Peter Pirkelbauer

phone: (205) 934 8532   e-mail: pirkelbauer@uab.edu   www: www.cis.uab.edu/users/peterp

Education

PhD in Computer Science with emphasis on Programming Languages and Tools (adviser: Dr. Bjarne Stroustrup), Texas A&M University, College Station, TX, December 2010.
MBA with emphasis on Finance, Texas A&M University, College Station, TX, May 2003.
Dipl.-Ing. in Computer Science with emphasis on Software Engineering and System Programming (adviser: Dr. Hanspeter Mössenböck), Johannes - Kepler Universität, Linz, Austria - July 1997.

Research Interests

Static Analysis and Source Code Transformation Systems
Runtime Monitoring and Bug Detection
High Performance Computing
Non-Blocking Software Design

Work Experience

Assistant Professor in the department for Computer and Information Sciences at the University of Alabama at Birmingham. August 2012 - present.

- Leads the i program reliable and scalable systems (iProgress) lab.
    * Development of scalable data structures:
      BLAZE concurrent Library http://iprogress.cis.uab.edu/blaze
    * Research on error detection tools (code instrumentation and error detection, guided model checking)
    * Rejuvenation of legacy MPI codes
    * Compiling MATLAB for heterogeneous systems
- Senior capstone, ethics and current topics in computer science (CS499); Object-oriented design patterns (CS302); Graduate course on compilers (CS702); Graduate course on nonblocking software design (CS792).
- Member of the advisory board for UABTeach.
  UABTeach is part of UTeach and gives science majors the opportunity to get a teaching certificate in their field alongside their regular degree.
- Organized summer course on Android programming (2014,2015)
  http://iprogress.cis.uab.edu/android-camp/

Postdoctoral Researcher in the ROSE Compiler group at the Lawrence Livermore National Laboratory, November 2010-July 2012.

- Lead the development of a dynamic error detection tool (ROSE-CIRM) providing a safety envelop for UPC code.
- Supervised students working on ROSE-CIRM.
- Wrote and contributed to grant proposals.
- Worked on recovery of modern MPI abstractions from legacy code.

Research Assistant in the Parasol Lab, Group for Programming Languages, Techniques, and Tools, Texas A&M University, July 2003 - October 2010.

- The Pivot source-to-source translation infrastructure:
    * Developed the Pivot's C++ frontend that converts EDG's intermediate program representation to IPR (the Pivot's internal program representation).

* Developed a generic abstract syntax tree traversal framework.
* Designed and implemented a light weight pattern specification language. The use of an extended C++ like syntax allows programmers define source code patterns in a form that abstracts away from representation details.
* Designed and implemented a source code rejuvenation framework (e.g., concept recovery from uninstantiated templates).
- Designed and implemented a lock-free dynamically resizable array.
- Extended C++ with open-methods and implemented a prototype compiler (based on the EDG C++ frontend).

Teaching Assistant for Introduction to Computer Systems (CSCE313). Fall 2009.
Taught lab classes and graded homework assignments and programming projects.
Software Engineer (Internship) in the Photoshop Team, Adobe Systems, Summer 2006.
Worked on a programming model that unifies the benefits of the generic programming and object oriented programming paradigm.
Research Assistant (Internship), Lawrence Livermore National Labs, Summer 2005.
Developed an interface between ROSE (LLNL) and The Pivot (Texas A&M), two compiler frameworks supporting analysis and transformations of C++ programs.
Consultant for a city development project, Ingenieurbüro Retter, Krems, Austria, May - June 2003.
Designed and implemented a route recovery software. Based on license plate readings obtained from a number of locations, the software calculates routes and time-sensitive estimates for the traffic volume within a city.
Software Engineer in an R&D project to integrate information systems into a modern manufacturing environment (steel industry), VA Stahl Linz GmbH, Austria, Oct 1998 - Aug 2001.
Tasks included all phases of software development from system analysis, system design, implementation, maintenance to subproject supervision and end user training.

- Designed and implemented a domain specific language (DSL) that dispenses with the direct use of a database query language (SQL). The DSL allows to model and modify client, plant, and process specific business rules during runtime. A DSL specific programming environment provides visual code blocks to further ease writing rules.
- Designed and implemented an on-line surface assistant comparing graded material defects against client specific defect tolerances in order to support human material inspectors in real-time.
- Designed and supervised the implementation of a workflow-component that controls the maintenance process (i.e., versioning, submission, approval) of master data (e.g., check lists, rules).

Social Worker in an elderly home, BAH Leonding, Austria, Oct 1997 - Sep 1998.
System Programmer (Internship), ModulaWare, La Chanenche, France, Summer 1997.
Migration of the OpenVMS AlphaOberon integrated development and runtime environment (e.g., loader, allocator, garbage collector) to support the 64bit memory management capabilities provided by the OpenVMS operating system running on a DEC Alpha microarchitecture.

Synergistic Activities

Program committee member: SAC/PL (2014-2016), IEEE IRI (2013,2014).
Reviewer for Journal of Computer Languages, Systems & Structures (2014,2015), for Central European Journal of Engineering (2013), Journal of Computing and Information Technology (2013), Journal of Systems and Software (2011).
Grant reviewer: NSF (2013), DOE SBIR (2011, 2012).
Book review: Herb Sutter, Andrei Alexandrescu: C++ Coding Standards, 101 Rules, Guidelines, and Best Practices (C++ in Depth Series), Addison-Wesley, November, 2004.

Journal Articles

Reed Milewicz, Marjan Mernik, Peter Pirkelbauer: *SimpleConcepts: A Lightweight Extension to C++ to Support Constraints on Generic Types.* In Computer Science and Information Systems, ComSIS, Online First Issue, 2014.

Peter Pirkelbauer, Yuriy Solodkyy, Bjarne Stroustrup: *Design and Evaluation of C++ Open Multi-Methods.* In Science of Computer Programming 75 (7), Elsevier, 2010, pp 638–667.

Peter Pirkelbauer, Sean Parent, Mat Marcus, Bjarne Stroustrup: *Dynamic Algorithm Selection for Runtime Concepts.* In Science of Computer Programming 75 (9), Elsevier, 2010, pp 773–786.

Peer Reviewed Publications

Hadia Ahmed, Anthony Skjellum, Peter Pirkelbauer: *Petal Tool for Analyzing and Transforming Legacy MPI Applications.* In 28th International Workshop on Languages and Compilers for Parallel Computing, 2015. to appear.

Reed Milewicz, Rajesh Vanka, James Tuck, Daniel Quinlan, Peter Pirkelbauer: *Runtime Error Checking of C Programs.* In ACM Symposium on Applied Computing, Track on Programming Languages, 2015.

Brendan Lynch, <u>Peter Pirkelbauer</u>, Damian Dechev: *Building Fast Concurrent Data Structures through Data Structure Families.* In Many-core Applications Research Community Symposium (MARC) at SPLASH'13, 2013.

Reed Milewicz, Marjan Mernik, <u>Peter Pirkelbauer</u>: *SimpleConcepts: Support for Constraints on Generic Types in C++.* In 4th Workshop on Advances in Programming Languages, 2013.

Peter Pirkelbauer, Chunhua Liao, Thomas Panas, Dan Quinlan: *Runtime Detection of C-Style Errors in UPC Code.* In 5th Conference on Partitioned Global Address Space Models (PGAS), 2011.

Peter Pirkelbauer, Damian Dechev, Bjarne Stroustrup: *Support for the Evolution of C++ Generic Functions.* In 3rd Conference on Software Language Engineering (SLE), LNCS 6563, Springer, 2011.

Damian Dechev, Peter Pirkelbauer, Bjarne Stroustrup: *Understanding and Effectively Preventing the ABA Problem in Descriptor-based Lock-free Designs.* In 13th IEEE International Symposium on Object/component/service-oriented Real-time distributed computing (ISORC), 2010.

Peter Pirkelbauer, Damian Dechev, Bjarne Stroustrup: *Source Code Rejuvenation is not Refactoring.* In 36th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), LNCS 5901, Springer 2010.

Damian Dechev, Peter Pirkelbauer, Nicolas Rouquette, Bjarne Stroustrup: *Semantically Enhanced Containers for Concurrent Real-Time Systems.* In Proceedings of 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (IEEE ECBS), April 2009.

Damian Dechev, Nicolas Rouquette, Peter Pirkelbauer, Bjarne Stroustrup: *Verification and Semantic Parallelization of Goal-Driven Autonomous Software* In Proceedings of 2nd International Conference on Autonomic Computing and Communication Systems (ACM Autonomics), 2008, Turin, Italy.

Peter Pirkelbauer, Sean Parent, Mat Marcus, Bjarne Stroustrup: *Runtime Concepts for the C++ Standard Template Library.* In Proceedings of the 2008 ACM symposium on Applied computing (SAC), 2008. ACM Press.

Peter Pirkelbauer, Yuriy Solodkyy, Bjarne Stroustrup: *Open Multi-Methods for C++.* In proceedings of the 6th International Conference on Generative Programming and Component Engineering (GPCE), 2007. ACM Press.

Damian Dechev, Peter Pirkelbauer, Bjarne Stroustrup: *Lock-free Dynamically Resizable Arrays.* In Proceedings of 10th International Conference on Principles of Distributed Systems (OPODIS), 2006, LNCS 4305, Springer 2006.

Markus Hof, Hanspeter Mössenböck, Peter Pirkelbauer: *Zero-Overhead Exception Handling Using Metainformation.* In Proceedings of 24th Seminar on Current Trends in Theory and Practice of Informatics (SOFSEM), LNCS 1338, Springer 1997.

Book Chapter

Damian Dechev, Nicolas Rouquette, Peter Pirkelbauer, Bjarne Stroustrup: *Programming and Validation Techniques for Reliable Goal-driven Autonomic Software.* Book Chapter in Autonomic Communication. Vasilakos, A.; Parashar, M.; Karnouskos, S.; Pedrycz, W. (Eds.), ISBN: 978-0-387-09752-7, Springer, May 2009.

Technical Reports

Peter Pirkelbauer, Yuriy Solodkyy, Bjarne Stroustrup: *Report on language support for Multi-Methods and Open-Methods for C++.* TR N2216, ISO WG21, March 2007.

Peter Pirkelbauer, Markus Hof, Hanspeter Mössenböck: *Zero-Overhead Exception Handling Using Metainformation* TR CS-SSW-P97-07, Johannes Kepler University Linz, Austria, September 1997.

Invited Talks, Posters, and Workshop Participation

Samuel Collie (undergraduate researcher in iProgress): *The Instrumentation of Unsafe to Safe Functions Using the RTC Runtime Checking Tool.* In ACM Mid-Southeast Fall Conference 2015.

Amalee Wilson (undergraduate researcher in iProgress): *Conversion of MATLAB to C++ to Improve Performance and Efficiency*, ACM Mid-Southeast Fall Conference, 2015.

Runtime Checking C Programs, Auburn University, Invited Talk. March 2015.

Braden Groom (undergraduate researcher in iProgress): *Improving Authorship Attribution Methods Using Compiler Information.* In ACM Southeast Fall Conference 2014.

Hadia Ahmed, Peter Pirkelbauer, Anthony Skjellum: *An Automatic Transformation and Analysis Tool for Improving Legacy MPI Applications*, In 28th IEEE International Parallel & Distributed Processing Symposium. PhD symposium, 2014.

Non-blocking Programming Techniques, University of Innsbruck, Invited Talk. July 2013.

Portable Non-blocking Data Structures, University of Alabama, Invited Talk. IEEE Alabama Computer Society talk of the month, March 2013.

Dynamic Bug Detection for C, C++, and UPC. Workshop on Quality Software: A Festschrift for Bjarne Stroustrup. Texas A&M University. April 2012.

ROSE-CIRM: Dynamically Finds C-Style Flaws in UPC code. Poster at the PGAS booth, Supercomputing (SC), November 2011.

Programming Language Evolution and Source Code Rejuvenation. Invited Talk. Lawrence Berkeley National Laboratory, May 2010.

The Pivot - a source-to-source framework for getting more elegant and efficient code. Invited Talk. ISCR - Lawrence Livermore National Laboratory, August 2006.

The Pivot Framework: Design and Implementation. Workshop on Domain Specific Languages. with Steve Cook, Damian Dechev, and Gabriel Dos Reis. Argonne National Lab, August 2004.