PUBLICATIONS

CHRISTOPH M. KIRSCH

ck@cs.uni-salzburg.at
www.cs.uni-salzburg.at/~ck

CONFERENCE AND WORKSHOP PAPERS

[1] T. Hütter et al. "JEDI: These aren't the JSON documents you're looking for..." In: *Proc. International Conference on Management of Data (SIGMOD)*. ACM, 2022. Click here for PDF file.

[2] A. Goel et al. "What We Eval in the Shadows: A Large-Scale Study of Eval in R Programs". In: *Proc. ACM SIGPLAN Conference on Systems, Programming, Languages, and Applications: Software for Humanity (OOPSLA)*. ACM, 2021. Click here for PDF file.

[3] A.S. Abyaneh and C.M. Kirsch. "ASE: A Value Set Decision Procedure for Symbolic Execution". In: *Proc. IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE/ACM, 2021. Click here for PDF file.

[4] A.S. Abyaneh et al. "Selfie: Towards Minimal Symbolic Execution". In: *Online Proc. Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs)*. 2018. Click here for PDF file.

[5] P. Donat-Bouillud and C.M. Kirsch. "Work in Progress: Adaptive Scheduling with Approximate Computing of Audio Graphs". In: *Proc. IEEE Real-Time Systems Symposium (RTSS)*. 2017. Click here for PDF file.

[6] C.M. Kirsch. "Selfie and the Basics". In: *Proc. ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)* ACM, 2017. Click here for PDF file.

[7] A. Haas et al. "Local Linearizability for Concurrent Container-Type Data Structures". In: *Proc. International Conference on Concurrency Theory (CONCUR)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, 6:1–6:15. DOI: 10.4230/LIPIcs.CONCUR.2016.6. Click here for PDF file.

[8] M. Aigner et al. "Fast, Multicore-Scalable, Low-Fragmentation Memory Allocation through Large Virtual Memory and Global Data Structures". In: *Proc. ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. ACM, 2015. Click here for PDF file.

[9] E. Pereira et al. "A Runtime System for Logical-Space Programming". In: *Proc. Workshop on the Swarm at the Edge of the Cloud (SWEC)*. 2015. Click here for PDF file.

[10] M. Dodds, A. Haas, and C.M. Kirsch. "A Scalable, Correct Time-Stamped Stack". In: *Proc. Symposium on Principles of Programming Languages (POPL)*. ACM, 2015. Click here for PDF file.

[11] M. Aigner et al. "ACDC-JS: Explorative Benchmarking of JavaScript Memory Management". In: *Proc. Dynamic Languages Symposium (DLS)*. ACM, 2014. Click here for PDF file.

[12] C. Krainer and C.M. Kirsch. "Cyber-Physical Cloud Computing Implemented as PaaS". In: *Proc. Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems (CyPhy)*. ACM, 2014. Click here for PDF file.

[13] C.M. Kirsch, M. Lippautz, and H. Payer. "Fast and Scalable, Lock-free k-FIFO Queues". In: *Proc. International Conference on Parallel Computing Technologies (PaCT)*. LNCS. Springer, 2013. Click here for PDF file.

[14] M. Aigner et al. "Analysis of Portfolio-Style Parallel SAT Solving on Current Multi-Core Architectures". In: *Proc. Workshop on Pragmatics of SAT (PoS)*. EPiC. EasyChair, 2013. Click here for PDF file.

[15] M. Aigner and C.M. Kirsch. "ACDC: Towards a Universal Mutator for Benchmarking Heap Management Systems". In: *Proc. International Symposium on Memory Management (ISMM)*. ACM, 2013. Click here for PDF file.

[16] E. Pereira et al. "Modeling and Controlling the Structure of Heterogeneous Mobile Robotic Systems: A BigActor Approach". In: *International Systems Conference (SysCon)*. IEEE, 2013. Click here for PDF file.

[17] E. Pereira et al. "BigActors - A Model for Structure-aware Computation". In: *Proc. International Conference on Cyber-Physical Systems (ICCPS)*. ACM, 2013. Click here for PDF file.

[18] T.A. Henzinger et al. “Quantitative Relaxation of Concurrent Data Structures”. In: *Proc. Symposium on Principles of Programming Languages (POPL)*. ACM, 2013. Click here for PDF file.

[19] S.S. Craciunas and C.M. Kirsch. “The Power of Isolation”. In: *Proc. International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE, 2012. Click here for PDF file.

[20] A. Haas et al. “How FIFO is Your Concurrent FIFO Queue?” In: *Proc. OOPSLA Workshop on Relaxing Synchronization for Multicore and Manycore Scalability (RACES)*. 2012. Click here for PDF file.

[21] C.M. Kirsch et al. “Performance, Scalability, and Semantics of Concurrent FIFO Queues”. In: *Proc. International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. LNCS. Springer, 2012. Click here for PDF file.

[22] C.M. Kirsch and H. Payer. “Incorrect Systems: It’s not the Problem, It’s the Solution”. In: *Proc. Design Automation Conference (DAC)*. ACM, 2012. Click here for PDF file.

[23] C.M. Kirsch et al. “Cyber-Physical Cloud Computing: The Binding and Migration Problem”. In: *Proc. International Conference on Design, Automation and Test in Europe (DATE)*. 2012. Click here for PDF file.

[24] C.M. Kirsch, H. Payer, and H. Röck. “Hierarchical PLABs, CLABs, TLABs in Hotspot”. In: *Proc. International Conference on Systems (ICONS)*. 2012. Click here for PDF file.

[25] H. Chen et al. “Cloud Computing on Wings: Applications to Air Quality”. In: *Proc. American Astronautical Society Guidance and Control Conference (AASGNC)*. AAS, 2012. Click here for PDF file.

[26] C.M. Kirsch et al. “Runtime Programming through Model-Preserving, Scalable Runtime Patches”. In: *Proc. International Conference on Application of Concurrency to System Design (ACSD)*. IEEE, 2011, pp. 77–86. Click here for PDF file.

[27] C.M. Kirsch et al. “Brief Announcement: Scalability versus Semantics of Concurrent FIFO Queues”. In: *Proc. Symposium on Principles of Distributed Computing (PODC)*. ACM, 2011, pp. 331–332. Click here for PDF file.

[28] M. Aigner et al. “Short-term Memory for Self-collecting Mutators”. In: *Proc. International Symposium on Memory Management (ISMM)*. ACM, 2011. Click here for PDF file.

[29] C.M. Kirsch et al. “Runtime Programming through Model-Preserving, Scalable Runtime Patches”. In: *Proc. International Workshop on Formal Aspects of Component Software (FACS), Doctoral Track*. Vol. 6921. LNCS. Springer, 2010, pp. 290–294. Click here for PDF file.

[30] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. “Power-aware Temporal Isolation with Variable-Bandwidth Servers”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. ACM, 2010. Click here for PDF file.

[31] S.S. Craciunas et al. “Information-Acquisition-as-a-Service for Cyber-Physical Cloud Computing”. In: *Proc. Workshop on Hot Topics in Cloud Computing (HotCloud)*. USENIX, 2010. Click here for PDF file.

[32] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. “Response Time versus Utilization in Scheduler Overhead Accounting”. In: *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2010. Click here for PDF file.

[33] T.A. Henzinger et al. “Distributed, Modular HTL”. In: *Proc. Real-Time Systems Symposium (RTSS)*. IEEE, 2009. Click here for PDF file.

[34] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. “A Workload-oriented Programming Model for Temporal Isolation with VBS”. In: *Online Proc. Workshop on Reconciling Performance with Predictability (RePP)*. 2009. Click here for PDF file.

[35] H. Röck et al. “Avoiding Unbounded Priority Inversion in Barrier Protocols Using Gang Priority Management”. In: *Proc. International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*. ACM, 2009. Click here for PDF file.

[36] S.S. Craciunas et al. “Programmable Temporal Isolation through Variable-Bandwidth Servers”. In: *Proc. Symposium on Industrial Embedded Systems (SIES)*. IEEE, 2009. Click here for PDF file.

[37] K. Hedrick et al. “CSL: A Language to Specify and Re-Specify Mobile Sensor Network Behaviors”. In: *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2009. Click here for PDF file.

[38] S.S. Craciunas et al. “Programmable Temporal Isolation in Real-Time and Embedded Execution Environments”. In: *Proc. Workshop on Isolation and Integration in Embedded Systems (IIES)*. ACM, 2009. Click here for PDF file.

[39] H. Payer et al. “Combo Drive: Optimizing Cost and Performance in a Heterogeneous Storage Device”. In: *Proc. Workshop on Integrating Solid-state Memory into the Storage Hierarchy (WISH)*. 2009. Click here for PDF file.

[40] S.S. Craciunas et al. “The JAviator: A High-Payload Quadrotor UAV with High-Level Programming Capabilities”. In: *Proc. AIAA Guidance, Navigation and Control Conference (GNC)*. 2008. Click here for PDF file.

[41] S.S. Craciunas et al. “A Compacting Real-Time Memory Management System”. In: *Proc. USENIX Annual Technical Conference*. 2008. Click here for PDF file.

[42] C.M. Kirsch, L. Lopes, and E.R.B. Marques. “Semantics-Preserving and Incremental Runtime Patching of Real-Time Programs”. In: *Proc. Workshop on Adaptive and Reconfigurable Embedded Systems (APRES)*. 2008. Click here for PDF file.

[43] K. Chatterjee et al. “Logical Reliability of Interacting Real-Time Tasks”. In: *Proc. International Conference on Design, Automation and Test in Europe (DATE)*. 2008. Click here for PDF file.

[44] C.M. Kirsch and R. Wilhelm. “Grand Challenges in Embedded Software”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. ACM, 2007, pp. 2–6.

[45] A. Ghosal et al. “Separate Compilation of Hierarchical Real-Time Programs into Linear-Bounded Embedded Machine Code”. In: *Online Proc. Workshop on Automatic Program Generation for Embedded Systems (APGES)*. 2007. Click here for PDF file.

[46] J. Auerbach et al. “Java Takes Flight: Time-portable Real-time Programming with Exotasks”. In: *Proc. ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*. ACM, 2007. Click here for PDF file.

[47] A. Ghosal et al. “A Hierarchical Coordination Language for Interacting Real-Time Tasks”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. ACM, 2006. Click here for PDF file.

[48] C.M. Kirsch, M.A.A. Sanvido, and T.A. Henzinger. “A Programmable Microkernel for Real-Time Systems”. In: *Proc. ACM/USENIX Conference on Virtual Execution Environments (VEE)*. ACM, 2005. Click here for PDF file.

[49] T.A. Henzinger, C.M. Kirsch, and S. Matic. “Composable Code Generation for Distributed Giotto”. In: *Proc. ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*. ACM, 2005. Click here for PDF file.

[50] C.M. Kirsch. “Threading by Appointment”. In: *Proc. Monterey Workshop*. CRC Press, 2004. Click here for PDF file.

[51] T.A. Henzinger and C.M. Kirsch. “A Typed Assembly Language for Real-Time Programs”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. ACM, 2004, pp. 104–113. Click here for PDF file.

[52] A. Ghosal et al. “Event-driven Programming with Logical Execution Times”. In: *Proc. International Workshop on Hybrid Systems: Computation and Control (HSCC)*. Vol. 2993. LNCS. Springer, 2004, pp. 357–371. Click here for PDF file.

[53] T.A. Henzinger, C.M. Kirsch, and S. Matic. “Schedule-Carrying Code”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. Vol. 2855. LNCS. Springer, 2003, pp. 241–256. Click here for PDF file.

[54] T.A. Henzinger and C.M. Kirsch. “The Embedded Machine: Predictable, Portable Real-Time Code”. In: *Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. ACM, 2002, pp. 315–326. Click here for PDF file.

[55] T.A. Henzinger et al. “Time Safety Checking for Embedded Programs”. In: *Proc. International Workshop on Embedded Software (EMSOFT)*. Vol. 2491. LNCS. Springer, 2002, pp. 76–92. Click here for PDF file.

[56] C.M. Kirsch et al. “A Giotto-Based Helicopter Control System”. In: *Proc. International Workshop on Embedded Software (EMSOFT)*. Vol. 2491. LNCS. Springer, 2002, pp. 46–60. Click here for PDF file.

[57] T.A. Henzinger, B. Horowitz, and C.M. Kirsch. “Giotto: A Time-triggered Language for Embedded Programming”. In: *Proc. International Workshop on Embedded Software (EMSOFT)*. Vol. 2211. LNCS. Springer, 2001, pp. 166–184. Click here for PDF file.

[58] T.A. Henzinger, B. Horowitz, and C.M. Kirsch. “Embedded Control Systems Development with Giotto”. In: *Proc. ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES)*. ACM, 2001. Click here for PDF file.

[59] T.B. Brown et al. “A Reusable and Platform-Independent Framework for Distributed Control Systems”. In: *Proc. Digital Avionics Systems Conference (DASC)*. IEEE, 2001. Click here for PDF file.

[60] H. Ganzinger, C. Meyer, and M. Veanes. “The Two-Variable Guarded Fragment with Transitive Relations”. In: *Proc. Symposium on Logic in Computer Science (LICS)*. IEEE, 1999. Click here for PDF file.

[61] H. Ganzinger et al. “A Resolution-Based Decision Procedure for Extensions of K4”. In: *Proc. Workshop on Advances in Modal Logic (AiML)*. Vol. 2. Lecture Notes. CSLI Publications, Stanford, CA, 1998. Click here for PDF file.

[62] F. Jacquemard, C. Meyer, and C. Weidenbach. “Unification in Extensions of Shallow Equational Theories”. In: *Proc. International Conference on Rewriting Techniques and Applications (RTA)*. Vol. 1379. LNCS. Springer, 1998. Click here for PDF file.

[63] H. Ganzinger, C. Meyer, and C. Weidenbach. “Soft Typing for Ordered Resolution”. In: *Proc. International Conference on Automated Deduction (CADE)*. Vol. 1249. LNCS. Springer, 1997. Click here for PDF file.

[64] P. Graf and C. Meyer. “Advanced Indexing Operations on Substitution Trees”. In: *Proc. International Conference on Automated Deduction (CADE)*. Vol. 1104. LNCS. Springer, 1996. Click here for PDF file.

JOURNAL PAPERS

[1] R. Ambros et al. “Two-Year Progress of Pilot Research Activities in Teaching Digital Thinking Project (TDT)”. In: *Zeitschrift für Hochschulentwicklung* 18 (2023). Ed. by Martin Ebner and Charlotte Zwiauer, pp. 117–136. DOI: 10.3217/zfhe-SH-HL/07. URL: https://zfhe.at/index.php/zfhe/article/view/1755. Click here for PDF file.

[2] J. Huang, C.M. Kirsch, and R. Sengupta. “Cloud Computing in Space”. In: *INFORMS Journal on Computing* 27.4 (2015), pp. 704–717. DOI: 10.1287/ijoc.2015.0652.

[3] S.S. Craciunas et al. “Temporal Isolation in Real-Time Systems: The VBS Approach”. In: *Software Tools for Technology Transfer (STTT)* 15.3 (2013), pp. 189–209. Click here for PDF file.

[4] A. Ghosal et al. “Separate Compilation of Hierarchical Real-Time Programs into Linear-bounded Embedded Machine Code”. In: *Science of Computer Programming* 77.2 (2012), pp. 96–112.

[5] J. Auerbach et al. “Low-Latency Time-portable Real-time Programming with Exotasks”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 8.2 (Jan. 2009), pp. 1–48. Click here for PDF file.

[6] S.S. Craciunas, C.M. Kirsch, and H. Röck. “I/O Resource Management through System Call Scheduling”. In: *ACM Operating Systems Review (OSR), Special Issue on Research and Developments in the Linux Kernel* (July 2008). Click here for PDF file.

[7] T.A. Henzinger and C.M. Kirsch. “The Embedded Machine: Predictable, Portable Real-Time Code”. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 29.6 (Oct. 2007), pp. 33–61. Click here for PDF file.

[8] T.A. Henzinger, B. Horowitz, and C.M. Kirsch. “Giotto: A Time-triggered Language for Embedded Programming”. In: *Proceedings of the IEEE* 91.1 (Jan. 2003), pp. 84–99. Click here for PDF file.

[9] T.A. Henzinger et al. “From Control Models to Real-Time Code using Giotto”. In: *IEEE Control Systems Magazine (CSM)* 23.1 (Feb. 2003), pp. 50–64. Click here for PDF file.

INVITED PAPERS AND ABSTRACTS

[1] C.M. Kirsch. “On the Self in Selfie (Invited Talk)”. In: *Proc. ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages (VMIL)*. ACM, 2018. Click here for PDF file.

[2] C.M. Kirsch. “From Logical Execution Time to Principled Systems Engineering”. In: *Dagstuhl Reports* 8.2 (2018). Ed. by Rolf Ernst et al., p. 133. ISSN: 2192-5283. DOI: 10.4230/DagRep.8.2.122. URL: http://drops.dagstuhl.de/opus/volltexte/2018/9293. Click here for PDF file.

[3] A.S. Abyaneh and C.M. Kirsch. “You can program what you want but you cannot compute what you want”. In: *Edward A. Lee Festschrift*. Vol. 10760. LNCS. Springer, 2018. Click here for PDF file.

[4] A. Haas et al. “Scal: A Benchmarking Suite for Concurrent Data Structures”. In: *Proc. International Conference on Networked Systems (NETYS)*. LNCS. Springer, 2015. Click here for PDF file.

[5] A. Haas et al. “Distributed Queues in Shared Memory—Multicore Performance and Scalability through Quantitative Relaxation”. In: *Proc. International Conference on Computing Frontiers*. ACM, 2013. Click here for PDF file.

[6] D.F. Bacon et al. “High-Level Real-Time Programming in Java”. In: *Proc. International Conference on Embedded Software (EMSOFT)*. ACM, 2005. Click here for PDF file.

[7] C.M. Kirsch. “Principles of Real-Time Programming”. In: *Proc. International Workshop on Embedded Software (EMSOFT)*. Vol. 2491. LNCS. Springer, 2002, pp. 61–75. Click here for PDF file.

PROCEEDINGS AND SPECIAL ISSUES

[1] Christoph M. Kirsch and Ben L. Titzer, eds. *Proc. ACM SIGPLAN International Symposium on Memory Management (ISMM)*. Barcelona, Spain: ACM, 2017.

[2] “Introduction to Special Section on Probabilistic Embedded Computing”. In: *ACM Trans. Embed. Comput. Syst.* 12.2s (May 2013). Ed. by C.M. Kirsch and V. Mooney, 86:1–86:2. ISSN: 1539-9087. DOI: 10.1145/2465787.2465788. URL: http://doi.acm.org/10.1145/2465787.2465788. Click here for PDF file.

[3] Gernot Heiser and Christoph M. Kirsch, eds. *ACM European Conference on Computer Systems, EuroSys 2011, Salzburg, Austria, April 10 - 13, 2011, Proceedings*. New York, NY, USA: ACM, 2011.

[4] Christoph M. Kirsch and Mahmut T. Kandemir, eds. *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems, LCTES 2009, Dublin, Ireland, June 19 - 20, 2009, Proceedings*. New York, NY, USA: ACM, 2009.

[5] Christoph M. Kirsch and Reinhard Wilhelm, eds. *ACM & IEEE International Conference on Embedded Software, EMSOFT 2007, Salzburg, Austria, September 30 - October, 3, 2007, Proceedings*. New York, NY, USA: ACM, 2007.

[6] Thomas A. Henzinger and Christoph M. Kirsch, eds. *Embedded Software, First International Workshop, EMSOFT 2001, Tahoe City, CA, USA, October, 8-10, 2001, Proceedings*. Vol. 2211. LNCS. Springer, 2001.

BOOK CHAPTERS

[1] C.M. Kirsch and A. Sokolova. “The Logical Execution Time Paradigm”. In: *Advances in Real-Time Systems*. 2012, pp. 103–120.

[2] “Handbook of Real-Time and Embedded Systems”. In: ed. by I. Lee, J. Leung, and S.H. Son. CRC Press, 2007. Chap. The Evolution of Real-Time Programming.

[3] “Software-Enabled Control: Information Technology for Dynamical Systems”. In: ed. by T. Samad and G. Balas. IEEE Press and Wiley-Interscience, 2003. Chap. "Embedded Control Systems Development with Giotto".

SYSTEM PAPERS

[1] R. Alur et al. “jMocha: A Model Checking Tool that Exploits Design Structure”. In: *Proc. International Conference on Software Engineering (ICSE)*. 2001. Click here for PDF file.

[2] C. Weidenbach et al. “SPASS v0.77”. In: *Journal of Automated Reasoning* 21.1 (1998).

SHORT TALKS AND POSTERS

[1] M. Dodds, A. Haas, and C.M. Kirsch. *Fairness vs. Linearizability in a Concurrent FIFO Queue*. Short Talk at the Joint Euro-TM/MEDIAN Workshop on Dependable Multicore and Transactional Memory Systems (DMTM). 2014.

[2] E. Pereira et al. *A Networked Robotic System and its Use in an Oil Spill Monitoring Exercise*. Short Talk at the International Workshop on the Swarm at the Edge of the Cloud. 2013. Click here for PDF file.

[3] J. Huang, C.M. Kirsch, and R. Sengupta. *Scalability of Vehicle Networks through Vehicle Virtualization*. Poster at the International Workshop on the Swarm at the Edge of the Cloud. 2013. Click here for PDF file.

[4] H. Payer, H. Röck, and C.M. Kirsch. *Get What You Pay For: Providing Performance Isolation in Virtualized Execution Environments*. Poster at the ACM SIGOPS European Systems Conference (EuroSys). 2010. Click here for PDF file.

[5] S.S. Craciunas et al. *Everyone Virtualizes Everything But Time*. Poster at the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). 2009. Click here for PDF file.

TECHNICAL REPORTS

[1] M. Dodds, A. Haas, and C.M. Kirsch. *Fast Concurrent Data-Structures Through Explicit Timestamping*. Tech. rep. 2014-03. Department of Computer Science, University of Salzburg, Feb. 2014. Click here for PDF file.

[2] C.M. Kirsch, M. Lippautz, and H. Payer. *Fast and Scalable k-FIFO Queues*. Tech. rep. 2012-04. Department of Computer Science, University of Salzburg, June 2012. Click here for PDF file.

[3] T.A. Henzinger et al. *Quantitative Relaxation of Concurrent Data Structures*. Tech. rep. 2012-03. Department of Computer Science, University of Salzburg, May 2012. Click here for PDF file.

[4] C.M. Kirsch et al. *Performance, Scalability, and Semantics of Concurrent FIFO Queues*. Tech. rep. 2011-03. Department of Computer Science, University of Salzburg, Sept. 2011. Click here for PDF file.

[5] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. *The Power of Isolation*. Tech. rep. 2011-02. Department of Computer Science, University of Salzburg, July 2011. Click here for PDF file.

[6] C.M. Kirsch et al. *Runtime Programming through Model-Preserving, Scalable Runtime Patches*. Tech. rep. 2010-08. Department of Computer Science, University of Salzburg, Dec. 2010. Click here for PDF file.

[7] C.M. Kirsch, H. Payer, and H. Röck. *Scal☠: Non-Linearizable Computing Breaks the Scalability Barrier*. Tech. rep. 2010-07. Department of Computer Science, University of Salzburg, Nov. 2010. Click here for PDF file.

[8] M. Aigner et al. *Short-term Memory for Self-collecting Mutators - Revised Version*. Tech. rep. 2010-06. Department of Computer Science, University of Salzburg, Oct. 2010. Click here for PDF file.

[9] M. Aigner et al. *Short-term Memory for Self-collecting Mutators*. Tech. rep. 2010-03. Department of Computer Science, University of Salzburg, Apr. 2010. Click here for PDF file.

[10] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. *Power-aware Temporal Isolation with Variable-Bandwidth Servers*. Tech. rep. 2010-02. Department of Computer Science, University of Salzburg, Apr. 2010. Click here for PDF file.

[11] S.S. Craciunas, C.M. Kirsch, and A. Sokolova. *Response Time versus Utilization in Scheduler Overhead Accounting*. Tech. rep. 2009-03. Department of Computer Science, University of Salzburg, Aug. 2009. Click here for PDF file.

[12] S.S. Craciunas et al. *Concurrency and Scalability versus Fragmentation and Compaction with Compact-fit*. Tech. rep. 2009-02. Department of Computer Science, University of Salzburg, Apr. 2009. Click here for PDF file.

[13] S.S. Craciunas et al. *Real-Time Scheduling for Workload-oriented Programming*. Tech. rep. 2008-02. Department of Computer Science, University of Salzburg, Sept. 2008. Click here for PDF file.

[14] S.S. Craciunas, C.M. Kirsch, and H. Röck. *Shaping Process Semantics*. Tech. rep. 2007-01. Department of Computer Science, University of Salzburg, Apr. 2007. Click here for PDF file.

[15] M. Törngren et al. *Co-Design of Control Systems and Their Real-Time Implementation — A Tool Survey*. Tech. rep. KTH/MMK/R-06/11-SE. Stockholm, Sweden: Department of Machine Design, Royal Institute of Technology (KTH), Sept. 2006. Click here for PDF file.

[16] A. Ghosal et al. *Hierarchical Timing Language*. Tech. rep. UCB/EECS-2006-79. EECS Department, University of California, Berkeley, May 2006. Click here for PDF file.

[17] C.M. Kirsch and H. Röck. *Traffic Shaping System Calls Using Threading by Appointment*. Tech. rep. T009. Department of Computer Science, University of Salzburg, Aug. 2005. Click here for PDF file.

[18] C.M. Kirsch. *Threading by Appointment*. Tech. rep. T003. Department of Computer Science, University of Salzburg, Sept. 2004. Click here for PDF file.

[19] C.M. Kirsch, T.A. Henzinger, and M.A.A. Sanvido. *A Programmable Microkernel for Real-Time Systems*. Tech. rep. UCB//CSD-03-1250. California: University of California at Berkeley, June 2003. Click here for PDF file.

[20] T.A. Henzinger, C.M. Kirsch, and S. Matic. *Schedule-Carrying Code*. Tech. rep. UCB//CSD-03-1230. California: University of California at Berkeley, Feb. 2003. Click here for PDF file.

[21] C.M. Kirsch. *The Embedded Machine*. Tech. rep. UCB//CSD-01-1137. California: University of California at Berkeley, Mar. 2001. Click here for PDF file.

[22] T.A. Henzinger, B. Horowitz, and C.M. Kirsch. *Giotto: A Time-Triggered Language for Embedded Programming*. Tech. rep. UCB//CSD-00-1121. California: University of California at Berkeley, 2000. Click here for PDF file.

[23] F. Jacquemard, C. Meyer, and C. Weidenbach. *Unification in Extensions of Shallow Equational Theories*. Tech. rep. MPI-I-98-2-002. Saarbrücken, Germany: Max Planck Institute for Computer Science, Jan. 1998. Click here for PDF file.

[24] P. Graf and C. Meyer. *Extended Path-Indexing*. Tech. rep. MPI-I-93-253. Saarbrücken, Germany: Max Planck Institute for Computer Science, Dec. 1993. Click here for PDF file.

THESES

[1] C. Meyer. "Soft Typing for Clausal Inference Systems". PhD Thesis. Saarbrücken, Germany: Saarland University, 1999. Click here for PDF file.

[2] C. Meyer. "Parallel Unit Resulting Resolution". Master's Thesis. Saarbrücken, Germany: Saarland University, 1996. Click here for PDF file.

SOFTWARE

1. Unicorn: Symbolic Execution, Bounded Model Checking, and Code Optimization of C Programs on Quantum Computers, with Daniel Kocher, Stefanie Lei Muroya, and Michael Starzinger. Web: `https://github.com/cksystemsgroup/unicorn`

2. The Selfie Project: A Fully Self-Referential System of a Tiny C Compiler, Tiny MIPS Emulator, and Tiny MIPS Hypervisor, with Martin Aigner, Christian Barthel, Michael Lippautz, and Simone Oblasser. Web: `https://selfie.cs.uni-salzburg.at`

3. The Scalloc Project: A Fast, Multicore-Scalable, Low-Memory-Overhead Allocator, with Martin Aigner, Michael Lippautz, and Ana Sokolova. Web: `https://scalloc.cs.uni-salzburg.at`

4. The ACDC Project: Towards a Universal Mutator for Benchmarking Heap Management Systems, with Martin Aigner. Web: `https://acdc.cs.uni-salzburg.at`

5. The Short-term Memory Project: Short-term Memory for Self-collecting Mutators, with Martin Aigner, Andreas Haas, Michael Lippautz, Ana Sokolova, Stephanie Stroka, and Andreas Unterweger. Web: `https://libscm.cs.uni-salzburg.at`

6. The Scal Project: High-Performance, Multicore-Scalable Data Structures, with Andreas Haas, Thomas A. Henzinger, Michael Lippautz, Hannes Payer, Ali Sezgin, and Ana Sokolova. Web: `https://scal.cs.uni-salzburg.at`

7. The Tiptoe Project: A Compositional Real-Time Operating System, with Silviu Craciunas, Hannes Payer, Harald Röck, Ana Sokolova, and Horst Stadler. Web: `https://tiptoe.cs.uni-salzburg.at`

8. The Jarol Project: A Java Infrastructure for Control Systems, with Bernhard Kast, Eduardo Marques, and Rainer Trummer. Web: `https://jarol.cs.uni-salzburg.at`

9. The JAviator Project: Quadrotor UAV Software Entirely Written in Java, with Joshua Auerbach, David Bacon, Harald Röck, and Rainer Trummer. Web: `https://javiator.cs.uni-salzburg.at`

10. The TAP Project: Concurrent Programming with Threading by Appointment, with Silviu Craciunas and Harald Röck. Web: `https://tap.cs.uni-salzburg.at`

11. The HTL Project: Compositional Real-Time Programming in a Hierarchical Timing Language, with Arkadeb Ghosal, Thomas A. Henzinger, Daniel Iercan, and Alberto L. Sangiovanni-Vincentelli. Web: `https://htl.cs.uni-salzburg.at`

12. Giotto: An Embedded Programming Language, Compiler, and Runtime System for Distributed Control Systems, with Arkadeb Ghosal, Thomas A. Henzinger, Slobodan Matic, and Marco A.A. Sanvido. Web: `http://embedded.eecs.berkeley.edu/giotto`

13. jMocha: A Model Checking Tool that Exploits Design Structure, with Rajeev Alur, Luca de Alfaro, Radu Grosu, Thomas A. Henzinger, Minsu Kang, Rupak Majumdar, Freddy Mang, and Bow-Yaw Wang. Web: `http://embedded.eecs.berkeley.edu/research/mocha`

14. SPASS v0.77: An Automated Theorem Prover for First-Order Logic with Equality, with Christoph Weidenbach, Christian Cohrs, Thorsten Engel, and Enno Keen. Web: `http://spass.mpi-sb.mpg.de`

15. PURR: Parallel Unit Resulting Resolution, a concurrent first-order theorem prover with advanced indexing operations, see Master's Thesis.

16. ACID: A Collection of Indexing Data Structures, implemented in C and Prolog, with Peter Graf.

HARDWARE

1. The JAviator: A Quadrotor Helicopter and Software Laboratory for Time-Portable Java Programming, with Rainer Trummer. Web: `https://javiator.cs.uni-salzburg.at`