management Alternatives for Real-time Java. In *27th Real-Time Systems Symposium* (**RTSS**), pp. 35–46, Rio de Janeiro, December 2006.

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

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

[63] A. Cunei, J. Vitek. A New Approach to Real-time Checkpointing. In *Conference on Virtual Execution Environments* (**VEE**), pp. 68–77, Ottawa, June 2006.

[64] 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 *Real-Time and Embedded Technology and Applications Symposium* (**RTAS**), pp. 384–396, San Jose, April 2006.

[65] J. Manson, J. Baker, A. Cunei, S. Jagannathan, M. Prochazka, B. Xin, J. Vitek. Preemptible Atomic Regions for Real-time Java. In *Real-Time Systems Symposium* (**RTSS**), pp. 62–71, Miami, December 2005.

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

[67] 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**), pp. 236–245, Vienna, September 2005.

[68] R. Gopalakrishna, E. Spafford, J. Vitek. Efficient Intrusion Detection using Automaton Inlining. In *Symposium on Security and Privacy* (**S&P**), pp. 18–31, Oakland, May 2005.

[69] T. Zhao, J. Noble, J. Vitek, Scoped Types for Real-time Java, In *Real-Time Systems Symposium* (**RTSS**), pp. 241–251, Lisbon, December 2004.

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

[71] B. Carbunar, A. Grama, J. Vitek. Coverage Preserving Redundancy Elimination in Sensor Networks. In *Conference on Sensor and Ad-Hoc Communications and Networks* (**SECON**). 2004.

[72] B. Carbunar, A. Grama and J. Vitek. Distributed and Dynamic Voronoi Overlays for Coverage Detection and Distributed Hash Tables

in Ad-Hoc Networks. In *International Conference on Parallel and Distributed Systems* (**ICPADS**), pp. 549–559, Newport Beach, July 2004.

[73] F. Pizlo, J. Fox, D. Holmes and J. Vitek. Real-Time Java scoped memory: design patterns and semantics. In *International Symposium on Object-oriented Real-Time Distributed Computing* (**ISORC**), pp. 101–112, Vienna, May 2004.

[74] J. Vitek, S. Jagannathan, A. Welc and A.L. Hosking. A semantic framework for designer transactions. In *European Symposium on Programming* (**ESOP**), pp. 249–263, Barcelona, April 2004.

[75] S. Jagannathan and J. Vitek. Optimistic concurrency semantics for transactions in coordination languages. In *Conference on Coordination Models and Languages* (**COORDINATION**), pp. 183–198, Pisa, March 2004.

[76] T. Chothia, D. Dugganand J. Vitek, Principals, Policies and Keys in a Secure Distributed Programming Language. In *Computer Security Foundations* (**CSF**), pp. 170–180 Turku, July, 2003.

[77] T. Zhao, J. Palsberg and J. Vitek. Lightweight confinement for featherweight Java. In *Conference on Object-Oriented Programming Systems and Languages* (**OOPSLA**), pp. 135–148, San Diego, October 2003.

[78] T. Chothia, D. Duggan and J. Vitek. Type-based distributed access control. In *Computer Security Foundations Workshop* (**CSFW**), pp. 170 – 187, Pacific Grove, July 2003.

[79] K. Palacz and J. Vitek. Subtype tests in real time. In *European Conference on Object Oriented Programming* (**ECOOP**), pp. 378–404, Darmstadt, July 2003.

[80] K. Palacz, J. Baker, C. Flack, C. Grothoff, H. Yamauchi and J. Vitek. intermediate representation. In *Workshop on Interpreters, Virtual Machines and Emulators* (**IVME**), pp. 1–12, San Diego, June 2003.

[81] K. Palacz, G. Czaikowski, L. Daynes and J. Vitek. Incommunicado: a communication substrate for Isolates. In *Conference on Object-Oriented Programming Systems and Languages* (**OOPSLA**), pp. 262–274, Seattle, November 2002.

[82] B. Carbunar, M. T. Valente and J. Vitek. Lime revisited. In *International Conference on Mobile Agents* (**MA**), pp. 54–69, Atlanta, December 2001.

[83] C. Grothoff, J. Palsberg and J. Vitek. Encapsulating objects with confined types. In *Conference on Object-Oriented Programming Systems and Languages* (**OOPSLA**), pp. 241–255, Florida, October 2001.

[84] P. Sewell and J. Vitek. Secure composition of untrusted code: wrappers and causality types. In *Computer Security Foundations Workshop* (**CSFW**), pp. 269–284, Cambridge, July 2000.

[85] P. Sewell and J. Vitek. Secure composition of insecure components. In *Computer Security Foundations Workshop* (**CSFW**), pp. 136–150, Mordano, June 1999.

[86] C. Bryce, M. Oriol and J. Vitek. Secure object spaces: a coordination model for agents. In *International Conference on Coordination Models and Languages* (**COORDINATION**), pp. 4–20, Amsterdam, April 1999.

[87] B. Bokowski and J. Vitek. Confined types. In *ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), pp. 82–97, Denver, October 1999.

[88] J. Vitek and C. Bryce. Security for mobile code: the JavaSeal experiment. In *Agent Systems and Applications Mobile Agents* (**ASA/MA**), pp. 103–118, Palm Springs, October 1999.

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

[90] J. Noble, J. Vitek and J. Potter. Flexible alias protection. In *European Conference on Object-Oriented Programming* (**ECOOP**), pp. 158–185, Brussels, July 1998.

[91] J. Vitek, R.N. Horspool and A. Krall. Efficient type inclusion tests. In *Conference on Object-Oriented Programming Systems, Languages and Applications* (**OOPSLA**), pp. 142–157, San Jose, October 1997.

[92] A. Krall, J. Vitek and R.N. Horspool. Near optimal hierarchical encoding of types. In *In European Conference on Object-Oriented Programming* (**ECOOP**), pp. 128–146, Jyvaskyla, June 1997.

[93] A. Krall and J. Vitek. On extending Java. In *Joint Modular Languages Conference (***JMLC***)*, pp. 321–325, Linz, March 1997.

[94] J. Vitek and R. N. Horspool. Compact dispatch tables for dynamically typed object oriented languages. In *Conference on Compiler Construction* (**CC**), pp. 309–326, Linkoping, April 1996.

[95] K. Driesen, U. Hölzle and J. Vitek. Message dispatch on pipelined processors. In *European Conference on Object-Oriented Programming* (**ECOOP**), pp. 253–283, Åarhus, August 1995.

[96] J. Vitek and R. N. Horspool. Taming message passing: efficient method look-up for dynamically typed languages. In *European Conference on Object-Oriented Programming* (**ECOOP**), pp. 432–449, Bologna, July 1994.

[97] J. Vitek, R.N. Horspool and J. Uhl. Compile-time analysis of object-oriented programs. In *Conference on Compiler Construction* (**CC**), pp. 236–250, Paderborn, October 1992.

[98] R. N. Horspool and J. Vitek. Static analysis of PostScript code. In *International Conference on Computer Languages* (**ICCL**), pp. 14–23, Oakland, April 1992.

Refereed Workshop Publications

[99] R. Macnak, F. Morandat, B. Hill, L. Osvald and J. Vitek. TraceR: A framework for understanding R performance. In *International R Users Meeting* (**UseR!**), Nashville, June, 2012.

[100] F. Meawad, K. Iyer, M. Schoeberl, J. Vitek. Real-Time Wait-free Queues using Micro-Transactions. In *International Workshop on Java Technologies for Real-time and Embedded Systems* (**JTRES**), York, September 2011.

[101] F. Meawad, B. Macnak, J. Vitek. Collecting Transactional Garbage. In TRANSACT, June, 2011.

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

[103] D. Tang, A. Plsek, J. Vitek. Static Checking of Safety Critical Java Annotations. In *Workshop on Java Technologies for Real-time and Embedded Systems* (**JTRES**), Prague, September 2010.

[104] A. Plsek, L. Zhao, V. Sahin, D. Tang, T. Kalibera, J. Vitek. Developing Safety Critical Java applications with oSCJ/L0. In *Workshop on Java Technologies for Real-time and Embedded Systems* (**JTRES**), Prague, September 2010.

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

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

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

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

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

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

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

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

[113] 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**), Berlin, July 2007.

[114] 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 October 2006.

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

[116] 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, July 2004.

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

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

[119] J. Vitek. New Paradigms in distributed computing. In *European Research Seminar in Advanced Distributed Systems* (**ERSADS**), pp. 117–122, Zinal March 1997.

[120] J. Vitek. Secure object spaces. In *Workshop on Mobile Object Systems* (**MOS**), pp. 41-48, Linz, July 1996.

Book Chapters

[121] J. Vitek. Atomicity in Real-time Computing. In *Patterns, Programming and Everything*, K. Breitman and N. Horspool (Ed.), Springer, June, 2012.

[122] D. Tang, A. Plsek, J. Vitek. Memory Safety for Safety Critical Java. In *Distributed, Embedded and Real-Time Java Systems*, T. Higuera-Toledano and A. Wellings (Eds)., pages 235–254, Springer, February, 2012.

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

[124] 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.

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

[126] 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, 1997.

Books

[127] J. Vitek, H. Lin and F. Tip (Eds.). Proceedings of the *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Beijing, June 2012.

[128] J. Vitek, B. De Sutter (Eds.). Proceedings of the ACM SIGPLAN/SIGBED 2011 conference on Languages, compilers, and tools for embedded systems, LCTES, Chicago, April, 2011.

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

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

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

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

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

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

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

Journal Special Issues

[136] B. De Sutter and J. Vitek. Introduction to the special section on LCTES'11. In *ACM Transaction Embedded Computer Systems* 12(1s): 38, 2013.

[137] J. Vitek and T. Kalibera. Introduction to the Special Issue on Java Technologies for Real-Time and Embedded Systems. In *Concurrency and Computation: Practice and Experience*, 24(8): 751-752, 2012.

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

[139] 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.

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

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

[142] 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). Boston, 1998.

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

Other Publications

[144] J. Vitek. The Case for the Three Rs of Systems Research: Repeatability, Reproducibility and Rigor (Keynote). In *Conference on Virtual Execution Environments*, (**VEE**), Salt Lake City, March 2014.

[145] S. Krishnamurthi and J. Vitek. Should Software Conferences Respect Software? To appear in *Communications of the ACM* (**CACM**), 2014.

[146] R. Hirschfeld, S. Krishnamurthi, J. Vitek. Foundations for Scripting Languages, pp 1–18, Dagstuhl Reports, 2192-5283, 2012.

[147] J. Vitek and T. Kalibera. R3: Repeatability, reproducibility and rigor. In ACM SIGPLAN Notices, 47(4a), pp. 30-36, April 2012.

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

[149] J. Vitek, C. Bryce and W. Binder. Designing JavaSeal, or how to make Java safe for agents. In *Electronic Commerce Objects*, pp. 105-126, U. of Geneva, 1998.

Invited Lectures

- The Case for the Three Rs of Systems Research: Repeatability, Reproducibility and Rigor. Keynote talk at the *Conference on Virtual Execution Environments*, Salt Lake City, March 2014.

* JavaScript Programmers Hate You: An ode to dynamic languages. Invited talk at the Workshop on Software Correctness and Reliability, Zurich, October 2013.

* Planet Dynamic or: How I Learned to Stop Worrying and Love Reflection. Invited talk at the *SIGPLAN Programming Languages Mentoring Workshop*, Rome, January 2013.

* JavaScript Programmers Hate You. Keynote talk at at *Formal Techniques for Java-like Programs*, Montpellier, July 2013.

* Planet Dynamic or: How I Learned to Stop Worrying and Love Reflection. Keynote talk at the *10th Asian Symposium on Programming Languages and Systems*, Kyoto, December 12.

* Repeatability, Reproducibility and Rigor. Invited talk at the *Conference on Languages Compilers and Tools for Embedded Systems*, Beijing, June 12.

* Thorn: Objects, Scripts and more... Invited talk at the *Concurrent Objects and Beyond* Symposium in Honor of Professor Akinori Yonezawa's 65th Birthday, Kobe, June 12.

* Repeatability, Reproducibility and Rigor in Systems Research, Keynote talk at the *International Conference on Embedded Software*, Taiwan, 11.

* The Rise of Dynamic Language for Scientific Computing, Invited talk at the *Microsoft Faculty Summit*, Redmond, July 11.

* The Rise of Dynamic Language, Lecture at the *ECOOP Summer School*, Lancaster, July 11.

* 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, June 10.

* Is Java Ready for Real-time?, Invited talk at the *Midwest Verification Day* (MVD), U of Iowa, September, 10.

* Of Scripts and Programs: Tall tales, Urban Legends and Future Prospects, Keynote talk at the *Dynamic Languages Symposium*, Orlando, October 09.

* Programming Models for Concurrency and Real-time. Keynote talk at the *47th International Conference on Objects, Models, Components, Patterns (TOOLS)*, Zurich, July 09.

* Memory Management for Hard Real-time Systems. Invited talk at the *Workshop on Virtual Machines and Intermediate Languages for emerging modularization mechanisms*, Nashville, October 08.

* Programming models for Concurrency and Real-time. Invited talk at *XII Brazilian Symposium on Programming Languages*, Fortaleza, August 08.

* Programming models for Concurrency and Real-time. Invited talk at *Programming Language Approaches to Concurrency and Communication-cEntric Software*, June 08, Oslo.

* Semantics-based Intrusion Detection, Invited Talk at the *Foundations of Computer Security*, Chicago, June 05.

* Java for Hard Real-Time, Invited Talk at the *Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems*, Nantes, July 06.

* Advances in Intrusion Detection, Keynote talk at the *Program Analysis for Security and Safety Workshop (PASSWORD)*, Nantes, July 06.

Talks at Universities and Other Institutions

Indiana University (14), Northeastern University (14), Institute of Science and Technology Austria (14). Northeastern University, Boston (13), Samsung, CA (13), Facebook, CA (13), Charles University, Prague, (13), Czech Technical University, Prague (13). Google, CA, (12), Stanford, CA, (12), UIUC, Ill, (12), Tsinghua, CN, (12). INRIA Rocquencourt, Paris (11), Initiative de Recherche et Innovation sur le Logiciel Libre, Paris (11), Laboratoire d'Informatique de Paris 6, Paris (11), Microsoft Research, Redmond (11), ETHZ, Zurich (11). INRIA-Rennes (10). Imperial College (09), Microsoft Research (09), Brown University (09), EPFL (09), University of Central Florida (09). University of Lugano, CH (08), INRIA Rocquencourt, FR (08), INRIA Rennes, FR (08), Ecole Polytechnique Fédérale, Lausanne, CH (08), Imperial College, UK (08), University of California, Los Angeles, CA (08), Edinburgh University, UK (07), IBM T.J. Watson, NY (07), Charles University, Prague, CZ (07), Microsoft Research, WA (07), IBM T.J. Watson, NY (06), Swiss Federal Institute of Technology, Zurich, CH(06), University of Bern (06), Ecole Polytechnique Fédérale, Lausanne (06), Portland State University (06), Microsoft Research (06), University of Utah (06), University of Washington (05), Carnegie Mellon University (05), University of Victoria (05), University of Alberta (05), University of Nice (03), Tokyo University (01), University of Waterloo (1999), University of Syracuse (1999), University of Pennsylvania (1999), University of Toronto (1999), University of Victoria (1999), University of Rennes (1999).

Talks at International Meetings

* Planet Dynamic or: How I Learned to Stop Worrying and Love Reflection, NSA HCSS Conference, May 13
* R in Java. *UseR!*, Albacete, July 13.
* Understanding R. *Foundations of Scripting Languages*, Dagstuhl, January, 12.
* Repeatability, Reproducibility and Rigor. *IFIP Working Group 2.4* , Vadvestena, Sweden, 12.
* Taming the Tiger: How to scale R to bigger data. *Purdue Symposium on Statistics*, West Lafayette, June 12.
* Evaluating the Design of the R Language. *European Conference on Object Oriented Programming*, Beijing, June 12.
* CDx: A Family of Real-time Java Benchmarks. *Workshop on Java Technologies for Real-time and Embedded Systems* , Madrid, September 09.
* A Technology Compatibility Kit for Safety Critical Java. *Workshop on Java Technologies for Real-time and Embedded Systems*, Madrid, September 09.
* Software Hardening: A Research Agenda. *Workshop on Script to Program Evolution*, Genoa, July 09.
* Programming Real-time Embedded Systems in Java. Summer school part of the *Wroclaw Information Technology Initiative*, Wroclaw, May 09.
* Java for Safety-Critical Applications, *Certification of Safety-Critical Software Controlled Systems*, York, March 09.
* Large-Scale Embedded Programming, *Software Quality Symposium*, ETHZ, Zurich, 07.
* Programming Highly Responsive Systems, *IFIP Working Group 2.4*, Lake Arrowhead, CA, 07.
* Transactions and Composability: Transactions Considered Harmful? *IBM Workshop on Transactional Memory and Programming Technologies*, Armonk, March 07.
* Data-centric Synchronization, *IBM Workshop on Transactional Memory and Programming Technologies*, Armonk, March 07.
* How not to get a job in research, *Summer School on Trends in Concurrency,* Bertinoro, July 06.
* Scoped Types and Aspects for Real-Time Systems, *European Conference on Object Oriented Programming*, Nantes, July 06.
* Real-time Java in Avionics Applications. *Real-Time and Embedded Technology and Applications Symposium*, 06.
* Preemptible Atomics, *IFIP Working Group 2.4* , Jackson's Mill, West Virginia, October, 05.
* Preemptible Atomics, *IFIP Working Group 2.4* , Jackson's
* Memory Safe RTSJ Programming, *Safety & Mission Critical Workshop*, Palo Alto, September 05.
* Preemptible Atomic Regions, SUN Microsystems, August 05.
* Adopting Ownership Types, *Dagstuhl Tool for Types Workshop*, Dagstuhl, June, 05.
* Stealth Types, *Foundations of Object-Oriented Languages* panel on Extreme Typing, Long Beach, CA, January 11, 05.
* The Real-time Specification for Java: issues and opportunities, *IFIP Working Group 2.4*, Baden, January 05.
* Scoped Types for Real-time Java, *International Real-Time Systems Symposium* Lisbon, December 04.
* A semantic framework for designer transactions, *European Symposium on Programming*, Barcelona, April 04.
* Transactional Facilities for Java. *Conference on Object Oriented Programming Systems, Languages and Applications*, Vancouver, 04.
* Security and Coordination. *School on Foundations of Security Analysis and Design*, Italy, September 04.
* Real-time Java with the Ovm virtual machine. *Real-time Java Symposium*, DARPA, Arlington, July 04.
* Engineering Intermediate Representations, *IFIP Working Group 2.4*, Santa Cruz, July 03.
* Lightweight confinement for featherweight Java, *Conference on Object-Oriented Programming Systems and Languages*, San Diego, October 03.
* Subtype tests in real time. In *European Conference on Object Oriented Programming*, Darmstadt, July 03.
* Engineering a customizable intermediate representation, *Workshop on Interpreters, Virtual Machines and Emulators*, San Diego, June 03.
* Encapsulating objects with confined types, *Conference on Object-Oriented Programming Systems and Languages*, Tempa, October 01.
* Confined Types, *IFIP Working Group 2.4*, Italy, July 01.
* Confined types, *Conference on Object-Oriented Programming Systems, Languages and Applications*, Denver, October 1999.
* Efficient type inclusion tests, *Conference on Object-Oriented Programming Systems, Languages and Applications*, San Jose, October 1997.

* Near optimal hierarchical encoding of types, *European Conference on Object-Oriented Programming*, Jyvaskyla, June 1997.
* Compact dispatch tables for dynamically typed object oriented languages, *Conference on Compiler Construction*, Linkoping, Sweden, April 1996.
* Taming message passing: efficient method look-up for dynamically typed languages, *European Conference on Object-Oriented Programming*, Bologna, July 1994.
* Compile-time analysis of object-oriented programs, *Conference on Compiler Construction*, Paderborn, October 1992.

Software artifacts

Vitek has acted as team leader and developer on several software systems. All are released in open source.

1. **FastR:** A new implementation of the R programming language with a runtime specializing interpreter running on top of a JVM. Publications: 3. Development: 11 – present. `https://github.com/allr/fastr`.
2. **JSBench:** JSBench is a new approach to JavaScript benchmarking that automates the process of adapting real-world code with a record-replay framework. Our benchmarks are generated from real JavaScript-utilizing web pages. Publications: 1. Development: 11 – present. Press: `https://www.facebook.com/PurdueCS/posts/10151644300119116`
`http://jsbench.cs.purdue.edu`
3. **Thorn:** Thorn is new concurrent and distributed programming language developed at Purdue and IBM. 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: [109, 110, 54, 48]. Development: 08 – 12. `http://www.thorn-lang.org`
4. **PJAz:** The Purdue JavaScript Analyzer package is the first open source trace-based analysis engine for JavaScript. 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: [43, 110] Development: 09 – 11.
5. **CDx:** The Collision Detector benchmark suite is an open source application benchmark suite that targets different hard and soft real-time virtual machines. Publications: [47, 49, 106, 105] Development: 04 – 12. `http://sss.cs.purdue.edu/projects/cdx/`
6. **Flexotasks:** 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. Publications: [57, 55, 52, 51, 6] Development: 07 – 09.
`http://flexotask.sourceforge.net`
7. **StmBench7:** To date, empiric evaluations of these implementations have suffered from the lack of realistic benchmarks. STMBench7 is a benchmark for evaluating STM implementations. Publications: [59] Development: 07 – 10. `http://lpdserver.epfl.ch/transactions/wiki/doku.php?id=stmbench7`
8. **Ovm:** An open source Java virtual machine framework with support for the Real-time Specification for Java. Publications: [11, 18, 56, 58, 60, 61, 63, 64, 65, 73, 80, 10, 9, 57, 62, 66] Development: 00 – 08. `http://ovmj.net`
9. **MBA:** A tool for *M*odel-*B*ased protein backbone nuclear magnetic resonance *A*ssignments. Publications: [13, 16, 67, 20] Development: 03 – 05. `www.stat.purdue.edu/~ovitek/mba/mba.html`
10. **Kacheck:** A tool for analyzing Java programs for detecting confinement violations. Kacheck has been used to analyze over 100MB of Java code. Publications: [12,24, 25, 83, 87] Development: 00 – 02.
11. **JavaSeal:** A mobile agent middleware system based on Java. Publications: [19, 23, 88, 125, 118, 149, 119, **??**] Development: 96 – 99.
12. **Jazz:** Compression of Java class files. Publication: [89]. Development: 98.

Graduated Students

1. Gregor Richards **PhD**, 14. (U Waterloo)
2. Nadya Ortiz **MSc**, 12. (Apple)
3. Fadi Meawad **MSc**, 13. (Google)
4. Brandon Hill **MSc**, 13. (Oracle Labs)
5. Filip Pizlo **PhD**, "Fragmentation tolerant real-time garbage collection", 12. (Apple)
6. Petr Maj **MSc** 11, (Sony).
7. Jacques Thomas **PhD**, 11. (Amazon)
8. Daniel Tang **MSc**, 11. (Google)
9. Johan Östlund **MSc**, 10. (Uupsala)
10. Jesper H. Spring **PhD** (with Prof. Guerraoui). "Reflexes: Programming Abstractions for Highly Responsive Computing in Java", 08.
11. Rajeev Gopalakrishna **PhD** (with Prof. Spafford). **PhD**. "Metric-driven feedback mechanism for secure software development", 06. (Intel Research Labs).
9. Bogdan Carbunar **PhD**. "Coverage Problems in Wireless Sensor Networks", 05 (Motorola Research)
10. Krzyzstof Palacz **PhD**. "Crusoe–Towards a Multicomputer Execution Environment for Java", 04. (Sun Labs).
11. Jason Baker **MSc**, 07, (Google).
12. Hiroshi Yamauchi **MSc**, 07, (Google).
13. Christian Grothoff **MSc**, 05 (University of Denver).
14. Andrey Madan **MSc**, 04, (Medtronics).
15. Gergana Markova **MSc**. "Analyzing the Visitor Design Pattern", 03 (IBM)
16. Jason M. Fox **MSc**, 03 (JPL)
17. James Liang **MSc**, 02 (Sandia).

Post-doctoral Researchers

1. Gustavo Petri, 12 – 14.
2. Rafal Kolanski, 13 – 14.
3. Michal Malohlava, 12 – 13 (0xdata).
4. Floreal Morandat, 11 – 12 (U Bordeaux)
5. Nicholas Kidd, 09 – 10. (Google)
6. Christian Hammer, 09 – 11. (U Saarland)
7. Ales Plsek, 09 – 11. (Oracle)
8. Sylvain Lebresne, 08 – 09. (yakaz.com)
9. Tomas Kalibera, 07 – 09. (Charles Univ)
10. Tobias Wrigstad, 07 – 09. (U Stockholm)
11. Antonio Cunei, 03 – 08. (EPFL)
12. Jean Privat, 06 – 07, (Univ du Québec)
13. Marek Prochazka, 03 – 05, (SciSys)
14. Jeremy Manson, 03 – 05, (Google)
15. Michael Richmond, 02 – 03, (IBM Research)

Undergraduate students

Brian Burg, 10 (University of Washington). Brett Mravec, Jason Ward, Chris Abernathy, 10. Rob Gevers, 09 (Purdue). Daniel Tang, 08 (Purdue). William Harris, 07. Andrew McClure, 06. Zacchary Wiggins, 05. Paul Kuliniewicz, 04. Wenchang Liu, 04 (Purdue). Filip Pizlo, 04 (Purdue). Chris Willmore, 03. Andrey Madan, 02 (Purdue). Ben Titzer, 03 (UCLA). Adam Lugowski, 02. Josh Moore, 02. Gergana Markova, 01. Theodore Witkamp, 03. Javed Siddique, Alen Montz, 04 (Purdue).

Internships

Scott Carr, Microsoft Research.14 Fadi Meawads, Google.13 Gregor Richards (12), Oracle. Brandon Hill (12), Oracle. Fadi Meawads (12), Google. Gregor Richards (11), Mozilla. Fadi Meawad (10), Microsoft Research. Lei Zhao (10), Oracle. Gregor Richards (10), Microsoft Research. Daniel Tang (10), Google. Fadi Meawad (09), Google. Gregor Richards (09), IBM Research. Armand Navabi (09), Microsoft. Johan Östlund (09), Adobe. Jesper Spring, IBM Research. Filip Pizlo, Microsoft Resarch. Jacques Thomas, Google. Jacques Thomas, Microsoft. Krzystof Palacz, SUN Labs. Adam Welc, SUN Labs. Hiroshi Yamauchi, SUN Labs. Gergana Markova, IBM Research. Filip Pizlo, IBM Research. Christian Grothof, IBM Research. Andrew McClure, SUN Labs. Ben Titzer, SUN Labs. Andrei Madan, Medtronics.

Teaching

While at Purdue, Dr. Vitek taught introductory courses in programming, senior software engineering, embedded and programming languages, as well as graduate programming languages, software engineering, and embedded systems. All of the classes where new offerings at Purdue or significant overhauls of existing classes.

Embedded systems. Embedded Computer Systems is Purdue's first embedded course. This course introduces students to the challenges of real-time programming and introduce Java as a general-purpose language capable of solving hard real-time problems. Low-level programs are written in C. The course covers the following topics: concurrent programming, real-time programming, worst case execution time analysis, schedulability analysis, real-time garbage collection, safety critical system verification and validation, model-driven architectures. Students performed hand on experiments in a lab configured with the RTEMS real-time operating system, LEON radiation-hardened processors, Real-time Java virtual machines, and the Rapita static program analysis tools.

Software engineering. Dr. Vitek overhauled Purdue's SE curriculum for both graduate and undergraduate offerings. The objective of the course is to teach the principles and practical techniques needed to engineer large-scale, reliable, and secure systems. The course introduces students to a typical industrial setting where the work is in small and specialized teams, and where the projects involve composing application from of off-the-shelf components rather than developing the applications from scratch. The course is based on the object-oriented approach which is particularly appropriate for achieving these goals. The material ranges from design and modelling, programming practices, refactoring, testings, design patterns and program analysis.

Programming languages. Dr. Vitek designed this course around the book *"Programming languages, an interpreter based approach"* by Sam Kamin and Norman Ramsey and developed the additional materials and the projects. The course studies programming languages through their operational semantics and language interpreters. It gives both a formal account of the languages, and a practical feeling for implementation concerns.

Introduction to C programming. Redesigning the C programming class was a major endeavor as it entailed scaling the class from 50 students to over 280 as the department's enrollment soared. Dr. Vitek developed software for automated grading, designed several assignment and project sequences, prepared short videos for offline learning, and used technologies such as clickers for rapid classroom feedback. Dr. Vitek gave the class three times and his changes have been adopted by the following instructors.

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, the International Workshop on Aliasing, Confinement and Ownership, the Workshop on Lanugages, Compilers, and Hardware Support for Transactional Computing, and was the first program chair of Virtual Excetion Environments. Together with his colleagues Jagannathan and Grama, he founded and ran the first two instances of the Summer School on Trends in Concurrency which were held in Bertinoro, Prague and Bengalore and attracted over 50 PhD students each time.

International Meetings and Schools Organized

1. NSF DALI Workshop on Dynamic Languages for Scalable Data Analytics, *October 2013*.
2. NSF Workshop on Programming with Big Data, *January 2013*.
3. Dagstuhl Seminar on *Foundations for Scripting Languages*, *February 2012*.
4. Virtual Execution Environments for Scientific Computing NSF Workshop, *September 2010*.
5. Third International Summer School on Trends in Concurrency, Bangalore *June 2010*.
6. IFIP WG 2.4 Working group meeting. Bormio, Italy, 2009
7. Second International Summer School on Trends in Concurrency, Prague *June 2008*.
8. Dagstuhl Seminar on *Types for Tools: Applications of Type Theoretic Techniques June 2005*.
9. First International Summer School on Trends in Concurrency, Bertinoro, *July 2006*.

General Chair

1. Conference on Languages, Programming Language Design and Implementation (**PLDI**), 2012.
2. Conference on Languages, Compilers and Tools for Embedded Systems (**LCTES**), 2011.
3. International Memory Management Symposium (**ISMM**), 2010.
4. Workshop on Lang., Comp., & Hw Support for Transactional Computing (**TRANSACT**), 2006.

Program Chair

1. European Symposium on Programming (**ESOP**) 2015.
2. Conference on Objects, Models, Components, Patterns (**TOOLS Europe**) 2010.
3. Java Technologies for Real-time and Embedded Systems (**JTRes**) symposium 2010.
4. European Conference on Object Oriented Programming (**ECOOP**) 2008.
5. Conference on Coordination Models and Languages (**COORDINATION**) 2007.
6. Conference on Virtual Execution Environments (**VEE**) 2005.
7. Formal Techniques for Java-like Programs (**FTfJP**) workshop 2005.
8. Java Technologies for Real-time and Embedded Systems (**JTRes**) workshop, 2005.

Conference Program Committees

AISB Symposium on Software Mobility and Adaptive Behavior, *2001*.

ASA/MA Agent Systems and Applications, Mobile Agents, *2001*.

APLAS Asian Symposium on Programming Languages and Systems, *2014*.

CATS Computing: The Australasian Theory Symposium, *2010*.

CC International Conference on Compiler Construction, *2003, 2008, 2012, 2014*.

CD Component Deployment, *2002, 2004*.

COORD International Conference on Coordination Models and Languages, *2005, 2008*.

CSF IEEE Computer Security Foundations Symposium, *2008*.

DATE Design, Automation & Test in Europe, Conference, *2010*.

DLS Dynamic Language Symposium Conference, *2010*.

ECOOP European Conference on Object Oriented Programming, *1998, 2000, 2001, 2002, 2003, 2007, 2008, 2009, 2010, 2013*.

ESOP European Symposium on Programming, *2002, 2007, 2009, 2011, 2014*.

EUC IEEE/IFIP International Conference On Embedded and Ubiquitous Computing, *2008, 2010*.

GPCE Conference on Generative Programming: Concepts & Experiences, *2013*

HOTPAR USENIX Hot Topics in Parallelism, *2013*.

ICFP ACM SIGPLAN International Conference on Functional Programming, *2005*.

ICALP International Conference on Automata, Languages and Programming, *2000*.

ISMM International Symposium on Memory Management, *2010*.

ISORC International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, *2012*.

JFLA Journées Francophones des Langages Applicatifs, *1995, 1998, 2000*.

MASS Symposium on Multi-Agent Security and Survivability, *2004, 2005*.

PLDI Conference on Programming Language Design and Implementation, *2001, 2010, 2013*.

PPPJ International conference on Principles and Practice of Programming in Java,*2006*.

PODC Symposium on Principles of Distributed Computing, *2010*.

POPL ACM SIGPLAN Conference on Principles of Programming Languages, *2001, 2007, 2011*.

RTSS IEEE International Real-Time Systems Symposium, *2009, 2010, 2011*.

SACMAT Symposium on Access Control Models and Technologies, *2001*.

TOOLS TOOLS Europe, *2011*.

OOPS Object Oriented Programming Languages and Systems *2004, 2005*.

OOPSLA ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, *2000, 2004, 2007, 2008*.

VEE Virtual Execution Environments, *2011*.

Workshop Program Committees

AIOOL International Workshop on Abstract Interpretation of Object-Oriented Languages, *2005*. **ACP4IS** Workshop on Aspects, Components, and Patterns for Infrastructure Software, *2003, 2004*. **ARRAY** Workshop on Libraries, Languages, and Compilers for Array Programming, *2014*. **Bytecode** Workshop on Bytecode Semantics, Verification Analysis and Transformation, *2007*. **CORD** Workshop on Concurrency, Real-Time and Distribution in Eiffel (CORDIE), *2006*. **CPS** International Workshop on Cyber-Physical Systems, *2008*. **CSJP** Workshop on Concurrency and Synchronization in Java Programs, *2004*. **DOSW**

Distributed Object Security Workshop. *1999*. **FOCLASA** International Workshop on the Foundations of Coordination Languages and Software Architectures, *2004, 2005, 2007*. **FOOL** Workshop on Foundations of Object-Oriented Languages, *2013*. **GCM** Green Computing Middleware, *2010*, *2011, 2012*. **ICOOO** Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems. (ICOOOLPS), *2006, 2013*. **IWACO** International Workshop on Aliasing, Confinement and Ownership, *2003, 2007*, 2014. **IWMSE** Third International Workshop on Multicore Software Engineering, *2010*. **IWAOOS** Intercontinental Workshop on Aliasing in Object-Oriented Systems. *1999*. **JTRes** Workshop on Java Technologies for Real-Time and Embedded Systems, *2003, 2004, 2005, 2006, 2007, 2008, 2009,*, *2010*, *2011*, *2012*. **MOS** Mobile Object Systems Workshop, *1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 2003, 2004, 2005*. **OBT** Off-the Beaten Track, *2014* **PLACES** Programming Language Approaches to Concurrency and Communication-cEntric Software, *2009*, *2010*, *2011*, *2012*. **PLAS** Workshop on Programming Languages Analysis for Security, *2007* **PLASTIC** Workshop on Programming Language And Systems Technologies for Internet Clients, *2011*. **PWD** Workshop on Program Analysis for Security and Safety (PASSWORD), *2006* **SecCo** International Workshop on Security Issues in Coordination Models, Languages and Systems, *2003, 2004, 2005, 2007*. **STOP** Script to Program Evolution, *2009*. **TRUST** Workshop on Reproducible Research Methodologies and New Publication Models, *2014*. **VMIL** Workshop on Virtual Machines and Intermediate Languages, *2009*. **WSIC** Workshop on Secure Internet Computations. Organizer *1999*.

Department and School service

- Graduate admissions Chair: *11 – 14*.
- Graduate admissions: *04 – 05*, *07 – 11*.
- Hiring committee: *09, 14*.
- Colloquium chair: *03, 04*.
- Graduate committee: *99 – 02*.
- Student appeal committee: *04 – 05*, *10 – 12*.

Grants

1. PI *HyDyS: Hybrid Dynamic/Static Analysis for Managed Languages*, **Google**, $60,000, 14.
2. PI *FastR: Open Source R*, **Oracle Inc**, $150,000, 13.
3. PI *Hybrid Dynamic and Static Techniques for Trustworthy Data Analytics*, **ONR**, $600,000, 13 – 16.
4. Co-PI (50%) *Verified Compilation of Concurrent Managed Languages*, **AFRL**, $300,000, 13 – 16.
5. PI *CSR: Language and Runtime Support for Large-Scale Data Analytics*, **NSF**, $246,635, 12.
6. PI *FastR: Open Source R*, **Oracle Inc**, $85,000, 12.
7. PI *Automated Generation of JavaScript Workloads*, **Mozilla Corporation**, $75,000, 12.
8. PI (25%) *CPS: Robust Distributed Wind Power Engineering*, **NSF**, $1,600,000, 11 – 15.
9. PI (50%) *SI2-SSE: A Tracing Virtual Machine For Statistical Computing*, **NSF**, $489,084, 10 – 13.
10. PI *EAGER: Foundations of Data-Centric Concurrency Control*, **NSF**, $110,000, 10 – 11.
11. PI *Virtual Execution Environments for Scientific Computing Workshop*, **NSF**, $45,000, 10 – 10.
12. PI *Third International Summer School on Trends in Concurrency*, **NSF**, $12,000, 10 – 10.
13. Co-PI (50%) *An Infrastructure for Scalable Transactional Memory Abstractions*, **NSF**, $536,000, 10.
14. Co-PI (50%) *SHF: Specification and Verification of Safety Critical Java*, **NSF**, $500,000, 09–11.
15. PI *A Computational Model for High-Assurance Dynamic Systems*, **ONR**, $200,000, 09– 09.
16. PI (50%) *Certified Garbage Collection for Highly Responsive Systems*, **NSF**, $498,952, 08–11.
17. Co-PI (50%) *Fault Determination and Recovery in Cycle Sharing Infrastructures*, **NSF**, $23,000, 08.
18. PI (25%) *Unified Open Source Transactional Infrastructure*, **NSF**, $1,000,000, 08 – 11.
19. Co-PI *Language & Runtime Support for Safe and Scalable Programs*, **Microsoft Research**, $50'000 (25%), 08.
20. PI *Second International Summer School: Trends in Concurrency*, **NSF**, $23,000, 08.
21. PI *Second International Summer School: Trends in Concurrency*, **IBM Research**, $1,000, 08.
22. PI *Second International Summer School: Trends in Concurrency*, **Microsoft Research**, $10,000, 08.
23. PI *Second International Summer School: Trends in Concurrency*, **Intel Research**, $5,000, 07.

24. PI *CSR-EHS: High-throughput Real-time Stream Processing in Java*, **NSF**, $210,000, 07 – 10.
25. Co-PI *Controlled Declassification with Software Transactional Memory*, **NSF**, $249,857 (50%), 07 – 09.
26. PI *IBM Faculty Award*, **IBM**, $30,000, 06.
27. PI *High-level Concurrency Control Abstractions* , **Microsoft Research Award**, $50,000, 06.
28. Co-PI *An Infrastructure for Relaxed Concurrency Abstractions*, **NSF**, $99,979 (20%), 06 – 08.
29. PI *International Summer School: Trends in Concurrency*, **Microsoft Research**, $5,000, 06.
30. PI *International Summer School: Trends in Concurrency*, **IBM Italy**, $5,000, 06.
31. Co-PI *Fault Determination and Recovery in Cycle-Sharing Infrastructures*, **NSF**, $300,000, 05 – 08.
32. PI *Aspectual Configuration of Real-time Embedded Middleware*, **NSF**, $250,000, 05 – 08.
33. Co-PI *A logically destructive imaging security & forensics facility*, **NSF**, $800,000 (7%), 04 – 07.
34. PI *Assured Software Composition For Real-Time Systems*, **NSF/NASA**, $500,000, 03 – 07.
35. PI *Language Abstractions for Parallel Computing*, **DARPA PERCS**, $400,000, 03 – 06.
36. Co-PI *Partage: An Open Peer-to-Peer Infrastructure for Cycle-Sharing*, **NSF**, $498,945, 03 – 06.
37. Co-PI *Distributed Access Control for Accountable Systems*, **NSF Cybertrust**, $318,375, 02 – 06.
38. PI *Foundations & Implementation of Mobile Object Systems*, **NSF CAREER**, $325,936, 01 – 06.
39. PI *Dynamic Compositional Middleware Frameworks for Networked Embedded Systems*, **DARPA**, $3,274,680 (60%), 01 – 05.
40. PI *Sw. Eng.: Research on Customizable Virtual Machines*, **Microsoft Research**, $100,000, 02.
41. PI *Trusted Software Composition*, **Eli Lilly**, $50,000, 01 – 02.
42. PI *ReAssure–Secure and Resilient Network Computing*, **Eli Lilly**, $90,000, 99 – 01.
43. PI *Resilient Mobile Agent Architecture*, **Motorola**, $62,543, 00 – 05.
44. PI *Type confinement in Java*, **Eli Lilly**, $25,000, 99 – 00.