CURRICULUM VITAE

PERSONAL INFORMATION

Peter Müller
Freiestrasse 159
8032 Zurich
Switzerland

Email: peter.mueller@inf.ethz.ch
Phone: ++41 44 63 22868
Date of Birth: January 28, 1972
Married, one daughter

EDUCATION

03/2003	PMI certification as Project Management Professional
04/2001	Dr. rer. nat. in Computer Science (summa cum laude) Fernuniversität Hagen, Germany Thesis: Modular Specification and Verification of Object-Oriented Programs
08/1998	International Marktoberdorf Summer School on Calculational System Design
11/1991 – 05/1996	Study of Computer Science (Diplom with distinction) Technische Universität München, Germany
08/1982 – 08/1991	Apian-Gymnasium Ingolstadt, Germany (Abitur)

EMPLOYMENT

Since 08/2008	Full Professor, ETH Zurich, Switzerland Programming Methodology Group
01/2024 – 03/2024	Visiting Professor, National University of Singapore
05/2013 – 07/2013	Visiting Professor, University of Washington, USA
02/2013 – 05/2013	Visiting Researcher, Microsoft Research, USA
07/2011	Visiting Researcher, Microsoft Research, USA
07/2009 – 08/2009	Research Consultant, Microsoft Research, USA
06/2007 – 07/2008	Researcher, Microsoft Research, USA
06/2003 – 07/2008	Assistant Professor, ETH Zurich, Switzerland Head of Software Component Technology Group on leave of absence from 06/2007 to 07/2008
02/2006 – 03/2006	Research Consultant, Microsoft Research, USA
06/2001 – 05/2003	Project Manager, Deutsche Bank AG, Frankfurt, Germany
04/1997 – 05/2001	Research Assistant, Fernuniversität Hagen, Germany
06/1998 – 08/1998	Visiting Scientist, Iowa State University, USA
06/1996 – 03/1997	Research Assistant, Technische Universität München, Germany
1992 – 1994	Three Internships as Programmer Siemens Nixdorf Informationssysteme AG, Germany

PUBLICATIONS

BOOKS AND MONOGRAPHS

[1] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig: *The Complete Guide to SCION*. Springer-Verlag, 2022. 656 pages.

[2] P. Müller. *Modular Specification and Verification of Object-Oriented Programs*, volume 2262 of LNCS. Springer-Verlag, 2002. 302 pages.

[3] P. Müller. *Modular Specification and Verification of Object-Oriented Programs*. PhD thesis, Fernuniversität Hagen, 2001. 275 pages.

JOURNAL PUBLICATIONS AND BOOK CHAPTERS

[1] B. Dongol, C., S., E. Hehner, C. Morgan, P. Müller, L. Ribeiro, A. Silva, G. Smith, and E. de Vink: On Formal Methods Thinking in Computer Science Education. *Formal Aspects of Computing*, ACM, 2024.

[2] F. A. Wolf, M. Schwerhoff, and P. Müller: Concise Outlines for a Complex Logic: A Proof Outline Checker for TaDA. *Formal Methods in System Design*, 2023.

[3] A. Bugariu, A. Ter-Gabrielyan, and P. Müller: Identifying Overly Restrictive Matching Patterns in SMT-based Program Verifiers (extended version). *Formal Aspects of Computing*, 2022.

[4] P. Müller and N. Shankar: The First Fifteen Years of the Verified Software Project. In *Theories of Programming: The Life and Works of Tony Hoare*, ACM, pages 93-123, 2021.

[5] C. Dross, C. A. Furia, M. Huisman, R. Monahan and P. Müller: VerifyThis 2019: a program verification competition. In *International Journal on Software Tools for Technology Transfer (STTT)*, 2021.

[6] A. J. Summers and P. Müller: Automating Deductive Verification for Weak-Memory Programs (extended version). *International Journal on Software Tools for Technology Transfer (STTT)*, 2020.

[7] P. Müller. Building Deductive Program Verifiers – Lecture Notes. In *Engineering Secure and Dependable Software Systems,* IOS Press, pages 189-206, 2019.

[8] M. Eilers, P. Müller, and S. Hitz. Modular Product Programs. *ACM Transactions on Programming Languages and Systems*, 41(1), 2019.

[9] P. Müller. The Binomial Heap Verification Challenge in Viper. In *Principled Software Development*, Springer-Verlag, 2018.

[10] W. Dietl and P. Müller. Object Ownership in Program Verification. In *Aliasing in Object-Oriented Programming*, volume 7850 of LNCS, pages 289-318, Springer-Verlag, 2013.

[11] J. Hatcliff and G.T. Leavens and K.R.M. Leino and P. Müller and M. Parkinson. Behavioral Interface Specification Languages, *Computing Surveys*, 2012.

[12] W. Dietl, S. Drossopoulou, and P. Müller. Separating Ownership Topology and Encapsulation with Generic Universe Types, *ACM Transactions on Programming Languages and Systems*, 33(6), 2011.

[13] M. Barnett and M. Fähndrich and K. R. M. Leino and P. Müller and W. Schulte and H. Venter. Specification and Verification: The Spec# Experience, *Communications of the ACM*, 54(6), 81-91, 2011.

[14] A.J. Summers, S. Drossopoulou, and P. Müller. Universe-Type-Based Verification Techniques for Mutable Static Fields and Methods, *Journal of Object Technology (JOT)*, 8(4), 2009.

[15] Á. Darvas and P. Müller. Faithful mapping of model classes to mathematical structures. *IET Software,* 2(6):477-499, 2008.

[16] G.T. Leavens, K.R.M. Leino, and P. Müller. Specification and verification challenges for sequential object-oriented programs*. Formal Aspects of Computing*, 19(2):159-189, 2007.

[17] P. Müller, A. Poetzsch-Heffter, and G.T. Leavens. Modular invariants for layered object structures. *Science of Computer Programming*, 62(3):253-286, 2006.

[18] Á. Darvas and P. Müller. Reasoning about method calls in interface specifications. *Journal of Object Technology (JOT)*, 5(5):59-85, 2006.

[19] W. Dietl and P. Müller. Universes: Lightweight ownership for JML. *Journal of Object Technology (JOT)*, 4(8):5-32, 2005.

[20] P. Müller, A. Poetzsch-Heffter, and G.T. Leavens. Modular specification of frame properties in JML. Concurrency and Computation: Practice and Experience, 15:117-154, 2003.

[21] P. Müller and A. Poetzsch-Heffter. Modular specification and verification techniques for object-oriented software components. In G.T. Leavens and M. Sitaraman, editors, *Foundations of Component-Based Systems,* pages 137-159, Cambridge University Press, 2000.

CONFERENCE PUBLICATIONS

[22] J. Pereira, T. Klenze, S. Giampietro, M. Limbeck, D. Spiliopoulos, F. Wolf, M. Eilers, C. Sprenger, D. Basin, P. Müller, A. Perrig: Protocols to Code: Formal Verification of a Secure Next-Generation Internet Router. *Computer and Communications Security (CCS)*, 2025.

[23] A. Bílý, J. Pereira, and P. Müller: A Refinement Methodology for Distributed Programs in Rust. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, Proc. ACM Program. Lang., 2025.

[24] Z. Grannan, A. Bílý, J. Fiala, J. Geer, M. de Medeiros, P. Müller, and A. Summers: Place Capability Graphs: A General-Purpose Model of Rust’s Ownership and Borrowing Guarantees. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, Proc. ACM Program. Lang., 2025.

[25] J. Pereira, I. van Bakel, P. Firlejczyk, M. Eilers, and P. Müller: Modular Reasoning about Global State. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, Proc. ACM Program. Lang., 2025.

[26] K. Gopinathan, D. Spiliopoulos, V. Goyal, P. Müller, M. Püschel, and I. Sergey: Accelerating Automated Program Verifiers by Automatic Proof Localization. *Computer Aided Verification (CAV)*, LNCS, Springer-Verlag, 2025.

[27] M. Eilers, M. Schwerhoff, A. Summers, and P. Müller: Fifteen Years of Viper. *Computer Aided Verification (CAV)*, LNCS, Springer-Verlag, 2025.

[28] S. Spies, N. Mück, H. Zeng, M. Sammler, A. Lattuada, P. Müller, and D. Dreyer: Destabilizing Iris. *Programming Language Design and Implementation (PLDI)*, Proc. ACM Program. Lang., pages 848-873, 2025. Distinguished Paper Award.

[29] Thibault Dardinier, Michael Sammler, Gaurav Parthasarathy, Alexander J. Summers, and Peter Müller: Formal Foundations for Translational Separation Logic Verifiers. *Principles of Programming Languages (POPL)*, Proc. ACM Program. Lang., 2025.

[30] F. A. Wolf and P. Müller: Verifiable Security Policies for Distributed Systems. *Computer and Communications Security (CCS)*, 2024.

[31] T. Dardinier, A. Li, and P. Müller: Hypra: A Deductive Program Verifier for Hyper Hoare Logic. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, Proc. ACM Program. Lang., pages 1279-1308, 2024.

[32] M. Eilers, M. Schwerhoff, and P. Müller: Verification Algorithms for Automated Separation Logic Verifiers. *Computer Aided Verification (CAV)*, volume 14681 of LNCS, pages 362-386, Springer-Verlag, 2024.

[33] T. Dardinier and P. Müller: Hyper Hoare Logic: (Dis-)Proving Program Hyperproperties. *Programming Language Design and Implementation (PLDI), Proc. ACM Program. Lang.*, pages 1485-1509, 2024.

[34] G. Parthasarathy, T. Dardinier, B. Bonneau, P. Müller, and A. J. Summers: Towards Trustworthy Automated Program Verifiers: Formally Validating Translations into an Intermediate Verification Language. *Programming Language Design and Implementation (PLDI), Proc. ACM Program. Lang.*, pages 1510-1534, 2024.

[35] L. Arquint, M. Schwerhoff, V. Mehta, and P. Müller: A Generic Methodology for the Modular Verification of Security Protocol Implementations. *Computer and Communications Security (CCS)*, ACM, 2023.

[36] T. Dardinier, G. Parthasarathy, and P. Müller: Verification-Preserving Inlining in Automatic Separation Logic Verifiers. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), Proc. ACM Program. Lang.,* pages 789-818, 2023.

[37] J. Fiala, S. Itzhaky, P. Müller, N. Polikarpova, and I. Sergey: Leveraging Rust Types for Program Synthesis. *Programming Language Design and Implementation (PLDI), Proc. ACM Program. Lang.,* pages 1414-1437, 2023.

[38] M. Eilers, T. Dardinier, and P. Müller: CommCSL: Proving Information Flow Security for Concurrent Programs using Abstract Commutativity. *Programming Language Design and Implementation (PLDI), Proc. ACM Program. Lang.,* pages 1682-1707, 2023.

[39] L. Arquint, F. A. Wolf, J. Lallemand, R. Sasse, C. Sprenger, S. N. Wiesner, D. Basin, P. Müller: Sound verification of security protocols: From design to interoperable implementations. *Security and Privacy (SP)*, pages 1077-1093, IEEE Computer Society, 2023.

[40] T. Dardinier, P. Müller, and A. J. Summers: Fractional Resources in Unbounded Separation Logic. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), Proc. ACM Program. Lang.,* pages 1066-1092, 2022. Distinguished Paper Award.

[41] T. Dardinier, G. Parthasarathy, N. Weeks, A. J. Summers, and P. Müller: Sound Automation of Magic Wands. *Computer Aided Verification (CAV)*, volume 13372 of LNCS, pages 130-151, Springer-Verlag, 2022.

[42] F. A. Wolf, M. Schwerhoff, and P. Müller: Concise Outlines for a Complex Logic: A Proof Outline Checker for TaDA. *Formal Methods (FM)*, volume 13047 of LNCS, pages 407-426, Springer-Verlag, 2021.

[43] A. Bugariu, A. Ter-Gabrielyan, and P. Müller: Identifying Overly Restrictive Matching Patterns in SMT-based Program Verifiers. *Formal Methods (FM)*, volume 13047 of LNCS, pages 273-291, Springer-Verlag, 2021.

[44] F. Wolff, A. Bílý, C. Matheja, P. Müller, and A. J. Summers: Modular Specification and Verification of Closures in Rust. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), Proc. ACM Program. Lang.,* 2021.

[45] C. Bräm, M. Eilers, P. Müller, R. Sierra, and A. J. Summers: Rich Specifications for Ethereum Smart Contract Verification. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), Proc. ACM Program. Lang.,* 2021.

[46] M. Eilers, S. Meier, and P. Müller: Product Programs in the Wild: Retrofitting Program Verifiers to Check Information Flow Security. *Computer Aided Verification (CAV)*, volume 12759 of LNCS, pages 718-741, Springer-Verlag, 2021.

[47] G. Parthasarathy, P. Müller, and A. J. Summers: Formally Validating a Practical Verification Condition Generator. *Computer Aided Verification (CAV)*, volume 12760 of LNCS, pages 704-727, Springer-Verlag, 2021.

[48] F. A. Wolf, L. Arquint, M. Clochard, W. Oortwijn, J. C. Pereira, and P. Müller: Gobra: Modular Specification and Verification of Go Programs. *Computer Aided Verification (CAV)*, volume 12759 of LNCS, pages 367-379, Springer-Verlag, 2021.

[49] V. Astrauskas, C. Matheja, P. Müller, F. Poli, and A. J. Summers: How Do Programmers Use Unsafe Rust? *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 136:1-136:27, ACM, 2020.

[50] C. Sprenger, T. Klenze, M. Eilers, F. A. Wolf, P. Müller, M. Clochard, and D. Basin: Igloo: Soundly Linking Compositional Refinement and Separation Logic for Distributed System Verification. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 152:1-152:31, ACM, 2020.

[51] A. Bugariu and P. Müller: Automatically Testing String Solvers. *International Conference on Software Engineering (ICSE)*, pages 1459-1470, ACM, 2020.

[52] V. Astrauskas, P. Müller, F. Poli, and A.J. Summers: Leveraging Rust Types for Modular Specification and Verification. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 147:1-147:30, ACM, 2019.

[53] A. Ter-Gabrielyan P. Müller, and A.J. Summers: Modular Verification of Heap Reachability Properties in Separation Logic. *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 121:1-121:28, ACM, 2019.

[54] N. Becker, P. Müller, and A.J. Summers: The Axiom Profiler: Understanding and Debugging SMT Quantifier Instantiations. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS),* volume 11427 of LNCS, pages 99-116, Springer-Verlag, 2019.

[55] A. Bugariu, M. Christakis, P. Müller, and V. Wüstholz. Automatic Testing of Implementations of Numerical Abstract Domains. *Automated Software Engineering (ASE)*, page 768-778. ACM, 2018.

[56] C. Urban, S. Ueltschi, and P. Müller: Abstract Interpretation of CTL Properties. *Static Analysis Symposium (SAS)*, volume 11002 of LNCS, pages 402-422, Springer-Verlag, 2018.

[57] J. Dohrau, A. J. Summers, C. Urban, S. Münger, and P. Müller. Permission Inference for Array Programs. *Computer Aided Verification (CAV)*, volume 10982 of LNCS, pages 55-74, Springer-Verlag, 2018.

[58] M. Eilers and P. Müller. Nagini: A Static Verifier for Python. *Computer Aided Verification (CAV)*, volume 10982 of LNCS, pages 596-603, Springer-Verlag, 2018.

[59] M. Hassan, C. Urban, M. Eilers, and P. Müller. MaxSMT-Based Type Inference for Python 3, *Computer Aided Verification (CAV)*, volume 10982 of LNCS, pages 12-19, Springer-Verlag, 2018.

[60] L. Brutschy, D. Dimitrov, P. Müller, and M. Vechev. Static Serializability Analysis for Causal Consistency. *Programming Language Design and Implementation (PLDI),* pages 90-104, ACM, 2018.

[61] A. J. Summers and P. Müller. Automating Deductive Verification for Weak-Memory Programs. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 10805 of LNCS, pages 190-209, Springer-Verlag, 2018.

[62] M. Eilers, P. Müller, and S. Hitz. Modular Product Programs. *European Symposium on Programming (ESOP)*, volume 10801 of LNCS, pages 502-529, Springer-Verlag, 2018.

[63] C. Urban and P. Müller. An Abstract Interpretation Framework for Input Data Usage. *European Symposium on Programming (ESOP)*, volume 10801 of LNCS, pages 683-710, Springer-Verlag, 2018.

[64] M. Christakis, P. Emmisberger, P. Godefroid, and P. Müller. A General Framework for Dynamic Stub Injection. *International Conference on Software Engineering (ICSE)*, pages 586-596, IEEE, 2017.

[65] D. Asenov, B. Guenat, P. Müller, and M. Otth. Precise Version Control of Trees with Line-based Version Control Systems. *Fundamental Approaches to Software Engineering (FASE)*, volume 10202 of *LNCS,* pages 152-169, Springer Verlag, 2017.

[66] L. Brutschy, D. Dimitrov, P. Müller, and M. Vechev. Serializability for Eventual Consistency: Criterion, Analysis, and Applications. *Principles of Programming Languages (POPL)*, pages 458-472, ACM, 2017.

[67] D. Asenov, P. Müller, and L. Vogel. The IDE as a Scriptable Information System. *Automated Software Engineering (ASE)*, pages 444-449, ACM, 2016.

[68] P. Müller, M. Schwerhoff, and A.J. Summers. Automatic Verification of Iterated Separating Conjunctions using Symbolic Execution. *Computer Aided Verification (CAV),* volume 9779 of *LNCS*, pages 405-425, Springer Verlag, 2016.

[69] M. Christakis, P. Müller, and V. Wüstholz. Guiding Dynamic Symbolic Execution toward Unverified Program Executions. *International Conference on Software Engineering (ICSE)*, pages 144-155, ACM, 2016. Distinguished Paper Award.

[70] D. Asenov, O. Hilliges, and P. Müller. The Effect of Richer Visualizations on Code Comprehension. *Human Factors in Computing Systems (CHI)*, pages 5040-5045, 2016.

[71] A.J. Summers and P. Müller. Actor Services: Modular Verification of Message Passing Programs. In P. Thiemann, editor, *European Symposium on Programming (ESOP)*, volume 9632 of *LNCS*, pages 699-726, Springer Verlag, 2016.

[72] M. Christakis and K.R.M. Leino and P. Müller and V. Wüstholz. Integrated Environment for Diagnosing Verification Errors. In M. Chechik and J.-F. Raskin, editors, *Tools and Algorithms for the Construction of Analysis of Systems (TACAS)*, volume 9636 of *LNCS,* pages 424-441, Springer-Verlag, 2016.

[73] C. Favre and H. Völzer and P. Müller. Diagnostic Information for Control-Flow Analysis of Workflow Graphs (aka Free-Choice Workflow Nets). In M. Chechik and J.-F. Raskin, editors, *Tools and Algorithms for the Construction of Analysis of Systems (TACAS),* volume 9636 of *LNCS*, pages 463-479, Springer-Verlag, 2016.

[74] P. Boström and P. Müller. Modular Verification of Finite Blocking in Non-terminating Programs. In J.T. Boyland, editor, *European Conference on Object-Oriented Programming (ECOOP),* pages 639-663, Schloss Dagstuhl, 2015.

[75] M. Christakis, P. Müller, and V. Wüstholz. An Experimental Evaluation of Deliberate Unsoundness in a Static Program Analyzer. In D. D'Souza, A. Lal, and K. Guldstrand Larsen, editors, *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 8931 of *LNCS*, pages 336-354, Springer-Verlag, 2015.

[76] P. Ferrara, P. Müller, and M. Novacek. Automatic Inference of Heap Properties Exploiting Value Domains. In D. D'Souza, A. Lal, and K. Guldstrand Larsen, editors, *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 8931 of *LNCS*, pages 393-411, Springer-Verlag, 2015.

[77] L. Brutschy, P. Ferrara, and P. Müller. Static Analysis for Independent App Developers. In A.P. Black and T.D. Millstein, editors, *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 847-860, ACM, 2014.

[78] M. Christakis, P. Emmisberger, and P. Müller. Dynamic Test Generation with Static Fields and Initializers. In B. Bonakdarpour and S.A. Smolka, editors, *Runtime Verification (RV)*, volume 8734 of *LNCS,* pages 269-284, Springer-Verlag, 2014.

[79] M. Christakis, P. Müller, and V. Wüstholz. Synthesizing Parameterized Unit Tests to Detect Object Invariant Violations. In D. Giannakopoulou and G. Salaün, editors, *Software Engineering and Formal Methods (SEFM)*, volume 8702 of *LNCS*, pages 65-80, Springer-Verlag, 2014.

[80] D. Asenov and P. Müller. Envision: A Fast and Flexible Visual Code Editor with Fluid Interactions (Overview), *Visual Languages and Human-Centric Computing (VL/HCC)*, pages 9-12, 2014

[81] D. Asenov and P. Müller. Customizing the Visualization and Interaction for Embedded Domain-Specific Languages in a Structured Editor, *Visual Languages and Human-Centric Computing (VL/HCC)*, pages 127-130, 2013.

[82] S. Heule, I.T. Kassios, P. Müller, and A.J. Summers. Verification Condition Generation for Permission Logics with Abstract Predicates and Abstraction Functions, *European Conference on Object-Oriented Programming (ECOOP)*, volume 7920 of *LNCS*, pages 451-476, Springer-Verlag, 2013.

[83] S. Heule, K.R.M. Leino, P. Müller, and A.J. Summers. Symbolic Read Permissions: Fractional Permissions without the Fractions, *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 7737 of *LNCS*, pages 315-334, Springer-Verlag, 2013.

[84] M. Christakis, P. Müller, and V. Wüstholz. Collaborative Verification and Testing with Explicit Assumptions*, Formal Methods (FM)*, volume 7436 of *LNCS*, pages 132-146, Springer-Verlag, 2012.

[85] I.T. Kassios, P. Müller, and M. Schwerhoff. Comparing Verification Condition Generation with Symbolic Execution: an Experience Report*, Verified Software Theories Tools Experiments (VSTTE),* volume 7152 of *LNCS*, pages 196-208, Springer-Verlag, 2012.

[86] P. Ferrara and P. Müller. Automatic inference of access permissions, *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 7148 of *LNCS*, pages 202-218, Springer-Verlag, 2012.

[87] A.J. Summers and P. Müller. Freedom Before Commitment – A Lightweight Type System for Object Initialisation, *Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, ACM, 2011.

[88] W. Dietl, M. Ernst, and P. Müller. Tunable Static Inference for Generic Universe Types. In M. Mezini, editor, *European Conference on Object-Oriented Programming (ECOOP)*, volume 6813 of *LNCS*, pages 333-357, Springer-Verlag, 2011. Best Paper Award.

[89] V. Klebanov, P. Müller, N. Shankar, G. T. Leavens, V. Wüstholz, E. Alkassar, R. Arthan, D. Bronish, R. Chapman, E. Cohen, M. Hillebrand, B. Jacobs, K. R. M. Leino, R. Monahan, F. Piessens, N. Polikarpova, T. Ridge, J. Smans, S. Tobies, T. Tuerk, M. Ulbrich, and B. Weiss. The 1st Verified Software Competition: Experience Report. In M. Butler and W. Schulte, editors, *Formal Methods (FM)*, volume 6664 of *LNCS*, pages 154-168, Springer-Verlag, 2011. Best Paper Award.

[90] P. Müller and J. N. Ruskiewicz. Using Debuggers to Understand Failed Verification Attempts. In M. Butler and W. Schulte, editors, *Formal Methods (FM)*, volume 6664 of *LNCS*, pages 73-87, Springer-Verlag, 2011.

[91] M. Nordio, C. Calcagno, B. Meyer, P. Müller, and J. Tschannen. Reasoning about Function Objects. In J. Vitek, editor, *TOOLS-EUROPE,* volume 6141 of *LNCS*, pages 79-76, Springer-Verlag, 2010.

[92] K.R.M. Leino and P. Müller and J. Smans. Deadlock-free Channels and Locks. In A.D. Gordon, editor, *European Symposium on Programming (ESOP)*, volume 6012 of *LNCS*, pages 407-426, Springer-Verlag, 2010.

[93] H. Lehner and P. Müller. Efficient Runtime Assertion Checking of Assignable Clauses with Datagroups. In D. Rosenblum and G. Taentzer, editors, *Fundamental Approaches to Software Engineering (FASE)*, volume 6013 of *LNCS*, pages 338-352, Springer-Verlag, 2010.

[94] Á. Darvas and P. Müller. Proving Consistency and Completeness of Model Classes Using Theory Interpretation. In D. Rosenblum and G. Taentzer, editors, *Fundamental Approaches to Software Engineering (FASE)*, volume 6013 of *LNCS*, pages 218-232, Springer-Verlag, 2010.

[95] M. Nordio, C. Calcagno, P. Müller, and B. Meyer. A Sound and Complete Program Logic for Eiffel. In M. Oriol and B. Meyer, editors, *TOOLS-EUROPE*, volume 33, *Lecture Notes in Business and Information Processing*, pages 195-214. Springer-Verlag, 2009.

[96] K.R.M. Leino and P. Müller. A basis for verifying multi-threaded programs. In G. Castagna, editor, *Programming Languages and Systems (ESOP)*, volume 5502 of *LNCS*, pages 378-393. Springer-Verlag, 2009.

[97] P. Müller and Joseph N. Ruskiewicz. A modular verification methodology for C# delegates. In U. Glässer and J.-R. Abrial, editors, *Rigorous Methods for Software Construction and Analysis, LNCS*, pages 187-203. Springer-Verlag, 2009.

[98] D. Cunningham, W. Dietl, S. Drossopoulou, A. Francalanza, P. Müller, and A.J. Summers. Universe types for topology and encapsulation. In F.S. de Boer, M.M. Bonsangue, S. Graf, and W.-P. de Roever, editors, *Formal Methods for Components and Objects (FMCO)*, volume 5382 of *LNCS*, pages 72-112. Springer-Verlag, 2008.

[99] K.R.M. Leino, P. Müller, and A. Wallenburg. Flexible immutability with frozen objects. In N. Shankar and J. Woodcock, editors, *Verified Software: Theories, Tools, and Experiments (VSTTE)*, volume 5295 of *LNCS*, pages 192-208. Springer-Verlag, 2008.

[100] S. Drossopoulou, A. Francalanza, P. Müller, and A.J. Summers. A unified framework for verification techniques for object invariants. In J. Vitek, editor, *European Conference on Object-Oriented Programming (ECOOP)*, volume 5142 of *LNCS*, pages 412-437. Springer-Verlag, 2008.

[101] M. Nordio, P. Müller, and B. Meyer. Proof-Transforming Compilation of Eiffel Programs. In R. Paige and B. Meyer, editors, *TOOLS-EUROPE*, volume11 of *Lecture Notes in Business and Information Processing*, pages 316-335. Springer-Verlag, 2008.

[102] A.Rudich, Á. Darvas, and P. Müller. Checking well-formedness of pure-method specifications. In J. Cuellar and T. Maibaum, editors, *Formal Methods (FM)*, volume 5014 of *LNCS*, pages 68-83. Springer-Verlag, 2008.

[103] K.R.M. Leino and P. Müller. Verification of equivalent-results methods. In S. Drossopoulou, editor, *Programming Languages and Systems (ESOP)*, volume 4960 of *LNCS*, pages 307-321. Springer-Verlag, 2008.

[104] P. Müller and A. Rudich. Ownership transfer in Universe Types. In *Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 461-478. ACM, 2007.

[105] B. Jacobs, P. Müller, and F. Piessens. Sound reasoning about unchecked exceptions. In M. Hinchey and T. Margaria, editors, *Software Engineering and Formal Methods (SEFM)*, pages 113-122. IEEE, 2007.

[106] W. Dietl, S. Drossopoulou, and P. Müller. Generic Universe Types. In E. Ernst, editor, *European Conference on Object-Oriented Programming (ECOOP)*, volume 4609 of *LNCS*, pages 28-53. Springer-Verlag, 2007.

[107] G.T. Leavens and P. Müller. Information hiding and visibility in interface specifications. In *International Conference on Software Engineering (ICSE)*, pages 385-395. IEEE, 2007.

[108] P. Müller. Reasoning about object structures using ownership. In B. Meyer and J. Woodcock, editors, *Verified Software: Theories, Tools, Experiments (VSTTE 2005)*, volume 4171 of *LNCS*. Springer-Verlag, 2008.

[109] F. Bannwart and P. Müller. Changing programs correctly: Refactoring with specifications. In J. Misra, T. Nipkow, and E. Sekerinski, editors, *Formal Methods (FM)*, volume 4085 of *LNCS*, pages 492-507. Springer-Verlag, 2006.

[110] D. Grolimund and P. Müller. A pattern language for overlay networks in peer-to-peer systems. In U. Zdun, editor, *European Conference on Pattern Languages of Programs (EuroPLoP)*. Universitätsverlag Konstanz, 2007. 48 pages.

[111] K.R.M. Leino and P. Müller. A verification methodology for model fields. In P. Sestoft, editor, *European Symposium on Programming (ESOP)*, volume 3924 of *LNCS*, pages 115-130. Springer-Verlag, 2006.

[112] K.R.M. Leino and P. Müller. Modular verification of static class invariants. In J.Fitzgerald, I.Hayes, and A.Tarlecki, editors, *Formal Methods (FM)*, volume 3582 of *LNCS*, pages 26-42. Springer-Verlag, 2005.

[113] W. Dietl, P. Müller, and A. Poetzsch-Heffter. A type system for checking applet isolation in Java Card. In G. Barthe, L. Burdy, M. Huisman, J.-L. Lanet, and T. Muntean, editors, *Construction and Analysis of Safe, Secure and Interoperable Smart devices (CASSIS)*, volume 3362 of *LNCS*, pages 129-150. Springer-Verlag, 2004.

[114] K.R.M. Leino and P. Müller. Object invariants in dynamic contexts. In M.Odersky, editor, *European Conference on Object-Oriented Programming (ECOOP)*, volume 3086 of *LNCS*, pages 491-516. Springer-Verlag, 2004.

[115] A. Poetzsch-Heffter and P. Müller. A programming logic for sequential Java. In S.D. Swierstra, editor, *European Symposium on Programming Languages and Systems (ESOP)*, volume 1576 of *LNCS*, pages 162-176. Springer-Verlag, 1999.

[116] P. Müller, J. Meyer, and A. Poetzsch-Heffter. Making executable interface specifications more expressive. In C.H. Cap, editor, *Java-Informations-Tage (JIT)*, Informatik Aktuell, pages 243-252. Springer-Verlag, 1999.

[117] A. Poetzsch-Heffter and P. Müller. Logical foundations for typed object-oriented languages. In D.Gries and W. De Roever, editors, *Programming Concepts and Methods (PROCOMET)*, pages 404-423, 1998.

[118] P. Müller and A. Poetzsch-Heffter. Kapselung und Methodenbindung: Javas Designprobleme und ihre Korrektur. In C.H. Cap, editor, *Java-Informations-Tage (JIT)*, Informatik Aktuell, pages 1-10. Springer-Verlag, 1998. (In German).

WORKSHOP PUBLICATIONS

[119] A. Bílý, J. Hansen, P. Müller, and A. J. Summers: Compositional Reasoning about Advanced Iterator Patterns in Rust. *International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO)*, 2023.

[120] L. Brutschy, P. Ferrara, and P. Müller. TouchGuru: Integrating Static Analysis with a Mobile Development Environment. In *Mobile Development Lifecycle*, 2014.

[121] J. Boyland, P. Müller, M. Schwerhoff, and A. J. Summers. Constraint Semantics for Abstract Read Permissions. In *Formal Techniques for Java-like Programs (FTfJP)*, 2014.

[122] S. Balzer, T. Gross, and P. Müller. Selective Ownership: Combining Object and Type Hierarchies for Flexible Sharing. In *Foundations of Object-Oriented Languages (FOOL)*, 2012.

[123] S. Heule, K. R. M. Leino, P. Müller, and A. J. Summers. Fractional Permissions without the Fractions. In *Formal Techniques for Java-like Programs (FTfJP)*, 2011.

[124] A.J. Summers, S. Drossopoulou, and P. Müller. The Need for Flexible Object Invariants. In *International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO)*, 2009.

[125] A.J. Summers, S. Drossopoulou, and P. Müller. A Universe-Type-Based Verification Technique for Mutable Static Fields and Methods. In *Formal Techniques for Java-like Programs*, 2008.

[126] S. Drossopoulou, A. Francalanza, and P. Müller. A Unified Framework for Verification Techniques for Object Invariants. In *Foundations of Object-Oriented Languages (FOOL)*, 2008.

[127] W. Dietl and P. Müller. Ownership Type Systems and Dependent Classes. In *Foundations of Object-Oriented Languages (FOOL)*, 2008.

[128] Á. Darvas and P. Müller. Faithful mapping of model classes to mathematical structures. In *Specification and Verification of Component-Based Systems (SAVCBS)*, pages 31-38, 2007.

[129] P. Müller and M. Nordio. Proof-transforming compilation of programs with abrupt termination. In *Specification and Verification of Component-Based Systems (SAVCBS)*, pages 39-46, 2007.

[130] W. Dietl and P. Müller. Runtime Universe Type Inference. In *International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO)*, 2007.

[131] W. Dietl and P. Müller. 2007 State of the Universe Address. In *International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO)*, 2007.

[132] H. Lehner and P. Müller. Formal Translation of Bytecode into BoogiePL. In M. Huisman and F. Spoto, editors, *Bytecode Semantics, Verification, Analysis and Transformation (BYTECODE)*, volume 190(1) of *Electronic Notes in Theoretical Computer Science*, pages 35-50, 2007.

[133] W. Dietl, S. Drossopoulou, and P. Müller. Generic Universe Types. In J. Reppy, editor, *Foundations and Developments of Object-Oriented Languages (FOOL/WOOD),* 2007. 13 pages.

[134] Á. Darvas and P. Müller. Reasoning about method calls in JML Specifications. In *Formal Techniques for Java-like Programs,* 2005. 12 pages.

[135] F.Y. Bannwart and P. Müller. A logic for bytecode. In F. Spoto, editor, *Bytecode Semantics, Verification, Analysis and Transformation (BYTECODE),* volume 141 of *Electronic Notes in Theoretical Computer Science*, pages 255-273. Elsevier, 2005.

[136] W. Dietl and P. Müller. Exceptions in ownership type systems. In E. Poll, editor, *Formal Techniques for Java-like Programs,* pages 49-54, 2004.

[137] P. Müller and A. Poetzsch-Heffter. A type system for checking applet isolation in Java Card. In *Formal Techniques for Java Programs,* 2001. 9 pages.

[138] P. Müller and A. Poetzsch-Heffter. A type system for controlling representation exposure in Java. In S. Drossopoulou, S. Eisenbach, B. Jacobs, G.T. Leavens, P. Müller, and A. Poetzsch-Heffter, editors, *Formal Techniques for Java Programs*. Technical Report 269, Fernuniversität Hagen, 2000. 8 pages.

[139] P. Müller and A. Poetzsch-Heffter. Universes: A type system for controlling representation exposure. In A. Poetzsch-Heffter and J. Meyer, editors, *Programming Languages and Fundamentals of Programming*, pages 131-140. Fernuniversität Hagen, 1999. Technical Report 263.

[140] P. Müller and A. Poetzsch-Heffter. Alias control is crucial for modular verification. In A. Moreira and D. Demeyer, editors, *Object-Oriented Technology. ECOOP'99 Workshop Reader*, volume 1743 of *LNCS*, pages 154-156. Springer-Verlag, 1999.

[141] P. Müller and A. Poetzsch-Heffter. Developing provably correct programs from object-oriented components. In G.T. Leavens and M. Sitaraman, editors, *Foundations of Component-Based Systems*, pages 188-197, 1997.

[142] A. Poetzsch-Heffter and P. Müller. A logic for the verification of object-oriented programs. In R. Berghammer and F. Simon, editors, *Programming Languages and Fundamentals of Programming*, pages 31-42. Christian-Albrechts-Universität Kiel, 1997. Technical Report 9717.

[143] P. Müller and A. Poetzsch-Heffter. Preserving the correctness of object-oriented programs under extension. In R. Berghammer and F. Simon, editors, *Programming Languages and Fundamentals of Programming*, pages 43-52. Christian-Albrechts-Universität Kiel, 1997. Technical Report 9717.

[144] P. Müller and A. Poetzsch-Heffter. Formal specification techniques for object-oriented programs. In M.Jarke, K.Pasedach, and K.Pohl, editors, *Informatik97: Informatik als Innovationsmotor*, Informatik Aktuell, pages 602-611. Springer-Verlag, 1997.

INVITED ARTICLES

[145] V. Astrauskas, A. Bílý, J. Fiala, Z. Grannan, C. Matheja, P. Müller, F. Poli, and A. J. Summers: The Prusti Project: Formal Verification for Rust. *NASA Formal Methods (NFM)*, volume 13260 of *LNCS*, pages 88-108, Springer-Verlag, 2022.

[146] P. Müller, M. Schwerhoff, and A.J. Summers. Viper: A Verification Infrastructure for Permission-Based Reasoning. In B. Jobstman and K.R.M. Leino, editors, *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 9583 of *LNCS*, pages 41-63, Springer-Verlag, 2016.

[147] K.R.M. Leino and P. Müller. Using the Spec# Language, Methodology, and Tools to Write Bug-Free Programs. In P. Müller, editor, *Advanced Lectures on Software Engineering – LASER Summer School 2007/2008*, volume 6029 of *LNCS,* pages 91-139, Springer-Verlag, 2010.

[148] K.R.M. Leino, P. Müller, and J. Smans. Verification of concurrent programs with Chalice. In A. Aldini and R. Gorrieri, editors, *Foundations of Security Analysis and Design V*, volume 5705 of *LNCS*, pages 195-222, Springer-Verlag, 2009.

[149] E. Albert, A. Banerjee, S. Drossopoulou, M. Huisman, A. Igarashi, G.T. Leavens, P. Müller, and T. Wrigstad. Formal techniques for Java-like programs (FTfJP). In P. Eugster, editor, *Object-Oriented Technology. ECOOP2008 Workshop Reader*, volume 5475 of *LNCS* pages 70-76. Springer-Verlag, 2008.

[150] D. Clarke, S. Drossopoulou, P. Müller, J. Noble, and T. Wrigstad. Aliasing, confinement and ownership in object-oriented programming (IWACO). In P. Eugster, editor, *Object-Oriented Technology. ECOOP2008 Workshop Reader*, volume 5475 of *LNCS*, pages 30-41. Springer-Verlag, 2008.

[151] G. Barthe, L. Beringer, P. Crégut, B. Grégoire, M. Hofmann, P. Müller, E. Poll, G. Puebla, I. Stark, and E.Vétillard. MOBIUS: Mobility, ubiquity, security. In U. Montanari, D. Sannella, and R. Bruni, editors, *Trustworthy Global Computing*, volume 4661 of *LNCS*, pages 10-29. Springer-Verlag, 2006.

[152] A. Coglio, M. Huisman, J. Kiniry, P. Müller, and E. Poll. Formal techniques for Java-like programs (FTfJP). In J. Malenfant and B.M. Østvold, editors, *Object-Oriented Technology. ECOOP2004 Workshop Reader*, volume 3344 of *LNCS*, pages 76-83. Springer-Verlag, 2005.

[153] S. Eisenbach, G.T. Leavens, P. Müller, A. Poetzsch-Heffter, and E. Poll. Formal techniques for Java-like programs. In F. Buschmann, A.P. Buchmann, and M. Cilia, editors, *Object-Oriented Technology. ECOOP'03 Workshop Reader*, volume 3013 of *LNCS*, pages 62-71. Springer-Verlag, 2004.

[154] S. Drossopoulou, S. Eisenbach, B. Jacobs, G.T. Leavens, P. Müller, and A. Poetzsch-Heffter. Formal techniques for Java programs. In J. Malenfant, S. Moisan, and A. Moreira, editors, *Object-Oriented Technology. ECOOP2000 Workshop Reader*, volume 1964 of *LNCS*, pages 41-54. Springer-Verlag, 2000.

[155] B. Jacobs, G.T. Leavens, P. Müller, and A. Poetzsch-Heffter. Formal techniques for Java programs. In A. Moreira and D. Demeyer, editors, *Object-Oriented Technology. ECOOP'99 Workshop Reader*, volume 1743 of *LNCS*, pages 97-115. Springer-Verlag, 1999.

TECHNICAL REPORTS (SELECTION)

[156] F. Poli, X. Denis, P. Müller, and A. J. Summers: Reasoning about Interior Mutability in Rust using Library-Defined Capabilities. arXiv, 2024.

[157] A. Bílý, J. C. Pereira, J. Schär, P. Müller. Refinement Proofs in Rust Using Ghost Locks, arXiv, 2023.

[158] T. Dardinier and P. Müller. Hyper Hoare Logic: (Dis-)Proving Program Hyperproperties (extended version), arXiv, 2023.

[159] T. Dardinier, G. Parthasarathy, N. Weeks, A. J. Summers, and P. Müller. Sound Automation of Magic Wands (extended version), arXiv, 2022.

[160] T. Dardinier, G. Parthasarathy and P. Müller: Verification-Preserving Inlining in Automatic Separation Logic Verifiers (extended version), arXiv, 2022.

[161] A. Bílý, J. Hansen, P. Müller, and A. J. Summers. Compositional Reasoning for Side-effectful Iterators and Iterator Adapters, arXiv, 2022.

[162] M. Eilers , T. Dardinier, and P. Müller. CommCSL: Proving Information Flow Security for Concurrent Programs using Abstract Commutativity, arXiv, 2022.

[163] L. Arquint , M. Schwerhoff, V. Mehta, and P. Müller. A Generic Methodology for the Modular Verification of Security Protocol Implementations, arXiv, 2022.

[164] L. Arquint , F. A. Wolf, J. Lallemand, R. Sasse, C. Sprenger, S. N. Wiesner, D. A. Basin, and P. Müller. Sound Verification of Security Protocols: From Design to Interoperable Implementations (extended version) , arXiv, 2022.

[165] C. Bräm, M. Eilers , P. Müller, R. Sierra, and A. J. Summers. Modular Verification of Collaborating Smart Contracts, arXiv, 2021.

[166] A. Bugariu, A. Ter-Gabrielyan, and P. Müller. Identifying Overly Restrictive Matching Patterns in SMT-based Program Verifiers, arXiv, 2021.

[167] F. A. Wolf , L. Arquint , M. Clochard, W. Oortwijn, J. Carlos Pereira , and P. Müller. Gobra: Modular Specification and Verification of Go Programs (extended version) , arXiv, 2021.

[168] G. Parthasarathy, P. Müller, and A. J. Summers. Formally Validating a Practical Verification Condition Generator (extended version) , arXiv, 2021.

[169] A. Bílý, C. Matheja, and P. Müller. Flexible Refinement Proofs in Separation Logic, arXiv, 2021.

[170] F. A. Wolf, M. Schwerhoff, and P. Müller: Concise Outlines for a Complex Logic: A Proof Outline Checker for TaDA (Full Paper). Technical Report, arXiv, 2020.

[171] C. Sprenger, T. Klenze, M. Eilers, F. A. Wolf, P. Müller, M. Clochard, and D. Basin: Igloo: Soundly Linking Compositional Refinement and Separation Logic for Distributed System Verification. *Technical Report, arXiv,* 2020.

[172] C. Dross, C.A. Furia, M. Huisman, R. Monahan, and P. Müller: VerifyThis 2019: A Program Verification Competition (Extended Report). Technical Report, arXiv, 2020.

[173] V. Astrauskas, P. Müller, F. Poli, and A.J. Summers: Leveraging Rust Types for Modular Specification and Verification. Technical Report, ETH Zurich, 2019.

[174] A.J. Summers and P. Müller. Automating Deductive Verification for Weak-Memory Programs (extended version). Technical Report. arXiv, 2018

[175] M. Christakis, P. Emmisberger, P. Godefroid, and P. Müller. A General Framework for Dynamic Stub Injection. Technical Report, Microsoft Research, 2016.

[176] D. Asenov, P. Müller, and L. Vogel. The IDE as a Scriptable Information System (extended version). arXiv:1607.04452 [cs.SE], 2016.

[177] M. Huisman, R. Monahan, P. Müller, and E. Poll. VerifyThis 2016: A Program Verification Competition. Technical Report, Centre for Telematics and Information Technology, University of Twente, 2016.

[178] M. Christakis and P. Müller and V. Wüstholz. Guiding Dynamic Symbolic Execution toward Unverified Program Executions. Technical Report, ETH Zurich, 2015.

[179] C. Favre and H. Völzer and P. Müller. Diagnostic Information for Control-Flow Analysis of Workflow Graphs (aka Free-Choice Workflow Nets). Technical Report, ETH Zurich, 2015.

[180] P. Boström and P. Müller. Modular Verification of Finite Blocking in Non-terminating Programs. Technical Report, ETH Zurich, 2014

[181] M. Christakis, P. Müller, and V. Wüstholz. An Experimental Evaluation of Deliberate Unsoundness in a Static Program Analyzer. Technical Report, ETH Zurich, 2014.

[182] J. T. Boyland, P. Müller, M. Schwerhoff, and A. J. Summers. Constraint Semantics for Abstract Read Permissions. Technical Report, ETH Zurich, 2014

[183] U. Juhasz, I. T. Kassios, P. Müller, M. Novacek, M. Schwerhoff, and A. J. Summers. Viper: A Verification Infrastructure for Permission-Based Reasoning, Technical Report, ETH Zurich, 2014

[184] D. Asenov and P. Müller. Envision. A Fast and Flexible Visual Code Editor with Fluid Interactions, Technical Report, ETH Zurich, 2014.

[185] P. Ferrara, P. Müller, and M. Novacek. Automatic Inference of Heap Properties Exploiting Value Domains. Technical Report 794, ETH Zurich, 2013.

[186] S. Heule, I.T. Kassios, P. Müller, and A.J. Summers, Verification Condition Generation for Permission Logics with Abstraction Functions, Technical Report 761, ETH Zurich, 2012.

[187] A. J. Summers and P. Müller. Freedom before Commitment: Simple Flexible Initialisation for Non-Null Types, Technical Report 716, ETH Zürich, 2011.

[188] P. Müller and J.N. Ruskiewicz. Using Debuggers to Understand Failed Verification Attempts, Technical Report 713, ETH Zurich, 2011.

[189] K. R. M. Leino, P. Müller, and J. Smans. Deadlock-free Channels and Locks, Technical Report CW573, K.U. Leuven, 2010.

[190] W. Dietl, M. Ernst, and P. Müller. Tunable Universe Type Inference, Technical Report 659, ETH Zurich, 2010.

[191] I. T. Kassios and P. Müller. Specification and Verification of Closures, Technical Report 660, ETH Zurich, 2010.

[192] A. Rudich and P. Müller. Using stereotypes to verify complex heap structures in regional logic, Technical Report 651, ETH Zurich, 2009.

[193] M. Nordio, C. Calcagno, P. Müller, and B. Meyer. Soundness and Completeness of a Program Logic for Eiffel, Technical Report 617, ETH Zurich, 2009.

[194] M. Nordio, C. Calcagno, B. Meyer, and P. Müller. Reasoning about Function Objects, Technical Report 615, ETH Zurich, 2009.

[195] A. Rudich, Á. Darvas, and P. Müller. Checking Well-Formedness of Pure-Method Specifications (full Paper).Technical Report 588, ETH Zurich, 2008.

[196] M. Nordio, P. Müller and B. Meyer. Formalizing Proof-Transforming Compilation of Eiffel Programs. Technical Report 587, ETH Zurich, 2008.

[197] S. Drossopoulou, A. Francalanza, and P. Müller. A Unified Framework for Verification Techniques for Object Invariants (Full Paper), 2007.

[198] P. Müller and M. Nordio. Proof-transforming Compilation of Programs with Abrupt Termination. Technical Report 565, ETH Zurich, 2007. 65 pages.

[199] P. Müller and A. Rudich. Formalization of Ownership Transfer in Universe Types. Technical Report 556, ETH Zurich, 2007. 53 pages.

[200] Á. Darvas and P. Müller. Formal encoding of JML Level 0 specifications in JIVE. Technical Report 559, ETH Zurich, 2007. Annual Report of the Chair of Software Engineering. 17 pages.

[201] W. Dietl, S. Drossopoulou, and P. Müller. Formalization of Generic Universe Types. Technical Report 532, ETH Zurich, 2006. 52 pages.

[202] G.T. Leavens, E. Poll, C. Clifton, Y. Cheon, C. Ruby, D. Cok, P. Müller, and J. Kiniry. JML reference manual. Department of Computer Science, Iowa State University, 2005.

[203] F.Y. Bannwart and P. Müller. A Logic for Bytecode. Technical Report 469, ETH Zurich, 2004. 59 pages.

[204] K.R.M. Leino and P. Müller. Modular verification of global module invariants in object-oriented programs. Technical Report 459, ETH Zurich, 2004. 16 pages.

[205] S. Eisenbach, G.T. Leavens, P. Müller, A. Poetzsch-Heffter, and E. Poll, editors. Formal Techniques for Java-like Programs. Technical Report 408, ETH Zurich, 2003.

[206] P. Müller, A. Poetzsch-Heffter, and G.T. Leavens. Modular Specification of Frame Properties in JML. Technical Report 01-03, Department of Computer Science, Iowa State University, Ames, Iowa, 50011, April 2001.

[207] P. Müller and A. Poetzsch-Heffter. Universes: A Type System for Alias and Dependency Control. Technical Report 279, Fernuniversität Hagen, 2001. 25 pages.

[208] S. Drossopoulou, S. Eisenbach, B. Jacobs, G.T. Leavens, P. Müller, and A.Poetzsch-Heffter, editors. Formal Techniques for Java Programs. Technical Report 269, Fernuniversität Hagen, 2000.

[209] M. Labeth, J. Meyer, P. Müller, and A. Poetzsch-Heffter. Formal Verification of a Doubly Linked List Implementation: A Case Study Using the JIVE System. Technical Report 270, Fernuniversität Hagen, 2000.

[210] B. Jacobs, G.T. Leavens, P. Müller, and A. Poetzsch-Heffter, editors. Formal Techniques for Java Programs. Technical Report 251, Fernuniversität Hagen, 1999.

[211] P. Müller, J. Meyer, and A. Poetzsch-Heffter. Programming and Interface Specification Language of JIVE Specification and Design Rationale. Technical Report 223, Fernuniversität Hagen, 1997.

[212] P. Müller and A. Poetzsch-Heffter. A Brief Study in Automating Proofs Based on a Refined Hoare-logic. Technical Report TUM-I9635, Technische Universität München, 1996.

Most articles are available from www.pm.inf.ethz.ch/publications.html.

AWARDS AND SCHOLARSHIPS

2025	Distinguished Paper Award, Programming Language Design and Implementation (PLDI)
2025	ETAPS Test-of-Time Award for the ESOP 2009 paper “A Basis for Verifying Multi-Threaded Programs”
2025	Best Reviewer Award, European Symposium on Programming (ESOP)
2024	CYD Award for Rising Security Technology, Cyber Defence Campus
2023	Distinguished Artifact Award, Programming Language Design and Implementation (PLDI)
2022	Distinguished Paper Award, Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)
2021	Distinguished Artifact Award, Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)
2021	Amazon Research Award
2019	Distinguished Artifact Award, Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)
2018	Distinguished Artifact Award, Tools and Algorithms for the Construction and Analysis of Systems (TACAS)
2017	Facebook Faculty Research Award
2016	Microsoft Research Outstanding Collaborator Award
2016	ACM SIGSOFT Distinguished Paper Award, International Conference on Software Engineering (ICSE)

2012	Silver Medal (with Rustan Leino), VSTTE 2012 Program Verification Competition
2011	Best Paper Award, European Conference on Object-Oriented Programming (ECOOP)
2011	Best Paper Award, Formal Methods (FM)
2008	Goldene Eule (ETH teaching award)
2001	Dissertation Award of Fernuniversität Hagen
1998	Scholarship for my research project in the US by the DAAD (German Academic Exchange Service)
1996	IBM Award for excellent Master's thesis at Technische Universität München
1995 - 1996	Scholarship by Siemens Nixdorf Informationssysteme AG
1991 - 1996	Bavarian scholarship for exceptionally talented students

ONGOING RESEARCH PROJECTS

Center for Cyber Trust

Funded by the Werner Siemens Foundation

Duration: 8 years, started January 2020

Funding: CHF 2’825’000

Development of Automated SBOM and VEX Verification Technologies for

Securing Software Supply Chains

Funded by Korea University Research and Business Foundation

Duration: 2.5 years, started July 2024

Funding: CHF 130’000

Toward a Fully-Verified SCION Router

Funded by NGI Zero

Duration: 12 months, started October 2024

Funding: € 30,000

Practical Rust Verification

Funded by Stellar Development Foundation

Duration: 12 months, started August 2024

Funding: US$ 145,000

COMPLETED RESEARCH PROJECTS

Formal Foundations of Translational Program Verifiers

Funding agency: Swiss National Science Foundation (SNF)

Duration: 3 years, started October 2020

Funding: CHF 315’778

Verification of Rust Programs against TLA+ Specifications

Funded by Amazon Web Services

Duration: 1 year, started September 2021

Funding: US$ 40’000, AWS credit US$ 20’000

Verified Secure Routing with SCION

Funded by the Horizon 2020 Framework Programme of the European Union, NGI Pointer

Duration: 15 months, started October 2020

Funding: € 68,000

Rust Verification

Funded by Amazon Web Services

Duration: 1 year, started January 2020

Funding: US$ 50’000

SCION Infrastructure for the ETH Domain (SCI-ED)

Funded by ETH Board

Duration: 2 years, started February 2019

Funding: 1 PhD student and 2 post-docs, 2 years each

From Type Capabilities to Permissions for Program Verification (and back again)

Funding agency: Swiss National Science Foundation (SNF)

Duration: 3 years, started January 2017

Funding: CHF 342’016

Reasoning for Safe Clients of Unsafe Rust Libraries

Funded by Facebook

Duration: 1 year, started July 2019

Funding: US$ 80’000

verifiedSCION

Funded by Zurich Information Security Center (ZISC)

Duration: 3 years, started September 2016

Funding: CHF 267’800

Verification Infrastructure for Permission-Based Reasoning

Funding agency: Swiss National Science Foundation (SNF)

Duration: 3 years, started September 2015

Funding: CHF 339'100

TANDEM – An Architecture for Complimentary Verification and Testing

Funding agency: ETH Zurich, ETH Research Grants

Duration: 3 years, started February 2011

Funding: CHF 304’000

Automatic Verification of Concurrent Programs with Tamed Mutable State

Funding agency: Hasler Foundation

Duration: 3 years, started May 2011

Funding: CHF 300’000

Verification Driven Inference of Contracts

Funding agency: Swiss National Science Foundation (SNF)

Duration: 3 years, started September 2010

Funding: CHF 302'896

Cloud-based Static Analyses to Improve Lay Programming on Mobile Devices

Funded by Microsoft Schweiz GmbH and Microsoft Research Limited

Duration: 1 year, started August 2012

Funding: CHF 298’000

Formal Verification of Object-Oriented Software (COST Action IC0701)

Project partners: 15 participating countries

Funding agency: European Cooperation in the field of Scientific and Technical Research

Duration: 4 years, started March 2008

Funding: Travelling to COST meetings

Automatic Verification of Heap Structures in C

Funded by Dassault Aviation

Duration: 1 year, started September, 2009

Funding: Euro 70’000

Mobility, Ubiquity and Security (Mobius)

Project partners: Consortium of 18 partners

Funding agency: European Union, Global Computing II

Duration: 4 years, started September 2005

Funding: Euro 758’801

Heterogeneous Proof-Carrying Components

Project partners: Peter Müller (principal investigator), Bertrand Meyer

Funding agency: ETH Zurich

Duration: 4 years, started March 2005

Funding: 2 assistant positions (60%) for 3 years, CHF 12’000 travel costs

TALKS

1. “Hyper Hoare Logic”. IFIP Working Group 1.9/2.15 Meeting, Paris, France, July 04, 2025.
2. “Automated Modular Program Verification Using Viper”. Korea University, June 17, 2025. Invited
3. “Hyper Hoare Logic”. IFIP Working Group 2.3 Meeting, Athens, Greece, May 19, 2025.
4. “A Basis for Verifying Multi-Threaded Programs – Sixteen Years Later”. ETAPS Test of Time Award, Hamilton, Canada, May 08, 2025. Invited
5. “CommCSL: Proving Information Flow Security for Concurrent Programs using Abstract Commutativity”, Workshop on Big Specification at the Isaac Newton Institute, Cambridge, UK, October 10, 2024.
6. “Verified Secure Routing”. Computer Security Foundations (CSF), Enschede, The Netherlands, July 09, 2024. Invited.
7. “On Long-Term Research Problem Selection”, PLDI workshop on Red-Hot Topics in Faculty Mentoring, Copenhagen, Denmark, June 25, 2024. Invited.
8. “Verified Secure Routing”, IFIP Working Group 2.3 Meeting, Princeton, USA, May 06, 2024. Invited.
9. “CommCSL: Proving Information Flow Security for Concurrent Programs using Abstract Commutativity”, Challenges of Software Verification Symposium 2024, Venice, Italy, June 06, 2024.
10. “Verified Secure Routing”, ETAPS workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software (PLACES), Luxembourg, April 06, 2024. Invited
11. “Verified Secure Routing”, National University of Singapore, February 09, 2024. Invited.
12. “Automated Modular Program Verification Using Viper”, National University of Singapore, February 08, 2024. Invited.
13. “Proving Information Flow Security for Concurrent Programs”, Nordic Workshop on Programming Theory, Västeras, Sweden, November 23, 2023. Invited.
14. “Proving Information Flow Security for Concurrent Programs”, Iowa State University, Ames, Iowa, USA, October 24, 2023. Invited.
15. “Verified Secure Routing”, Verified Software – Theory, Tools, Experiments (VSTTE), Ames, Iowa, USA, October 23, 2023. Invited.
16. “Proving Information Flow Security for Concurrent Programs”, IFIP Working Group 2.3 Meeting, Trento, Italy, October 09, 2023. Invited.
17. “Verified Secure Routing”, Hochschule St. Gallen, Switzerland, September 21, 2023.Invited.
18. “Verified Secure Routing”, Networked Systems (NETYS), Marrakesh, Morocco, May 22, 2023. Invited.
19. “Verified Secure Routing”, Dagstuhl seminar on Unifying Formal Methods for Trustworthy Distributed Systems, Dagstuhl, Germany, March 13, 2023. Invited.
20. “Modular Program Verification in Viper”, Certora, online, January 26, 2023. Invited.
21. “Automated Modular Program Verification”, INSAIT Conference, Sofia, Bulgaria, October 01, 2022. Invited.
22. “Prusti: Deductive Verification for Rust”, Workshop on Challenges of Software Verification, Venice, Italy, May 20, 2022. Invited.
23. “You and Your PhD”, ETAPS Mentoring Workshop, Munich, Germany, April 03, 2022. Invited.
24. “Prusti - Deductive Verification for Rust”, Workshop on Verified Software: Tools and Experiments at the Isaac Newton Institute, Cambridge, UK, June 09, 2021.
25. “Modular Verification of Safe Rust Programs”, Dfinity, Zurich, Switzerland, December 17, 2019.
26. “Modular Verification of Safe Rust Programs”, IFIP Working Group 2.3 Meeting, Los Altos, USA, October 28, 2019.
27. “Verified Secure Routing”, KU Leuven, Leuven, Belgium, August 21, 2019. Invited.

28. “Program Verification for Correct and Secure Code”, Huawei Security Technology SAB Summit, Munich, Germany, July 10, 2019. Invited.
29. “Modular Verification of Safe Rust Programs”, Amazon, Seattle, USA, May 28, 2019. Invited.
30. “Verified Secure Routing”, Max-Planck Institute for Software Systems, Kaiserslautern, Germany, November 19, 2018. Distinguished Lecture.
31. “The Binomial Heap Verification Challenge in Viper”, Max-Planck Institute for Software Systems, Kaiserslautern, Germany, November 16, 2018.
32. “Abstract Interpretation of CTL Properties”. ETH Workshop on Dependable and Secure Software Systems, Zurich, Switzerland, October 19, 2018.
33. “Abstract Interpretation of CTL Properties”. Static Analysis Symposium (SAS), Freiburg, Germany, August 30, 2018.
34. “Reasoning with Permissions in Viper”. University of Twente, March 02, 2018. Invited.
35. “Modular Verification of Secure Information Flow”. ETH Workshop on Software Correctness and Reliability, Zurich, Switzerland, October 13, 2017.
36. “Reasoning with Permissions in Viper”. Static Analysis Symposium (SAS), New York, USA, August 30, 2017. Invited.
37. “Modular Verification of Secure Information Flow”. IFIP Working Group 2.3 Meeting, Mooloolaba, Australia, July 17, 2017.
38. “Modular Verification of Finite Blocking in Non-terminating Programs”. University of Freiburg, Germany, May 31, 2017. Invited.
39. “Partial Verification Results”, IFIP Working Group 1.9/2.15 Meeting, Leuven, Belgium, May 11, 2017. Invited.
40. “Prototyping Relaxed Separation Logic in Viper”, ETH Workshop on Software Correctness and Reliability, Zurich, Switzerland, October 07, 2016.
41. “Partial Verification Results”, Dagstuhl seminar on Synergies among Testing, Verification, and Repair for Concurrent Programs, Dagstuhl, Germany, May 19, 2016.
42. “Reasoning about Concurrent Programs in Viper”, ETAPS workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software (PLACES), Eindhoven, The Netherlands, April 08, 2016. Invited.
43. “Building Modular Program Verifiers using the Viper Infrastructure”, ETAPS workshop on Verification and Synthesis for Software Evolution (VSSE), Eindhoven, The Netherlands, April 02, 2016. Invited.
44. “Viper – A Verification Infrastructure for Permission-based Reasoning”, Verification, Model Checking, and Abstract Interpretation (VMCAI), St. Petersburg, USA, January 19, 2016. Invited.
45. “Viper – A Verification Infrastructure for Permission-based Reasoning”, ETH Workshop on Software Correctness and Reliability, Zurich, Switzerland, October 03, 2015.
46. “Toward Practical Program Verification”, Lorentz Center seminar on Verification of Concurrent and Distributed Software, Leiden, The Netherlands, September 14, 2015. Invited.
47. “Modular Verification of Finite Blocking in Non-terminating Programs”. European Conference on Object-Oriented Programming (ECOOP), Prague, Czech Republic, July 10, 2015.
48. “Viper – A Verification Infrastructure for Permission-based Reasoning”. Dagstuhl seminar on Compositional Verification Methods for Next-Generation Concurrency, Dagstuhl, Germany, May 04, 2015. Invited.
49. “Partial Verification Results”. IFIP Working Group 2.3 Meeting, Istanbul, Turkey, March 23, 2015. Invited.
50. “Building Automatic Program Verifiers”. Programming Languages Mentoring Workshop (PLMW), Mumbai, India, January 14, 2015. Invited.
51. “TouchGuru – Static Analysis for a Mobile Development Environment, ETH Workshop on Software Correctness and Reliability, Zurich, Switzerland, October 03, 2014.

52. “Tools for the Development of Correct Software”. Oracle Virtual Machine Meetup, Zurich, Switzerland, September 12, 2014. Invited.

53. “Viper–A Verification Infrastructure for Permission-based Reasoning”. Dagstuhl Seminar on Next Generation Static Software Analysis Tools, Dagstuhl, Germany, August 29, 2014. Invited.

54. “Modular Verification of Finite Blocking”. IFIP Working Group 1.9/2.15 Meeting, Orlando, USA, December 11, 2013. Invited.

55. “Collaborative Verification and Testing”. ETH Workshop on Software Correctness and Reliability, Zurich, Switzerland, October 04, 2013.

56. “Modular Verification of Finite Blocking”. Max-Planck Institute for Software Systems, Kaiserslautern, Germany, August 29, 2013. Invited.

57. “TouchBoost – Static Analyses for TouchDevelop”. Tutorial at Programming Language Design and Implementation (PLDI), Seattle, USA, June 16, 2013. Invited.

58. “Modular Verification of Finite Blocking”. University of Washington, Seattle, USA, June 04, 2013. Invited.

59. “Towards a Cost Analysis for TouchDevelop”. Microsoft Research, Redmond, USA, May 02, 2013. Invited.

60. “TouchBoost – Static Analyses for TouchDevelop”. TouchDevelop Workshop 2013, Mountain View, USA, February 15, 2013. Invited.

61. “Fractional Permissions without the Fractions”. IFIP Working Group 1.9/2.15 Meeting, Kirkland, USA, July 17, 2012. Invited.

62. “Using the Spec# Language, Methodology, and Tools to Write Bug-Free Programs”. Deploy Tutorial, Newcastle, UK, April 03, 2012. Invited.

63. “Contracts Reloaded”. SWEN – Software Engineering Network, Zurich, Switzerland, November 15, 2011. Invited.

64. “Freedom before Commitment – A Lightweight Type System for Object Initialization”. Université Paris-Sud, Paris, France, October 14, 2011. Invited.

65. “Freedom before Commitment – A Lightweight Type System for Object Initialization”. Tools for Automatic Program Analysis (TAPAS), Venice, Italy, September 17, 2011. Invited.

66. “Freedom before Commitment – A Lightweight Type System for Object Initialization”. Microsoft Research, Redmond, USA, July 28, 2011. Invited.

67. “Freedom before Commitment – A Lightweight Type System for Object Initialization”. University of Washington, Seattle, USA, July 11, 2011. Invited.

68. “Freedom before Commitment – A Lightweight Type System for Object Initialization”. IFIP Working Group 1.9/2.14 Meeting, Menlo Park, USA, June 15, 2011. Invited.

69. “Contracts Reloaded”. Avaloq Product Development Conference, Zurich, June 07, 2011. Invited.

70. “Freedom before Commitment – Simple Flexible Initialization for Non-Null Types”. IFIP Working Group 2.3 Meeting, Santa Barbara, USA, January 17, 2011. Invited.

71. “Automatic Verification of Concurrent Programs in Chalice”. Rigorous Systems Engineering Seminar at TU Vienna, Austria, May 18, 2010. Invited.

72. “Specification and Verification of Closures”. EPF Lausanne, Switzerland, March 30, 2010. Invited.

73. “Efficient Runtime Assertion Checking of Assignable Clauses with Datagroups”. Fundamental Approaches to Software Engineering (FASE), Paphos, Cyprus, March 24, 2010.

74. “Proving Consistency and Completeness of Model-Class Specifications by Theory Interpretation”. Fundamental Approaches to Software Engineering (FASE), Paphos, Cyprus, March 23, 2010.

75. “Deadlock-free Channels“. COST meeting, Eindhoven, The Netherlands, November 03, 2009.

76. “The Spec# Programming System“. ifi Colloquium at University of Zurich, Zurich, Switzerland, October 29, 2009. Invited.

77. “The Spec# Verification Methodology”. Workshop of the PUMA doctorate program, Venice, Italy, October 06, 2009. Invited.

78. “A Basis for Verifying Multi-Threaded Programs”. Microsoft Research, Redmond, USA, August 11, 2009. Invited.

79. “Locks, Channels, and Deadlock Freedom”. Microsoft Research, Redmond, USA, August 13, 2009. Invited.

80. “PIP – A Case Study Using Regions and Stereotypes”. Dagstuhl seminar on Typing, Analysis and Verification of Heap-Manipulating Programs, Dagstuhl, Germany, July 21, 2009. Invited.

81. “A Quick Tour of Ownership”. Dagstuhl seminar on Typing, Analysis and Verification of Heap-Manipulating Programs, Dagstuhl, Germany, July 20, 2009. Invited.

82. “JML@ETH”. Dagstuhl seminar on The Java Modeling Language (JML), Dagstuhl, Germany, July 15, 2009. Invited.

83. “A Basis for Verifying Multi-Threaded Programs”. IFIP Working Group 2.3 Meeting, Boston, USA, June 08, 2009. Invited.

84. “A Basis for Verifying Multi-Threaded Programs”. European Symposium on Programming (ESOP), York, UK, March 27, 2009.

85. "The Spec# Programming System". IBM Research, Rüschlikon, Switzerland, December 11, 2008. Invited.

86. "A Basis for Verifying Multi-Threaded Programs". COST meeting, Madrid, Spain, December 01, 2008.

87. "The Spec# Programming System". Google, Zurich, Switzerland, November 11, 2008. Invited.

88. "Alias Control with Universe Types". Microsoft Research-INRIA Joint Centre, Paris, France, June 20, 2008. Invited.

89. "Verification of Equivalent-Results Methods". European Symposium on Programming (ESOP), Budapest, Hungary, April 02, 2008.

90. "A Unified Framework for Verification Techniques for Object Invariants". Foundations of Object-Oriented Languages (FOOL), San Francisco, USA, January 13, 2008.

91. "Writing Better Programs with Spec#". prio Conference, Baden-Baden, Germany, November 13, 2007. Invited.

92. "Ownership transfer in Universe Types". Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Montreal, Canada, October 25, 2007.

93. "A Unified Framework for Verification Techniques for Object Invariants". IFIP Working Group 2.3 Meeting, Santa Fe, USA, October 08, 2007. Invited.

94. "A Programming Methodology for Verifiable Object-Oriented Programs". ETH Zurich, Switzerland, October 03, 2007. Invited.

95. "2007 State of the Universe Address". International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO), Berlin, Germany, July 29, 2007.

96. "Generic Universe Types". ETAPS tutorial on Mobility, Ubiquity, and Security, Braga, Portugal, April 01, 2007.

97. "Modular Verification of Object Invariants in Spec#". ETAPS workshop on Bytecode Semantics, Verification, Analysis and Transformation (BYTECODE), Braga, Portugal, March 31, 2007. Invited.

98. "Alias Control with Universe Types". ETAPS workshop on Heap Analysis and Verification (HAV), Braga, Portugal, March 25, 2007. Invited.

99. "Modular Verification of Object Invariants in Spec#". Copenhagen Programming Language Seminar, the IT University of Copenhagen, Denmark, March 15, 2007. Invited.

100. "Generic Universe Types". Dagstuhl seminar on Mobility, Ubiquity, and Security, Dagstuhl, Germany, March 02, 2007. Invited.

101. "Alias Control with Universe Types: Overview and Challenges". IFIP Working Group 2.3 Meeting, Sydney, Australia, January 11, 2007. Invited.

102. "Alias Control with Universe Types: Overview and Challenges". TU Kaiserslautern, Germany, December 11, 2006. Invited.

103. "Program Verification with Spec# and Boogie". EPF Lausanne, Switzerland, November 23, 2006. Invited.

104. "Alias Control with Generic Universe Types". Trustworthy Global Computing, Lucca, Italy, November 07, 2006.

105. "Specification and Verification Challenges". ESF Exploratory Workshop on Challenges in Java Program Verification, Nijmegen, The Netherlands, October 17, 2006. Invited.

106. "Reasoning about Method Calls in Interface Specifications". KeY Meeting, Speyer, Germany, June 07, 2006. Invited.

107. "Data Abstraction in Spec# and Boogie". Microsoft Research, Bellevue, USA, May 24, 2006. Invited.

108. "Data Abstraction in Spec# and Boogie". Dagstuhl seminar on Rigorous Methods for Software Construction and Analysis, Dagstuhl, Germany, May 08, 2006. Invited.

109. "Modular Verification of Object Invariants". Dagstuhl seminar GIBU2006: GI-Beirat der Universitätsprofessoren, Dagstuhl, Germany, April 11, 2006. Invited.

110. "A Verification Methodology for Model Fields". ESOP, Vienna, Austria, March 27, 2006.

111. "Data Abstraction in Spec# and Boogie". IFIP Working Group 2.3 Meeting, Bruges, Belgium, March 13, 2006. Invited.

112. "Ownership-Overview and Challenges". Microsoft Research, Bellevue, USA, February 28, 2006. Invited.

113. "Modular Verification of Object Invariants". University of Saarbrücken, Germany, January 17, 2006. Invited.

114. "Program Verification with Spec# and Boogie". European Computer Science Summit, Zurich, Switzerland, October 21, 2005. Invited.

115. "Reasoning About Object Structures Using Ownership". VSTTE, Zurich, Switzerland, October 11, 2005.

116. "Modular Verification of Static Class Invariants". FM, Newcastle, UK, July 20, 2005.

117. "Modular Verification of Invariants over Static Fields". Microsoft Research, Bellevue, USA, October 05, 2004. Invited.

118. "Modular Verification of Object and Module Invariants". IFIP Working Group 2.3 Meeting, Prato, Italy, September 06, 2004. Invited.

119. "Object Invariants in Dynamic Contexts". ECOOP, Oslo, Norway, June 17, 2004.

120. "A Type System for Checking Applet Isolation in Java Card". CASSIS, Marseille, France, March 12, 2004. Invited.

121. "Modular Invariants for Object Structures". FATS seminar, ETH Zurich, Switzerland, February 04, 2004.

122. "Programming with Object Structures". Inaugural lecture, ETH Zurich, Switzerland, January 12, 2004.

123. "Modular Specification and Verification of Object-Oriented Software Components". ETH Zurich, Switzerland, August 20, 2002. Invited.

124. "Modular Specification and Verification of Object-Oriented Programs". Fernuniversität Hagen, Germany, May 10, 2001.

125. "Tool-Based Specification and Verification of Java Programs". T-Nova, Darmstadt, Germany, March 12, 2001. Invited.

126. "A Type System for Controlling Representation Exposure in Java". VerifiCard project meeting, Hagen, Germany, February 20, 2001.

127. "Logical Techniques for Applet Verification". VerifiCard kick-off meeting, Nijmegen, The Netherlands, January 20, 2001.

128. "Logical Techniques for Applet Verification". European Commission, Brussels, Belgium, August 22, 2000.
129. "A Type System for Controlling Representation Exposure in Java". ECOOP Workshop \it Formal Techniques for Java Programs, Cannes, France, June 12, 2000.
130. "A Programming Logic for Object-Oriented Languages". Ludwig-Maximilian-Universität München, Munich, Germany, November 16, 1999. Invited.
131. "Executable Interface Specifications for Java". Java User Group Switzerland, Zurich, Switzerland, November 11, 1999. Invited.
132. "Expressive Executable Interface Specifications for Java". Workshop Programming Languages and Fundamentals of Programming, Heinsberg, Germany, September 27, 1999.
133. "Expressive Executable Interface Specifications for Java". Java Information Days, Düsseldorf, Germany, September 21, 1999.
134. "Alias Control is Crucial for Modular Verification". ECOOP Workshop Aliasing in Object-Oriented Systems, Lisbon, Portugal, June 15, 1999.
135. "Encapsulation and Dynamic Method Binding: Java's Design Flaws and their Remedy". Java Information Days, Frankfurt, Germany, November 12, 1998.
136. "Formal Techniques for the Development of Correct Object-Oriented Programs". CCI Workshop Software Engineering, Bonn, Germany, October 21, 1998.
137. "Formal Specification and Verification of Object-Oriented Programs". Iowa State University, Ames, Iowa, USA, July 07, 1998. Invited.
138. "Preserving the Correctness of Object-Oriented Programs under Extension". Workshop Programming Languages and Fundamentals of Programming, Avendorf, Germany, September 24, 1997.
139. "Formal Specification Techniques for Object-Oriented Programs". GI Workshop Programming Languages, Aachen, Germany, September 23, 1997.

TEACHING

COURSES

AS 25	Concepts of Object-Oriented Programming
SS 25	Formal Methods and Functional Programming (with D. Basin, R. Jung)
SS 25	Program Verification (with M. Eilers)
AS 24	Concepts of Object-Oriented Programming
SS 24	Formal Methods and Functional Programming (with D. Basin)
SS 24	Program Verification (with M. Eilers)
AS 23	Concepts of Object-Oriented Programming
SS 23	Formal Methods and Functional Programming (with D. Basin)
SS 23	Program Verification (with M. Eilers)
AS 22	Concepts of Object-Oriented Programming
SS 22	Formal Methods and Functional Programming (with C. Sprenger)
SS 22	Program Verification
AS 21	Concepts of Object-Oriented Programming
SS 21	Formal Methods and Functional Programming (with D. Basin)
SS 21	Program Verification (with C. Matheja)
AS 20	Concepts of Object-Oriented Programming
SS 20	Formal Methods and Functional Programming (with D. Traytel)
AS 19	Concepts of Object-Oriented Programming
SS 19	Formal Methods and Functional Programming (with D. Basin)
AS 18	Concepts of Object-Oriented Programming
SS 18	Formal Methods and Functional Programming (with D. Basin)
SS 18	Software Architecture and Engineering (with M. Vechev)
AS 17	Concepts of Object-Oriented Programming
SS 17	Formal Methods and Functional Programming (with D. Basin)
SS 17	Software Architecture and Engineering (with M. Vechev)

AS 16	Concepts of Object-Oriented Programming
SS 16	Formal Methods and Functional Programming (with D. Basin)
SS 16	Software Architecture and Engineering (with M. Vechev)
AS 15	Concepts of Object-Oriented Programming
SS 15	Formal Methods and Functional Programming (with D. Basin)
SS 15	Software Architecture and Engineering (with M. Vechev
AS 14	Concepts of Object-Oriented Programming
SS 14	Formal Methods and Functional Programming (with A. Lochbihler)
SS 14	Software Architecture and Engineering (with M. Vechev)
AS 13	Concepts of Object-Oriented Programming
AS 12	Concepts of Object-Oriented Programming
SS 12	Software Architecture and Engineering
SS 12	Formal Methods and Functional Programming (with D. Basin)
AS 11	Concepts of Object-Oriented Programming
SS 11	Software Engineering
SS 11	Formal Methods and Functional Programming (with D. Basin)
AS 10	Concepts of Object-Oriented Programming
SS 10	Software Engineering
SS 10	Formal Methods and Functional Programming (with D. Basin)
AS 09	Concepts of Object-Oriented Programming
SS 09	Software Engineering
SS 09	Formal Methods and Functional Programming (with D. Basin)
AS 08	Concepts of Object-Oriented Programming
WS 06/07	Concepts of Object-Oriented Programming
SS 06	Software Engineering
WS 05/06	Concepts of Object-Oriented Programming
WS 05/06	IT Project Management
WS 04/05	Concepts of Object-Oriented Programming
WS 04/05	IT Project Management
SS 04	Semantics of Programming Languages
WS 03/04	Concepts of Object-Oriented Programming
WS 03/04	IT Project Management

SEMINARS

AS 25	Research Topics in Software Engineering (with R. Jung, Michalis Kokologiannakis)
AS 24	Research Topics in Software Engineering (with R. Jung, Michalis Kokologiannakis)
AS 23	Research Topics in Software Engineering (with R. Jung)
AS 22	Research Topics in Software Engineering (with M. Püschel)
AS 21	Research Topics in Software Engineering (with M. Püschel)
AS 20	Software Engineering (with M. Püschel, M. Schwerhoff)
AS 19	Research Topics in Software Engineering (with M. Püschel, Z. Su, M. Vechev)
SS 18	Programming Systems (with M. Vechev)
AS 17	Research Topics in Software Engineering (with T. Gross, M. Püschel, M. Vechev)
AS 17	Programming Languages (with M. Vechev)
SS 17	Programming Systems (with M. Vechev)
AS 16	Research Topics in Software Engineering (with M. Püschel)
AS 16	Programming Languages (with M. Vechev)
SS 16	Programming Systems (with M. Vechev)
AS 15	Research Topics in Software Engineering
AS 15	Programming Languages (with M. Vechev)
SS 15	Programming Systems (with M. Vechev)
AS 14	Software Engineering
AS 14	Programming Languages (with M. Vechev)
SS 14	Research Topics in Software Engineering (with M. Vechev)
SS 14	Programming Systems (with M. Vechev)
AS 13	Programming Languages (with M. Vechev)
SS 12	Research Topics in Software Engineering (with M. Vechev)
AS 11	Software and Security Testing
AS 09	Software Engineering
WS 06/07	Software Engineering

WS 05/06	Specification and Verification of Object-Oriented Software
WS 04/05	Specification and Verification of Object-Oriented Software
SS 04	References and Aliasing in Object-Oriented Programs
WS 03/04	Specification and Verification of Object-Oriented Software

INDUSTRY COURSES

Oct. 2013	Quality Assurance in .NET with Code Contracts
Oct. 2012	Quality Assurance in .NET with Code Contracts
Jan. 2012	Quality Assurance in .NET with Code Contracts
Feb. 2011	Principles of Project Management
Sep. 2010	Quality Assurance in .NET with Code Contracts
Feb. 2010	Principles of Project Management
Sep. 2009	Advanced Topics in Project Management
Feb. 2009	Principles of Project Management
Feb. 2007	Advanced Topics in Project Management
Sep. 2006	Principles of Project Management
Mar. 2006	Advanced Topics in Project Management
Sep. 2005	Principles of Project Management
Mar. 2005	IT Project Management
Feb. 2004	IT Project Management

SUMMER SCHOOLS

1. Lecturer at the Summer School on Security Testing and Verification. Brussels, Belgium, September 9 – 11, 2024.
2. Co-director and lecturer of the Marktoberdorf Summer School on Engineering Secure and Dependable Software Systems. Herrsching, Germany, August 06-17, 2024.
3. Lecturer at the Summer School on Verification Technology, Systems & Applications. Nancy, France, August 28 – September 01, 2023.
4. Lecturer at the Summer School on Formal Techniques. Atherton, USA, May 20-24, 2019.
5. Co-director and lecturer of the Marktoberdorf Summer School on Dependable Software Systems Engineering. Marktoberdorf, Germany, August 01-11, 2018.
6. Lecturer at the Marktoberdorf Summer School on Dependable Software Systems Engineering. Marktoberdorf, Germany, August 03-12, 2016.
7. Lecturer at the ICTAC School on Software Engineering. Bangalore, India, September 24, 2012.
8. Lecturer at the ARTIST Summer School on Embedded System Design. Aix-les-Bains, France, September 17-21, 2012.
9. Lecturer at the LASER Summer School on Software Engineering-Applied Software Verification. Elba, Italy, September 09-15, 2007.

SUPERVISED PH.D. STUDENTS (ALL ETH ZURICH)

1. A. Bílý. *Verification of Distributed Rust Programs*. In progress
2. L. Brugger. *Large-scale Verification*. In progress
3. J. Fiala. *Rust Verification*. In progress
4. A. Keusch. *Proving Program Equivalence*. In progress
5. N. Klose. *Specification Inference.* In progress
6. A. Lee. *Modular Verification of Privacy.* In progress

7. A. Li. *Type-Supported Deductive Verification*. In progress
8. H. Ling. *A Refinement Approach to Verify Probabilistic Distributed Systems*. In progress
9. Th. Mayerl. *Verification of Unsafe Rust*. In progress
10. J. Pereira. *Verification of a Secure Router*. In progress
11. D. Spiliopoulos. *Performance Optimization for Program Verifiers*. In progress
12. L. Arquint. *Modular Verification of Security Protocol Implementations*. 2025.
13. D. Asenov. Envision: *Reinventing the Integrated Development Environment*. 2017.
14. V. Astrauskas. *Leveraging Uniqueness for Modular Verification for Heap-Manipulating Programs.* 2024.
15. L. Brutschy. *Dynamic and Static Analysis for Weakly Consistent Systems*. 2017.
16. A. Bugariu. *Automatically Identifying Soundness and Completeness Errors in Program Analysis Tools*. 2022.
17. M. Christakis. *Narrowing the Gap between Verification and Systematic Testing*. 2015.
18. T. Dardinier. *Formal Foundations for Automated Deductive Verifiers*. 2025.
19. Á. Darvas. *Reasoning about Data Abstraction in Contract Languages*. 2009.
20. W. Dietl. *Universe Types: Topology, Encapsulation, Genericity and Tools*, 2009.
21. J. Dohrau. *Automatic Inference of Permission Specifications*. 2022.
22. M. Eilers. *Modular Specification and Verification of Security Properties for Mainstream Languages*. 2022.
23. C. Favre. *Detecting, Understanding, and Fixing Control-Flow Errors in Business Process Models*. 2014. External PhD student from IBM Research Rüschlikon.
24. U. Juhasz. *Incremental Verification.* 2016.
25. H. Lehner. *A Formal Definition of JML in Coq and its Application to Runtime Assertion Checking*. 2011.
26. F. Mehta. *Proofs for the Working Engineer*. 2008. Co-supervisor.
27. M. Nordio. *Proofs and Proof Transformations for Object-Oriented Programs*. 2009. Co-supervisor.
28. G. Parthasarathy. *Formally Validating Translational Program Verifiers*. 2024.
29. F. Poli. *Enabling Rich Lightweight Verification of Rust Software*. 2024.
30. A. Rudich. *Automatic Verification of Heap Structures with Stereotypes*. 2011.
31. J. Ruskiewicz. *Localizing and Understanding Verification Errors*. 2012.
32. M. Schwerhoff. *Advancing Automated, Permission-Based Program Verification Using Symbolic Execution*. 2016.
33. A. Ter-Gabrielyan. *Compositional Verification of Rich Program Properties in Separation Logic*. 2021
34. F. Wolf. *Automated Verification of Advanced Correctness and Security Properties*. 2024.
35. V. Wüstholz. *Partial Verification Results*. 2015.

SUPERVISED POST-DOCS (ALL ETH ZURICH)

1. Martin Clochard, 2018-21
2. Xavier Denis, 2024
3. Marco Eilers, 2022-
4. Pietro Ferrara, 2009-13
5. Nicu Fruja, 2007

6. Yannis Kassios, 2009-14
7. Laura Kovács, 2009-10
8. Christoph Matheja, 2020-21
9. Wytse Oortwijn, 2019-20
10. Michael Sammler, 2024
11. Alexander J. Summers, 2009-20
12. Caterina Urban, 2015-19
13. Valentin Wüstholz, 2017

SERVICE ACTIVITIES

DEPARTMENT AND UNIVERSITY ADMINISTRATION

Since 2023	Department's representative on ETH's Good Scientific Practice Commission
Since 2022	Member of the ETH Digital Transformation Committee
Since 2009	Delegate for continuing education
2021-2022	Head of a working group to develop a digitalization strategy for the ETH administration
2018-2020	Head of the Institute for Programming Languages and Systems
2017-2019	Director of studies
2017-2019	Member of the department executive board, also 2010-2016 and 2004-2006
2016-2017	Member of the curriculum committee
2016	Member of the evaluation committee of ETH's IT Services Department
2010-2016	Supervisor of the communications group
2013	Member of strategy committee
2009-2013	Member of the teaching commission
2004-2006	Supervisor of the communications group
2004-2005	Computer Science Colloquium (organizer)
2003-2007	Formal Approaches to Software Seminar (co-organizer)

CONFERENCE, WORKSHOP, AND TUTORIAL ORGANIZATIONS

Since 2016	Steering committee member of the *VerifyThis* program verification competition.
Since 2013	Co-organizer of the Workshop on Dependable and Secure *Software Systems*. ETH Zurich, Switzerland.
2024	Co-director of the Marktoberdorf Summer School on Engineering Secure and Dependable Software Systems.
2021-2024	Co-organizer of the Rust Verification workshop at ETAPS.
2023	Co-organizer for Dagstuhl seminar on Theoretical Advances and Emerging Applications in Abstract Interpretation. Dagstuhl, Germany.
2018-2022	Steering committee member of the ETAPS *European Joint Conferences on Theory and Practice of Software*, Germany
2021	Ask-me-anything chair at CAV.
2020	Co-organizer for Dagstuhl seminar on Theoretical Advances and Emerging Applications in Abstract Interpretation. Dagstuhl, Germany.
1999-2019	Steering committee member of the ECOOP Workshop on *Formal Techniques for Java-like Programs*.
2018	Co-director of the Marktoberdorf Summer School on *Engineering Secure and Dependable Software Systems*.
05/2016	Co-organizer for Dagstuhl seminar on Synergies among Testing, Verification, and Repair for Concurrent Programs. Dagstuhl, Germany.
04/2016	Co-organizer of the *VerifyThis* program verification competition at ETAPS.
06/2012	Student volunteers co-chair for *International Conference on Software Engineering* (ICSE)

08/2010 Workshop general chair and competition co-chair for *Verified Software: Theories, Tools and Experiments* (VSTTE)

2008 Steering committee member of the ECOOP *International Workshop on Aliasing, Confinement and Ownership in Object-Oriented Programming*.

04/2007 ETAPS tutorial on *Mobility, Ubiquity, and Security*. Braga, Portugal.

02/2007 Co-organizer of Dagstuhl workshop on Mobility*, Ubiquity, and Security*. Dagstuhl, Germany.

03/2006 *Collaborative Software Engineering*. Symposium of the Swiss Informatics Society (Software Engineering chapter). Zurich, Switzerland.

10/1999 *Colloquium on Programming Languages and Fundamentals of Programming*. Heinsberg, Germany.

EDITORSHIPS

- Member of the editorial board of the *ACM Transactions on Programming Languages and Systems (TOPLAS)*, since 2021.
- Member of the editorial board of the *Journal of Automated Reasoning*, since 2020.
- Member of the editorial board of *Software Testing, Verification and Reliability*, 2007-2025.
- Member of the advisory board of the *Formal Methods* sub-line of LNCS, 2016-2022
- Member of the editorial board of *Science of Computer Programming*, 2013-2022.
- Editor of *Programming Languages and Systems, ESOP 2020*, Springer-Verlag, 2020.
- Co-Editor of *Engineering Secure and Dependable Software Systems*, IOS Press, 2019.
- Co-Editor of *Principled Software Development*, Springer-Verlag, 2018.
- Editor of *31st European Conference on Object-Oriented Programming, ECOOP 2017*, LIPIcs 74, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- Co-Editor of *Verified Software: Theories, Tools, Experiments*, volume 7152 of LNCS, Springer-Verlag, 2012.
- Co-Editor of *Workshop Proceedings - VSTTE 2010*, Technical Report 676, ETH Zurich, 2010
- Editor of *Advanced Lectures on Software Engineering – LASER Summer School 2007/2008*, volume 6029 of LNCS, Springer-Verlag, 2010.
- Co-Editor of special issue on Formal Techniques for Java-like Programs and Aliasing, Confinement and Ownership in object-oriented programming, *Journal of Object Technology (JOT)*, 2009.
- Co-Editor of special issue on Formal Techniques for Java-like Programs, *Journal of Object Technology (JOT)*, 2004.
- Co-Editor of special issue on Formal Techniques for Java-like Programs, *Concurrency and Computation: Practice and Experience*, 2003.

REFEREEING

Program Committees:

- Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), TODO, 2026. (area chair)
- Computer-Aided Verification (CAV), Zagreb, Croatia, 2025. (area chair)
- European Symposium on Programming (ESOP), Hamilton, Canada, 2025. (PC member)
- Programming Language Design and Implementation (PLDI), Copenhagen, Denmark, 2024. (area chair)
- Dafny Workshop, London, UK, 2024. (PC member)
- Principles of Programming Languages (POPL), London, UK, 2024. (PC member)
- Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Paris, France, 2023. (PC Member)
- International Symposium on Software Testing and Analysis (ISSTA), Daejeon, South Korea, 2022. (PC Member)
- Programming Language Design and Implementation (PLDI), San Diego, USA, 2022. (PC member)
- Verified Software: Theories, Tools, and Experiments (VSTTE), New Haven, USA, 2021. (PC member)

- Verified Software: Theories, Tools, and Experiments (VSTTE), Los Angeles, USA, 2020. (PC member)
- Software Engineering and Formal Methods (SEFM), Amsterdam, The Netherlands, 2020. (PC member)
- European Symposium on Programming (ESOP), Dublin, Ireland, 2020. (PC chair)
- Computer-Aided Verification (CAV), New York, USA, 2019. (PC member)
- International Symposium on Software Testing and Analysis (ISSTA), Bejing, China, 2019. (PC member)
- Logic for Programming, Artificial Intelligence and Reasoning (LPAR), Awassa, Ethiopia, 2018. (PC member)
- Foundations of Software Engineering (FSE), Lake Buena Vista, USA, 2018. (PC member)
- Formal Methods (FM), Oxford, UK, 2018. (PC member)
- Formal Methods Industry Day, Oxford, UK, 2018. (PC member)
- Principles of Programming Languages (POPL), Los Angeles, USA, 2018. (PC member)
- Verified Software: Theories, Tools, and Experiments (VSTTE), Heidelberg, Germany, 2017. (PC member)
- Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Vancouver, Canada, 2017. (EPC member)
- European Conference on Object-Oriented Programming (ECOOP), Barcelona, Spain, 2017. (PC chair)
- Principles of Programming Languages (POPL), Paris, France, 2017. (ERC member)
- ACM SIGPLAN International Workshop on the State of the Art in Program Analysis (SOAP), Santa Barbara, USA, 2016. (PC member)
- Formal Methods (FM), Limassol, Cyprus, 2016. (PC member)
- International Conference on Computer Aided Verification (CAV), Toronto, Canada, 2016. (ERC member)
- European Conference on Object-Oriented Programming (ECOOP), Rome, Italy, 2016. (PC member)
- Verification, Model Checking and Abstract Interpretation (VMCAI), St. Petersburg, USA, 2016. (PC member)
- Principles of Programming Languages (POPL), St. Petersburg, USA, 2016. (ERC member)
- Brazilian Symposium on Formal Methods (SBMF), Belo Horizonte, Brazil, 2015. (PC member)
- Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Pittsburgh, USA, 2015. (PC member)
- NASA Formal Methods Symposium (NFM), Pasadena, USA, 2015. (PC member)
- Formal Methods (FM), Oslo, Norway, 2015. (PC member)
- European Conference on Object-Oriented Programming (ECOOP), Prague, Czech Republic, 2015. (PC member)
- International Conference on Modularity, Denver, USA, 2015. (Member of External Review Committee)
- European Symposium on Programming (ESOP), London, United Kingdom, 2015. (PC member)
- Verified Software: Theories, Tools, and Experiments (VSTTE), Vienna, Austria, 2014. (PC member)
- Formal Methods (FM), Singapore, 2014. (PC member)
- International Conference on Formal Engineering Methods (ICFEM), Luxembourg, 2014. (PC member)

- International Conference on Formal Engineering Methods (ICFEM), Queenstown, New Zealand, 2013. (PC member)
- International Workshop on Higher-Order Programming with Effects (HOPE), Boston, USA, 2013. (PC member)
- Verified Software: Theories, Tools, and Experiments (VSTTE), Atherton, USA, 2013. (PC member)
- International Workshop on Foundations of Object-Oriented Languages (FOOL), Tucson, USA, 2012. (PC member)
- Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), Tucson, USA, 2012. (ERC member)
- International Conference on Theoretical Computer Science (TCS), Amsterdam, The Netherlands, 2012. (PC member)
- International Workshop on Intermediate Verification Languages (BOOGIE), Berkeley, USA, 2012. (PC member)
- International Conference on Runtime Verification (RV), Istanbul, Turkey, 2012. (PC member)
- Formal Methods for Open Object-based Distributed Systems (FMOODS/FORTE), Stockholm, Sweden, 2012. (PC member)
- Bytecode Semantics, Verification, Analysis and Transformation (Bytecode), Tallinn, Estonia, 2012. (PC member)
- Verified Software: Theories, Tools, and Experiments (VSTTE), Philadelphia, USA, 2012. (PC co-chair)
- Programming Languages meets Program Verification (PLPV), Philadelphia, USA, 2012. (PC member)
- International Conference on Formal Engineering Methods (ICFEM), Durham, United Kingdom 2011. (PC member)
- Perspectives of System Informatics (PSI), Novosibirsk, Russia, 2011. (PC member)
- Developing Tools as Plug-ins (TOPI), Waikiki, USA, 2011. (PC member)
- International Conference on Compiler Construction (CC), Saarbrücken, Germany, 2011. (PC member)
- Principles of Programming Languages (POPL), Austin, USA, 2011. (PC member)
- Computing: the Australasian Theory Symposium (CATS), Perth, Australia 2011. (PC member)
- Formal Techniques for Java-like Programs （FTfJP), Genova, Italy, 2010. (PC member)
- TOOLS EUROPE, Malaga, Spain, 2010. (PC member)
- FoVeOOS 2010, Paris, France, 2010. (PC member)
- Formal Methods for Open Object-based Distributed Systems (FMOODS/FORTE), Amsterdam, The Netherlands, 2010. (PC member)
- Member, Computing: the Australasian Theory Symposium (CATS), Brisbane, Australia, 2010. (PC member)
- Formal Methods (FM), Eindhoven, The Netherlands, 2009. (PC member)
- Object-Orientated Programming, Systems, Languages and Applications (OOPSLA), Orlando, USA, 2009. (PC member)
- Specification and Verification of Component-Based Systems (SAVCBS), Amsterdam, The Netherlands, 2009. (PC member)
- Theoretical Aspects of Software Engineering (TASE), Tianjin, China, 2009. (PC member)
- TOOLS EUROPE, Zurich, Switzerland, 2009. (PC member)

- Formal Techniques for Distributed Systems (FMOODS/FORTE), Lisbon, Portugal, 2009. (PC member)
- Perspectives of System Informatics (PSI), Novosibirsk, Russia, 2009. (PC member)
- Automatic Program Verification, Rio Cuarto, Argentina, 2009. (PC member)
- Foundations of Object-Oriented Languages (FOOL), Savannah, USA, 2009. (PC member)
- IJCAR International Verification Workshop (VERIFY), Sydney, Australia, 2008. (PC member)
- ECOOP workshop on Aliasing, Confinement and Ownership in Object-Oriented Programming (IWACO), Paphos, Cyprus, 2008. (PC chair)
- ECOOP workshop selection committee, Paphos, Cyprus, 2008. (PC member)
- TOOLS EUROPE, Zurich, Switzerland, 2008. (PC member)
- Formal Methods for Open Object-based Distributed Systems (FMOODS), Oslo, Norway, 2008. (PC member)
- TAP: Tests and Proofs, Prato, Italy, 2008. (PC member)
- European Symposium on Programming (ESOP), Budapest, Hungary, 2008. (PC member)
- GI Tagung Software Engineering, Munich, Germany, 2008. (PC member)
- Specification and Verification of Component-Based Systems, Dubrovnik, Croatia, 2007. (PC member)
- ECOOP Workshop on Aliasing, Confinement and Ownership in Object-Oriented Programming (IWACO), Berlin, Germany, 2007. (PC member)
- TOOLS EUROPE, Zurich, Switzerland, 2007. (PC member)
- 2nd ETAPS Workshop on Bytecode Semantics, Verification, Analysis, and Transformation. Braga, Portugal, 2007. (PC member)
- TAP: Tests And Proofs. Zurich, Switzerland, 2007. (PC member)
- 15th International Conference on Computing, Mexico City, Mexico, 2006. (PC member)
- International Conference on Software and Data Technologies. Setubal, Portugal, 2006. (PC member)
- Perspectives of System Informatics. Novosibirsk, Russia, 2006. (PC member)
- Specification and Verification of Component-Based Systems. Lisbon, Portugal, 2005. (PC member)
- .NET Technologies 2005. Plzen, Czech Republic, 2005. (PC member)
- Specification and Verification of Component-Based Systems. Newport Beach, USA, 2004. (PC member)
- Formal Techniques for Java-like Programs 2004. Oslo, Norway, 2004. (PC member)
- .NET Technologies 2004. Plzen, Czech Republic, 2004. (PC member)
- GPCE Young Researchers Workshop 2003. Erfurt, Germany, 2003. (PC member and panelist)
- Formal Techniques for Java-like Programs 2003. Darmstadt, Germany, 2003. (PC chair)
- Formal Techniques for Java-like Programs 2002. Malaga, Spain, 2002. (PC member)
- Formal Techniques for Java Programs 2001. Budapest, Hungary, 2001. (PC chair)
- Formal Techniques for Java Programs 2000. Cannes, France, 2000. (PC member)
- Formal Techniques for Java Programs '99. Lisbon, Portugal, 1999. (PC member)

Conferences:

OOPSLA 2025, CAV 2021, POPL 2020, OOPSLA 2019, PLDI 2019, RV 2014, ICFP 2013, CAV 2013, APLAS 2010, PPDP 2010, ESOP 2010, POPL 2010, ECOOP 2009, SEFM 008, OOPSLA 2008, FMCO 2008, AMAST 2008, IJCAR 2008, ECOOP 2008, POPL 2008, Dagstuhl post-proceedings 2007, SBMF

2006, OOPSLA 2006, FM 2006, POPL 2006, ECOOP 2005, COCV 2005, POPL 2005, OOPSLA 2004, ESOP 2004, ECOOP 2002

Journals:

- Acta Informatica (Springer) 2008
- Communications of the ACM 2020
- The Computer Journal (Oxford University Press) 2010
- Computing Surveys (ACM), 2010
- Concurrency and Computation - Practice and Experience (John Wiley & Sons)
- Decision Support Systems (Elsevier)
- Formal Aspects of Computing (Springer), 2007, 2011, 2015
- International Journal of Parallel and Distributed Systems and Networks (Acta Press)
- International Journal on Software and Systems Modeling (Springer), 2022
- Journal of Computer and System Sciences (Elsevier), 2010
- Logical Methods in Computer Science, 2013
- The Journal of Logic and Algebraic Programming (Elsevier)
- Journal of Object Technology (JOT) 2004, 2005, 2007, 2012
- Science of Computer Programming (Elsevier) 2007, 2008, 2009, 2010, 2011, 2012, 2014, 2015
- Software: Practice and Experience (Wiley) 2011, 2012
- Software Testing, Verification and Reliability (Wiley) 2007
- Theoretical Computer Science (Elsevier) 2011
- Transactions on Design Automation of Electronic Systems (ACM) 2007
- Transactions on Modularity (Springer) 2015
- Transactions on Programming Languages and Systems (ACM) 2008, 2009, 2010, 2011, 2016, 2018

Thesis Committees:

- L. van den Haak: *Verifying Optimised Parallel Code*, TU Eindhoven, 2025
- F. Linker: *Protocol Design and Analysis in the Symbolic Model*, ETH Zurich, 2025
- G.-V. Saioc: *Program Analysis Techniques for Detecting and Preventing Partial Deadlocks*, Aarhus University, 2025
- B. Gregoire: *Provable Security of Cryptographic Primitives: from Algorithms to Assembly*, habilitation thesis, Université de Nice, 2025
- D. Winterer: *Solidifying Modern SMT Solvers*, ETH Zurich, 2024
- M. Knecht: *A Robust Smart Contract Approach based on Substructural Types and On-Chain Code Verification*, University of Zurich, 2024
- T. Reinhard. *Semi-Automated Modular Formal Verification of Critical Software*, KU Leuven, 2024
- J. Wise DiVincenzo. *Gradual Verification of Recursive Heap Data Structures*, CMU, 2023
- X. Denis. *Deductive Verification of Rust Programs*, Université Paris-Saclay, 2023
- J. Schneider. *Scalable and Trustworthy Monitoring*, ETH Zurich, 2023

- L. Negrini: *A generic framework for multilanguage analysis.* University of Venice, Italy, 2023
- R. Monat. *Static Type and Value Analysis by Abstract Interpretation of Python Programs with Native C Libraries*. Sorbonne, 2021
- S. Buro. *On Multi-Language Semantics*. University of Verona, 2021
- M. Chevalier. *Proving the Security of Software-Intensive Embedded Systems by Abstract Interpretation*. ENS Paris, 2020
- F. Liu: *Safe Initialization of Objects*, EPF Lausanne, 2020
- D. Dimitrov: *Concurrency Analysis for Abstract Data Types*, ETH Zurich, 2020
- J. Haman: *Modular Formal Verification of Total Correctness Properties of Concurrent Imperative Programs*, KU Leuven, 2019
- M. Fäs: *A Dynamic Approach to Deterministic Parallel Programming*, ETH Zurich, 2019
- N. Voirol: *Verifying Functional Programs*, EPF Lausanne, 2019
- A. Dan: *Program Analysis for Weak Memory Models*, ETH Zurich, 2018
- S. Darabi: *Verification of Program Parallelization*, University of Twente, 2018
- T. Morciniec: *A Logic-Based Approach to System Testing*, University of Freiburg, 2017
- G. Ofenbeck: *Generic Programming in Space and Time – Abstractions for High Performance Code Generation*, ETH Zurich, 2017
- A. Fürst: Formal Development of a Train Control System using Event-B. ETH Zurich, 2015
- D. K. Le: *Specification and Verification of Shared-Memory Concurrent Programs.* National University of Singapore, 2014
- N. Polikarpova: Specified and Verified Reusable Components. ETH Zurich, 2014
- L. Rytz: *A Practical Effect System for Scala*. EPF Lausanne, 2013
- Y. Welsch: *Reasoning about Backward Compatibility of Class Libraries*. Technische Universität Kaiserslautern, 2013
- D. Keller: *Paradigms and Tools for Developing Dependable Realtime Software*. ETH Zurich, 2012
- M. Schmalz: *Formalizing the Logic of Event-B – Partial Functions, Definitional Extensions, and Automated Theorem Proving*. ETH Zurich, 2012
- L. Brügger: *A Framework for Modelling and Testing of Security Policies*. ETH Zurich, 2012
- G. Fourny: *More Flexibility in Web Application Development: Time, Space, Scale*. ETH Zurich, 2011
- R. Bardou: *Verification of Pointer Programs Using Regions and Permissions*. Université Paris-Sud, 2011
- S. Balzer: *Rumer: A Programming Language and Modular Verification Technique Based on Relationships*. ETH Zurich, 2011
- N. Matsakis: *Intervals: Data-race-free Parallel Programming*. ETH Zurich, 2011
- C. Gladisch: *Verification-based Software-fault Detection*. Karlsruher Institut für Technologie, 2011
- P. Haller: *Isolated Actors for Race-Free Concurrent Programming*. EPF Lausanne, 2010
- J. Smans: *Specification and Automatic Verification of Frame Properties for Java-like Programs*. Katholieke Universiteit Leuven, 2009
- C. Kunz: *Certificate Translation alongside Program Transformations*. École Nationale Supérieure des Mines de Paris, 2009
- T. Hubert: *Analyse Statique et preuve de Programmes Industriels Critiques*. Université Paris-Sud, 2008
- M. Wahler: *Using Patterns to Develop Consistent Design Constraints*. ETH Zurich, 2008

- Y. Lu: *Object Validity, Effects and Accessibility with Ownership*. University of New South Wales, 2008
- A. Brucker: *An Interactive Proof Environment for Object-Oriented Specifications*. ETH Zurich, 2007
- A. Potanin: *A Practical Approach to Ownership and Confinement in Object-Oriented Programming Languages*. Victoria University of Wellington, 2007
- P. Nienaltowski: *Practical Framework for Contract-Based Concurrent Object-Oriented Programming*. ETH Zurich, 2007
- T. Gedell. Chalmers University of Technology, 2006 (Licentiate committee)
- K. Arnout: *From Patterns to Components*. ETH Zurich, 2004

Grant Proposals:

- Belgian National Fund for Scientific Research (FWO), 2007
- DFG, 2021, 2024, 2025
- Dutch National Science Foundation (NWO), 2007, 2009, 2011
- ETH Innovedum, 2010
- ERC Advanced Grants, 2022
- ERC Consolidator Grants, 2024
- ERC Starting Grants, 2017
- KIT Young Investigator Group, 2014
- Microsoft Software Engineering Innovation Foundation Awards, 2013
- Qatar Foundation, 2009, 2010, 2011
- Royal Society of New Zealand, 2011
- Science Foundation Ireland, 2010
- Swiss National Science Foundation (SNF), 2008, 2010, 2024

Academic Committees outside ETH Zurich:

- Hiring committee TU Vienna, 2024
- Advisory Committee, Department of Computer Science, Eastern Switzerland University of Applied Science, 2021–2024
- Hiring committee, Max Planck Institute for Software Systems, 2021
- Hiring committee Aarhus University, 2018
- Hiring committee EPF Lausanne, 2010, 2011

Award Committees:

- EAPLS Best Dissertation Award 2020
- EAPLS Best Dissertation Award 2019
- Dahl-Nygaard Award Committee, 2019
- Dahl-Nygaard Award Committee, 2018
- SIGPLAN 2013 Outstanding Doctoral Dissertation Award, 2014
- SIGPLAN 2012 Outstanding Doctoral Dissertation Award, 2013

PROFESSIONAL SOCIETIES

- Member of the IFIP Working Group 1.9/2.15 (Verified Software)
- Member of the IFIP Working Group 2.3 (Programming Methodology), vice chair 2019-2024
- Member of the Association Internationale pour les Technologies Objets (AITO)
- Member of the Association for Computing Machinery (ACM)
- Member of the Swiss Informatics Society (SI)
- Board member of the Software Engineering chapter of the Swiss Informatics Society (2005-2007)