

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, 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.

Erasmus Exchange Student, Universidad de Murcia, Murcia, Spain, Fall 1995 - Spring 1996.

Research Interests

Concurrent and Nonblocking Algorithms

Scientific Computing

Runtime Monitoring and Bug Detection

Compilers, Static Analysis, and Source Code Transformation Systems

Work Experience

Assistant Professor in the department of Computer Science at the University of Alabama at Birmingham. August 2012 - present.

- Leads research in the “i program reliable and scalable systems” (iprogress) lab:
 - * The Blaze concurrent library: Generic library of concurrent data structures in C++. True to generic design principles, a data structure can be instantiated with different memory managers (e.g., garbage collector, epochs, hazard pointers) to meet different system requirements (e.g., progress guarantees, efficiency).
 - * Error detection tools: Tool support for detecting runtime errors in programs. RTC supports runtime error detection of software written in C. Ariadne uses priors from static analysis to guide the Java Pathfinder to interesting interleavings.
 - * Optimizing transformations for HPC applications: ROSE-Petal is a tool to optimize HPC applications, by replacing legacy MPI primitives with newer and more efficient versions. Petal finds code to overlap computation and communication.
 - * Compiling Matlab to C++ for execution on heterogeneous architectures: A ROSE based transformation that produces C++ from Matlab source code and schedules the execution of large scale Matrix computations on available accelerators using high-performance libraries (BLAS, LAPACK).
- Teaches graduate and undergraduate courses in computer science:
 - * Multiprocessor Programming (CS792), Spring 2013, Spring 2015: developed a new graduate course discussing nonblocking algorithms and programming techniques for multiprocessor architectures.
 - * Software Engineering (CS720), Spring 2018: developed and taught a new graduate course in Software Engineering with an emphasis of software integration and the design of enterprise level applications.

- * Compiler Design (CS402,CS602,CS702), Spring 2014, Summer 2018: Developed a new course for undergraduate and graduate students covering fundamental compiler topics, such as front-end, semantic checking, and simple code generation, and advanced topics, such as dataflow analysis, and automatic loop parallelization using the polyhedral model.
 - * Programming Language Seminar (CS708), recurring: a seminar for Ph.D. students discussing recent papers published at top-tier conferences.
 - * Senior Capstone Course (CS499), Fall 2012: Extended an undergraduate course that reviews key concepts in the curriculum, intellectual property, privacy, topics related to ethics in computer science, scientific thought, and philosophy. Added 60% new material.
 - * Programming Language Overview (CS401,CS501), Spring 2016, Spring 2017, Spring 2018: taught an undergraduate course and bridge course for early graduate students. The course discusses major programming paradigms, their realization in programming languages, and their impact on application design and implementation. Added 30% new material to cover C++, generics in Scala, and an overview of Coq.
 - * Functions and Modeling (MA 361): developed jointly with Dr. John Mayer (who is teaching this course) a course on mathematical modeling of systems. The course is required for Mathematics students, and an elective for CS students (mainly students in the UABTeach program). Infused the use of Mathematica and its programming language for problem modeling into a standard UTeach course.
 - * Object-Oriented Design (CS302), Fall 2013-Fall 2015: Developed a new programming language course in Java (as a second course in Java) discussing object-oriented design patterns and implementation techniques.
 - * Object-Oriented Programming (CS203), Fall 2016-Fall 2017, Fall 2018: developed a new course on object-oriented programming as a second programming language course (and a first course in OOP).
 - * Web Development (CS221), Fall 2018: developed a new course on web technologies (incl. HTML, CSS, Forms, Javascript), and client-server design and implementation.
- Service activities:
- * UAB-CAS Curriculum and educational policies committee (CEPC): 2017-present.
The CEPC reviews proposals on educational policies and new programs in the College of Arts and Sciences and makes recommendations to the Dean when warranted.
 - * UAB Undergraduate curriculum committee (UCC): Alternate member, 2018-present.
The UCC reviews proposals on educational policies and new programs at UAB and makes recommendations to the Provost when warranted.
 - * UABTeach advisory board: 2014-2018.
Worked with UABTeach and the Alabama State Board of Education to create a pathway for UABTeach / Computer Science graduates towards obtaining a teaching certificate as mathematics educator.
 - * Oversees CS department website: 2017-2018.
 - * High-school programming contest: 2014-2017 (assisted Dr. Sprague), 2018 (organizer).
 - * Helped organize ImpactAmerica summer sessions (2016, 2017).
 - * Organized CS departmental seminars: 2013-2017.
 - * Organized programming camps for middle and high school students: Girls Can Code (2016, held by my Ph.D. students), Android Programming (2014, 2015).

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

- Led 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.

Jr. Teaching Assistant, Johannes - Kepler University, Austria, Fall 1993 - Spring 1995, Spring 1997.

Jr. Teaching Assistant for Computer Science courses including Introduction to Programming and Algorithms.

Synergistic Activities

Co-chair: ICPP - Software Track (2018).

Program committee member: SAC/PL (2014-present), LCPC (2017), IEEE IRI (2013 and 2014).

Reviewer for Journal of Computer Languages, Systems, & Structures (2014-2018), IEEE Access (2018), Journal of Parallel and Distributed Computing (2017), Central European Journal of Engineering (2013/14), Journal of Computing and Information Technology (2013), and Journal of Systems and Software (2011).

Working Group Member: UTeach CS education working group (2016-2018).

Grant reviewer: NSF (2013, 2016, 2017), DOE SBIR (2011, 2012).

Book review: Maurice Herlihy, Nir Shavit, Victor Luchangco: The Art of Multiprocessor Programming, 2nd edition, Kaufmann, to appear.

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

Peter Pirkelbauer, Amalee Wilson, Christina Peterson, Damian Dechev. Blaze-Tasks: A Framework for Computing Parallel Reductions Over Tasks, ACM Transactions on Architecture and Code Optimizations, to appear.

Olga Pearce, Hadia Ahmed, Rasmus W. Larsen, Peter Pirkelbauer and David F. Richards. Exploring Dynamic Load Imbalance Solutions with the CoMD Proxy Application, Future Generation Computer Systems, 2018. in press.

Reed Milewicz, Peter Pirkelbauer: *Refinement of Structural Heuristics for Model Checking of Concurrent Programs through Data Mining*, Journal on Computer Languages, Systems and Structures, vol. 47, part 2, pp 170–188, Elsevier, 2017.

Reed Milewicz, Rajesh Vanka, James Tuck, Daniel Quinlan, Peter Pirkelbauer: *Lightweight runtime checking of C programs with RTC*, Journal on Computer Languages, Systems and Structures, Elsevier, 2016.

Reed Milewicz, Marjan Mernik, Peter Pirkelbauer: *SimpleConcepts: A Lightweight Extension to C++ to Support Constraints on Generic Types*. In Advances in Systems, Modeling, Languages and Agents, ComSIS Journal, 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, Purushotham Bangalore, [Peter Pirkelbauer](#). *Transforming Blocking MPI Collectives to Nonblocking and Persistent Operations*, EuroMPI, pp 3:1–3:11, 2017.

Viktoria Riiman, Amalee Wilson, Reed Milewicz, [Peter Pirkelbauer](#): *Comparing Artificial Neural Network and Cohort-Component Models for Population Forecasts*. Population Association of America, Annual Meeting 2017.

Reed Milewicz, [Peter Pirkelbauer](#): *Ariadne: Hybridizing Directed Model Checking and Static Analysis*. In 10th IEEE International Conference on Software Testing, Verification and Validation, (ICST '17), pp 442-447, 2017.

[Peter Pirkelbauer](#), Amalee Wilson, Hadia Ahmed, Reed Milewicz: *Memory Management for Concurrent Data Structures on Hardware Transactional Memory*. In Transact 2017, workshop at PPOPP'17, 2017.

[Peter Pirkelbauer](#), Reed Milewicz, Juan Felipe Gonzalez: *A portable lock-free bounded queue*. In 16th International Conference on Algorithms and Architectures for Parallel Processing, Springer, LCNS 10048, pp 55–73, 2016.

Hadia Ahmed, Anthony Skjellum, [Peter Pirkelbauer](#): *Petal Tool for Analyzing and Transforming Legacy MPI Applications*. In Languages and Compilers for Parallel Computing (LCPC '15), Springer, LCNS 9519, pp 156–170, 2016.

Reed Milewicz, Rajesh Vanka, James Tuck, Daniel Quinlan, [Peter Pirkelbauer](#): *Runtime Checking C Programs*. In Symposium on Applied Computing, Programming Language Track, 2015.

Brendan Lynch, [Peter Pirkelbauer](#), 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, [Peter Pirkelbauer](#): *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.

Peer Reviewed Journal Articles by iProgress Students

Ezio Bartocci, Borzoo Bonakdarpour, Christian Colombo, Norman Decker, Yliès Falcone, Felix Klaedtke, Klaus Havelund, Yogi Joshi, Reed Milewicz¹, Giles Reger, Grigore Rosu, Julien Signoles, Daniel Thoma, Eugen Zalinescu, Yi Zhang. First international Competition on Runtime Verification: rules, benchmarks, tools, and final results of CRV 2014, The International Journal on Software Tools for Technology Transfer (STTT), Springer, 2017.

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

Petal: Rejuvenation of MPI Application & Ariadne: Static Analysis Meets Model Checking, University of Tennessee at Chattanooga, Invited Talk, March 2018.

Concurrent Object Design on Systems with Hardware Transactional Memory, University of Central Florida, Invited Talk, August 2017.

Concurrent Object Design on Systems with Hardware Transactional Memory, Texas A&M University, Invited Talk, January 2017.

¹The article reports on our participation in the First International Runtime Verification Competition 2014. Each team named one representative for the article. Our team's work was conducted by my doctoral student Reed Milewicz under my supervision.

Runtime Checking C Programs, Auburn University, March 2015.

Nonblocking Programming Techniques, University of Innsbruck, Austria, July 2013.

Portable Nonblocking Data Structures, University of Alabama, 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.

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

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

Posters and Workshop Participation

Hadia Ahmed, [Peter Pirkelbauer](#), Purushotham Bangalore, Anthony Skjellum: Work-in-Progress Abstract: Compiler-Assisted Scientific Workflow Optimization, 2nd Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems at Super Computing (SC), 2017.

Viktoria Riiman, Reed Milewicz, Amalee Wilson, [Peter Pirkelbauer](#): Using Machine Learning for Population Projections, Presentation at the The Federal-State Cooperative for Population Projections Meeting, U.S. Census Bureau, 2016.

Hadia Ahmed, Anthony Skjellum, [Peter Pirkelbauer](#): *From Blocking to Nonblocking MPI Collectives*. Poster at EuroMPI, 2016.

Hadia Ahmed, Anthony Skjellum, [Peter Pirkelbauer](#): *Analyzing and Transforming Legacy MPI Applications*. Poster at Languages and Compilers for Parallel Computing, 2015.

Hadia Ahmed, [Peter Pirkelbauer](#), Anthony Skjellum: *Automatic Transformation and Analysis Tool for Improving Legacy MPI Systems*, Poster, IDPDS, 2014.

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

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.

Student Research Presentations (under my supervision)

Amalee Wilson: Compiling MATLAB to C++ to Improve Performance and Efficiency. Poster presentation. Grace Hopper Celebration of Women in Computing. Houston, TX. October 2016.

Amalee Wilson: Compiling MATLAB to C++. 15th Annual University of Alabama System Honors Research. Huntsville, AL. April 2016.

Amalee Wilson: Compiling MATLAB to C++. University of Alabama Birmingham Expo: Engaged Scholarship Symposium. Birmingham, AL. April 2016.

Sam Collie: The Instrumentation of Unsafe to Safe Functions Using the RTC Runtime Checking Tool, ACM Mid-Southeast Conference, 2015.

Amalee Wilson: Conversion of MATLAB to C++ to Improve Performance and Efficiency, ACM Mid-Southeast Conference, 2015.

Braden Groom: *Improving Authorship Attribution Methods Using Compiler Information*, ACM Mid-Southeast Conference, 2014.

Software Releases

The BLAZE concurrent library, scalable data structures for C++, 2014. Available at: <http://iprogress.cs.uab.edu/blaze/>

Experimental Matlab frontend for the ROSE compiler infrastructure, 2017. Available at: <http://rosecompiler.org/>

Graduate Students

Current (Ph.D.)

Amalee Wilson: Internship at Intel, Boston, MA.

Graduated (Ph.D.)

Dr. Hadia Ahmed: Improving Performance Portability and Productivity of Parallel Scientific Applications for Exascale, August 2017.

First Position: Postdoctoral Researcher at the Lawrence Berkeley National Laboratory, CA.

Internships at the Lawrence Livermore National Laboratory and the Oak Ridge National Laboratory.

Dr. Reed Milewicz: Improving the Scalability of Directed Model Checking of Concurrent Java Code through Hybrid and Distributed Analysis, August 2016.

First Position: Postdoctoral Researcher at the Sandia National Laboratory, NM.

Internship at the Lawrence Livermore National Laboratory.

Graduated (M.S. - Research)

Sujan Khadka: FastNumerics: Compiling Matlab To C++, April 2016.

First Position: Uber Advanced Technologies Group, Pittsburg, PA.

Internship at the Lawrence Livermore National Laboratory.

Graduated (M.S. - Research) Amalee Wilson: A Scalable Red-Black Tree for Hardware with Transactional Memory, April 2018.

First Position: Los Alamos National Laboratory, NM.

Internship at Intel, Boston, MA.

Graduated (M.S.)

Juan Felipe Gonzalez (2015, Motorola Systems), Nick Dzugan (2016, Software Engineer at Walmart), Deekshitha Pasnoor (2016, BBVA Compass), Manasa Chitrashekar (2016, Perficient), Weida Tan (2017, Ph.D. student at UAB).

Ph.D. Committee Member

Current: Walker Haddock (UAB, Dr. Purushotham Bangalore)

Graduated: Dr. Jushua Yue (University of Alabama, Dr. Jeff Gray), Dr. Amin Hasani (UAB, Dr. Purushotham Bangalore), Dr. Sagar Thapaliya (UAB, Dr. Purushotham Bangalore).

Supervised Undergraduate Students

Nicholas Bowen: Concurrent dispatch structures for pattern matching in C++, 2016.

First Position: (not yet graduated).

Rachel Adamec: Translating Matlab's Nargin to C++, 2016.

First Position: n/a

Sam Collie: Accelerating Runtime Error Checking Through Compile-time Check Elimination (Honors Project), presented at the ACM Mid-Southeast Meeting, 2015.

First Position: Full stack web developer at a healthcare IT company, Ontario, Canada.

Amalee Wilson: Translation of Matlab to C++ (Honors Project), presented at the ACM Mid-Southeast Meeting (2015) and the UAB honors project and the University of Alabama honors project symposia (2016).
First Position: Ph.D. student in my research group.

Joshua Akpoebi Agberebi: OpenMP and Transactions, 2016.
First Position: Center For Advanced Public Safety, The University of Alabama.

Braden Groom: Improving Authorship Attribution Methods Using Compiler Information, presented at the ACM Mid-Southeast Meeting, 2014.
First Position: Apple Inc.

Joshua Pritchett: A generator for secure but memorable passwords (with Kalyani Bhagat), 2014.
Internship at the Lawrence Berkeley National Laboratory.
First Position: Ph.D. student at UCLA

Kalyani Bhagat: A generator for secure but memorable passwords (with Joshua Pritchett), 2014
First Position: Blue Cross Blue Shields of Alabama.

Michael Gotfryd: implementation of a web based event management system for CIS, 2014.
First Position: Brasfield and Gorrie.

Atsuhiko Ishikawa (exchange student from Ibaraki University in Japan in Spring 2013, topic: UML bad smell detection and refactoring)

Supervised High School Students

Kayleigh Alexander (ASFA), Verifying Loop Termination of Java `for` Loops. (2nd place in the Central Alabama Science and Engineering Fair; 3rd place in the Alabama Science and Engineering Fair – won Intel Excellence in Computer Science award). (May 2016-April 2017)