IBM Research Intern Recruitment
Programming Languages and Software Engineering Department

The Programming Languages and Software Engineering Department at IBM Research is looking for top students interested in the following areas

- Programming Language Design and Implementation
- Real-Time Systems
- Virtual Machine Performance
- Multicore and GPU Processing
- Reconfigurable Hardware
- Programming Tools
- Software Quality/Analysis/Verification
- Software Engineering
- Human Centric Tools

for both summer internships and full-time positions. Our summer internship program provides students with a unique opportunity to experience the research environment that exists here at IBM. The program is highly competitive, and is designed for top students who are interested in tackling challenging research problems. Many of our student interns have authored papers, or even theses, out of their sojourn with us, and many have contributed significantly to open source projects.

IBM is a leading provider of products and services that relate to the full software life cycle activities (i.e., requirements, design, code, test, deployment and maintenance). IBM Research fosters close partnerships with the other IBM divisions that work in these areas and consequently we are constantly exposed to real world problems. This puts IBM Research in a unique place where we become aware of fundamental research challenges and opportunities in the real world before they are visible to the research community at large.

Our department performs leading research in a number of areas, all centered on making software development and deployment more efficient and effective while improving productivity of the organizations and quality of the resulting software.

A few representative examples of active research in these areas are listed on our webpage: `http://www.research.ibm.com/compsci/plansoft/index.html`.

Students that are interested in programming languages and software engineering, and that want to pursue summer internships with our department at the NY T. J. Watson Research Center, should send a resume to the appropriate contact listed in the project descriptions that follow, or simply email our summer internship coordinator, Rodric Rabbah (rabbah@us.ibm.com), for further instructions. Students may also directly apply online at the general IBM Research jobs site: `http://www.research.ibm.com/about/career.shtml`.

1 Programming Language Design and Implementation

- **Cross-Language Integration**
  Contact: Martin Hirzel (hirzel@us.ibm.com)

  Many programs are written in more than one language. Reasons for this include reuse of legacy libraries, access to platform functionality, and efficiency. IBM research investigates language designs for cross-language integration, aiming at programmer productivity, static and dynamic error checking, portability, and efficiency.

- **Programming Models for Distributed and Web Applications**
  Contact: John Field (jfield@us.ibm.com)

  Increasingly, application software is being organized around collections of loosely-coupled distributed components, often communicating over the internet. Current programming languages are ill-suited for this domain because they were designed for monolithic applications. We are designing a new programming model for distributed applications which combines the ease of use of current web scripting languages with the ability to package such scripts into robust, reusable distributed components. Some critical challenges include designing an appropriate type system for this setting and defining security annotations that will allow separately-developed components to interact safely.

- **Domain-specific concurrent programming**
  Contact: John Field (jfield@us.ibm.com)

  We are investigating several novel approaches to programming targeted classes of applications which can exploit modern multicore architectures. One project synthesizes lock-free algorithms using a model checker to exhaustively explore interleavings of user-specified sequential building blocks. Another project is targeting special-purpose compilation of streaming applications.

- **Atomic Sets**
  Contact: Frank Tip (ftip@us.ibm.com)

  Concurrency-related bugs may happen when multiple threads access shared data and interleave in ways that do not correspond to any sequential execution. Their absence is not guaranteed by the traditional notion of "data race" freedom. We have developed a new definition of data races in terms of 11 problematic interleaving scenarios, and proved that it is complete by showing that any execution not exhibiting these scenarios is serializable for a chosen set of locations. Our definition subsumes the traditional definition of a data race as well as high-level data races such as stale-value errors and inconsistent views. We have also developed a language feature called atomic sets of locations, which lets programmers specify the existence of consistency properties between fields in objects, without specifying the properties themselves. We use static analysis to automatically infer those points in the code where synchronization is needed to avoid data races under our new definition. An important benefit of this approach is that, in general, far fewer annotations are required than is the case with existing approaches such as synchronized blocks or atomic sections. Future work consists of developing a type system and inference for atomic sets.

2 Real-Time Systems

Contact: David Bacon (bacon@us.ibm.com)

IBM Research's work in real time garbage collection has created an entire new market, allowing Java to be used in application domains where it was previously considered impossible. Current projects include time-portable programming models, real-time visualization tools, dynamic compilation to reconfigurable hardware, and a lock-free, massively scalable, autonomic real-time garbage collector. We are also building real-time systems in Java: a music synthesizer and a helicopter.
More information: `http://www.research.ibm.com/metronome`.

3 Virtual Machine Performance

Contact: Erik Altman (ealtman@us.ibm.com)

Higher-level and managed languages make software development easier, but introduce serious performance challenges. IBM Research has the unique opportunity to develop novel solutions to these challenges, and watch these solutions have global impact in IBM's production JVMs.

4 Multicores, GPUs, and Reconfigurable Hardware

- **Evaluation of Application Development on GPUs**
  Contact: Rajesh Bordawekar (bordaw@us.ibm.com)

  Commodity graphics processing units (GPUs) are being increasing used for developing general-purpose applications (GPGPUs). We have recently started a project to study key aspects of the GPGPU problem: programming models, algorithms, and performance modeling/optimizations. Specifically, we are investigating architectural and programming issues (e.g., thread scheduling, work partitioning, and tiling) using key application kernels on nVIDIA GPUs using CUDA.

- **Liquid Metal**
  Contact: Rodric Rabbah (rabbah@us.ibm.com)

  A new project with the goal of developing a programming model and runtime that allows skilled Java-programmers to write portable high level code that can be efficiently refined into parallel software or directly into hardware (FPGAs). Liquid Metal aims to facilitate moving computation fluidly between software and hardware, offering significant performance and power advantages compared to other computing approaches.
  More information: `http://www.research.ibm.com/liquidmetal`.

5 Programming Tools

- **Parallel Tools**
  Contact: Evelyn Duesterwald (duester@us.ibm.com)

  The trend towards multicore and multi-threaded architectures calls for a new generation of tools to enable a larger class of programmers to effectively develop parallel programs. The goal of the Parallel Tools Project is to develop advanced parallel programming tools in an integrated Eclipse environment that include developments tools, performance tools and tools for the detection of common concurrency related errors.

- **Refactoring and Program Transformation**
  Contact: Frank Tip (ftip@us.ibm.com)

  The work on refactoring at IBM Research has resulted in significant advances of the state-of-the-art, and in the contribution of several advanced refactorings to Eclipse. Current research efforts are focused on refactorings for advanced language features (e.g., concurrency), and supporting program analysis and infrastructure.

- **Miniatur**
  Contact: Frank Tip (ftip@us.ibm.com)

  We have developed a tool, Miniatur, for finding bugs in object-oriented code based on bounded model checking with SAT solving. The tool takes as input a specification in first-order logic and a fragment of code, and checks for consistency. Its output consists of counterexamples, if they can be found. Because it considers an under-approximation of the code, it never outputs a spurious error report. Miniatur has been used successfully to check structural properties of classes in the Java Collections Framework, as well as the equality contract for a number of open-source Java programs. We are currently adding support for concurrency, in order to check small but complex multi-threaded code such as concurrent garbage collectors and non-blocking data structures.

- **TuningFork Visualization Platform**
  Contact: David Bacon (bacon@us.ibm.com)

  TuningFork is an extensible platform for data visualization, based on a functional stream architecture with high-level operators. The work is highly interdisciplinary, spanning highly scalable (many GB) trace processing, design of new data visualizations, human interface design, and networked collaboration.

6 Software Quality/Analysis/Verification

- **Advanced Tools for Software Quality**
  Contact: John Field (jfield@us.ibm.com)

  We are building a number of tools to detect bugs, recover design/specification information, manage concurrency, and ensure proper allocation of scarce runtime resources, all with the goal of improving software quality and productivity. Research challenges in this area include scaling deep analyses to large applications while providing precise and focused feedback to application developers, providing extremely lightweight runtime instrumentation for dynamic monitoring of large applications, and combining testing, dynamic analysis, and static analysis in ways that exploit each of their best attributes.

- **Language-based Security**
  Contact: John Field (jfield@us.ibm.com)

  We are working on a number of projects whose purpose is to enforce application security through programming language constructs and/or program analysis. The problems addressed include design and development of automated tools for access control and information flow security enforcement.

7 Software Engineering

Contact: Edith Schonberg (ediths@us.ibm.com)

Topics in Software Engineering include (but not limited to): software architecture evolution and design, model analysis and transformation, testing, traceability across artifacts, development governance, agile and iterative software development, risk management, understanding team dynamics via mining software repositories and communications information, run-time analysis of J2EE applications, scalable ontology reasoning, etc.

8 Human Centric Tools

Contact: Juerg von Kaenel (jvk@us.ibm.com)

The Human Centric Tools group is seeking qualified summer interns to work on a variety of challenging projects: Our tools, applications and services are becoming more and more interconnected and complex. We are looking at ways to make these tools usable and simplify our environments to increase productivity. This ranges from new programming models for web components to end user programming and collaborative environments, such as virtual worlds, to User-Centered Design methodologies for these new environments, to glanceable interfaces and visualizations. Preference will be given to candidates with prior experience working in research environments.