Curriculum Vitae

Personal Data

Name	Adrian Prantl
Residence	San Jose, CA, USA

Employment

2013 – ongoing
Compiler Engineer, **Apple**
Developer Tools
Responsible for the debugger support in all Apple compilers. This includes the LLVM-based Clang compiler for C, C++, and Objective-C and the Swift compiler.
Implemented debug info support in the Swift compiler, module debug info in Clang, and contributed many improvements to the debug info handling and DWARF emission code in LLVM. Very engaged with the LLVM and Swift open source community and presented and organized sessions at the annual LLVM developer meeting. Member of the DWARF standards committee for debug information.
I'm also pushing for the adoption of Clang modules to replace textual inclusion of headers, for faster compile times and a better debugging experience.
Very interested in better automated testing of compilers and helped with setting up the Green Dragon LLVM continuous integration system, and CI systems for the Swift compiler.

2010 – 2013
Postdoctoral Researcher, **LLNL**
Center of Applied Scientific Computing, LLNL
Worked on programming language interoperability tools (Babel, Braid) and term-based program transformation frameworks.

2006 – 2010
Research Assistant, **Vienna University of Technology**
Institute of Computer Languages
Worked on high-level static analysis for WCET applications (Worst-Case Execution Time analysis for hard real-time systems).

2005 – 2006
Student intern, **Vienna University of Technology**
Christian Doppler Lab. *"Compilation Techniques for Embedded Processors"*
Design and implementation of a compiler infrastructure for the 24-bit *Ilvy* VLIW Processor, including ports of GCC, GNU Binutils and GDB.

2006
Contractor, **OnDemand Microelectronics**
An implementation of the compiler for the 32-Bit *Chili* VLIW Processor, based on my previous design.

2004 – 2006
Student Assistant, **Vienna University of Technology**
TA for the courses *Functional Programming* and *Compiler Construction.*

Education

2006 – 2010	Computer Science at Vienna University of Technology Graduated with honors as “Dr. techn.” (PhD) Thesis topic: *High-level compiler support for timing analysis* Supervisor: Prof. Jens Knoop
2000 – 2006	Computer Science at Vienna University of Technology Graduated with honors as “Dipl.-Ing.” (M. Sc.) Master’s Thesis: *Creating a GCC back end for a VLIW-architecture* Supervisor: Prof. Andreas Krall
1992 – 2000	Bundesgymnasium Leoben I (Secondary school) Graduated with honors (“Matura”)

Publications

[1] Daniel G. Chavarría-Miranda, Ajay Panyala, Wenjing Ma, Adrian Prantl, and Sriram Krishnamoorthy. Global transformations for legacy parallel applications via structural analysis and rewriting. *Parallel Computing*, 43:1–26, 2015. doi: 10.1016/j.parco.2015.01.001. URL `http://dx.doi.org/10.1016/j.parco.2015.01.001`.

[2] Markus Schordan and Adrian Prantl. Combining static analysis and state transition graphs for verification of event-condition-action systems in the RERS 2012 and 2013 challenges. *STTT*, 16(5):493–505, 2014. doi: 10.1007/s10009-014-0338-x. URL `http://dx.doi.org/10.1007/s10009-014-0338-x`.

[3] Thomas G. W. Epperly, Gary Kumfert, Tamara Dahlgren, Dietmar Ebner, Jim Leek, Adrian Prantl, and Scott Kohn. High-performance language interoperability for scientific computing through Babel. *International Journal of High Performance Computing Applications*, 26:260–274, August 2012.

[4] Adrian Prantl, Thomas Epperly, Shams Imam, and Vivek Sarkar. Interfacing Chapel with traditional HPC programming languages. In *PGAS 2011: Fifth Conference on Partitioned Global Address Space Programming Models*, Galveston TX, October 2011.

[5] Adrian Prantl, Thomas G. W. Epperly, and Shams Imam. Poster: connecting PGAS and traditional HPC languages. In Scott Lathrop, Jim Costa, and William Kramer, editors, *SC Companion*, pages 69–70. ACM, 2011. ISBN 978-1-4503-1030-7.

[6] Raimund Kirner, Jens Knoop, Adrian Prantl, Markus Schordan, and Albrecht Kadlec. Beyond loop bounds: comparing annotation languages for worst-case execution time analysis. *Software and System Modeling*, 10(3):411–437, 2011.

[7] Gergö Barany and Adrian Prantl. Source-level support for timing analysis. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA (2)*, volume 6416 of *Lecture Notes in Computer Science*, pages 434–448. Springer, 2010. ISBN 978-3-642-16560-3.

[8] Adrian Prantl. *High-level compiler support for timing analysis.* Phd thesis, Vienna University of Technology, May 2010.

[9] Raimund Kirner, Peter P. Puschner, and Adrian Prantl. Transforming flow information during code optimization for timing analysis. *Real-Time Systems*, 45(1-2):72–105, 2010.

[10] Adrian Prantl. Towards a static profiler. In *15. Kolloquium Programmiersprachen und Grundlagen der Programmierung (KPS'09)*, page 230, Maria Taferl, Austria, Oct. 2009.

[11] Adrian Prantl, Jens Knoop, Raimund Kirner, Albrecht Kadlec, and Markus Schordan. From trusted annotations to verified knowledge. In *Proceedings of the 9th International Workshop on Worst-Case Execution Time Analysis (WCET 2009)*, pages 39–49, Dublin, Ireland, June 2009. Österreichische Computer Gesellschaft. ISBN: 978-3-85403-252-6.

[12] Adrian Prantl, Jens Knoop, Markus Schordan, and Markus Triska. Constraint solving for high-level wcet analysis. In *The 18th Workshop on Logic-based methods in Programming Environments (WLPE 2008)*, pages 77–89, Udine, Italy, December 2008.

[13] Adrian Prantl, Markus Schordan, and Jens Knoop. TuBound – A Conceptually New Tool for Worst-Case Execution Time Analysis. In *8th International Workshop on Worst-Case Execution Time Analysis (WCET 2008)*, pages 141–148, Prague, Czech Republic, 2008. Österreichische Computer Gesellschaft. ISBN: 978-3-85403-237-3.

[14] Niklas Holsti, Jan Gustafsson, Guillem Bernat (eds.), Clément Ballabriga, Armelle Bonenfant, Roman Bourgade, Hugues Cassé, Daniel Cordes, Albrecht Kadlec, Raimund Kirner, Jens Knoop, Paul Lokuciejewski, Nicholas Merriam, Marianne de Michiel, Adrian Prantl, Bernhard Rieder, Christine Rochange, Pascal Sainrat, and Markus Schordan. WCET Tool Challenge 2008: Report. In *8th International Workshop on Worst-Case Execution Time Analysis (WCET 2008)*, pages 149–171, Prague, Czech Republic, July 2-4 2008. Österreichische Computer Gesellschaft. ISBN: 978-3-85403-237-3.

[15] Raimund Kirner, Albrecht Kadlec, Peter Puschner, Adrian Prantl, Markus Schordan, and Jens Knoop. Towards a common wcet annotation language: Essential ingredients. In *8th International Workshop on Worst-Case Execution Time Analysis (WCET 2008)*, pages 53–65, Prague, Czech Republic, July 2-4 2008. Österreichische Computer Gesellschaft. ISBN: 978-3-85403-237-3.

[16] Adrian Prantl. The CoSTA Transformer: Integrating Optimizing Compilation and WCET Flow Facts Transformation. In Walter Dosch, Clemens Grelck, and Anette Stümpel, editors, *14. Kolloquium „Programmiersprachen und Grundlagen der Programmierung (KPS'09)“*, number A-07-07 in Schriftenreihe A, pages 172–177. Institute für Informatik und Mathematik der Universität zu Lübeck, 2007.

[17] Raimund Kirner, Jens Knoop, Adrian Prantl, Markus Schordan, and Ingomar Wenzel. WCET Analysis: The Annotation Language Challenge. In *Proceedings 7th Int'l Workshop on Worst-Case Execution Time (WCET) Analysis*, Pisa, Italy, 2007.

[18] Adrian Prantl. Source-to-Source Transformations for WCET Analysis: The CoSTA Approach. In Michael Hanus and Bernd Braßel, editors, *24. Workshop der „GI-Fachgruppe Programmiersprachen und Rechenkonzepte“*, number 0707 in Technische Berichte des Instituts für Informatik, pages 51–60, Olshausenstr. 40, D – 24098 Kiel, 2007. Institut für Informatik der Christian-Albrechts-Universität zu Kiel.

[19] Adrian Prantl. Creating a GCC back end for a VLIW-architecture. Master's thesis, Vienna University of Technology, May 2006.

References

References are available upon request.