Research Statement
Miryung Kim (miryung@cs.ucla.edu)
Associate Professor of Computer Science at University of California, Los Angeles
Reporting Period: September 2014-August 2016

Research

I have not only published papers in top-tier software engineering conferences (two papers at ICSE 2015, two papers at ICSE 2016), but also broadened my interest through an inter-disciplinary collaboration (VLDB 2016, HotCloud 2016, and SOCC 2016).

Interactive Big Data Debugging

I am leading a new collaboration project on big data debugging with Professors Tyson Condie and Todd Millstein. An abundance of data in science, engineering, national security, and health care has led to the emerging field of big data analytics and has accelerated the adoption of distributed technologies such as Hadoop and Spark. However, the current cloud computing model lacks the kinds of expressive and interactive debugging features found in traditional desktop computing. Our project aims to address these challenges with the development of novel debugging primitives and tool-assisted fault localization services for big data analytics. This requires re-thinking traditional step-through debugging. BIGDEBUG builds on top of the data provenance capability to support fine-grained record level tracing. The efficiency of the debugging process can be optimized through incremental computation.

Our team was productive in carving out a road-map for this new emerging area and producing early publications. I and my PhD student Muhammad Gulzar led the design of novel debugging primitives, which was published in ICSE 2016. We also released our tool BIGDEBUG in a public domain1, and produced a tool demonstration paper at FSE 2016. Professor Condie and his post-doc Matteo Interlandi led the work on data provenance for Apache Spark, which was published in VLDB 2016, a top venue in the database community. Professors Millstein and Condie led the work on incremental computation, which was accepted at SOCC 2016. I and my student Gulzar led the writing of our vision paper in HotCloud 2016, which sketches a faulty data localization service for big data processing. This project is a fruitful interdisciplinary collaboration that combines expertise from multiple fields. I also led a proposal submission to NSF as a PI. Though the proposal was not funded last year, based on the positive feedback, we plan to re-work the proposal and submit again this coming year.

The Emerging Role of Data Scientists on Software Development Teams

We are at a tipping point where the demand for analyzing large scale software telemetry, machine, and quality data is rapidly increasing. Data scientists are becoming popular in software industry, e.g., Facebook, LinkedIn, etc. Microsoft is creating a new career path for data scientists. To understand this emerging role of data scientists, in collaboration with Tom Zimmermann, Rob DeLine, and Andrew Begel at Microsoft Research, I interviewed data scientists in eight different organizations at Microsoft. As a result, we identified five distinct working styles of data scientists and cataloged strategies for increasing the impact and actionability of their work. I led this collaboration project as a first author. This work was published at ICSE 2016 and was also featured as the main material for my collaborator, Rob DeLine’s keynote speech at SPLASH 2015.

Interactive Inspection, Refactoring, and Testing of Systematic Changes

Interactive Inspection: To help developers ensure that they applied similar changes to all relevant locations, my students Tianyi Zhang, Myoungkyu Song, Joseph Pinedo, and I designed an interactive summarization approach for searching systematic changes and detecting anomalies. When a developer selects a sub region within a diff patch, this approach allows developers to interactively create an abstract diff template by parameterizing its edit content and context. By matching the

1https://sites.google.com/site/sparkbigdebug/
diff-template against the rest of the program, it also detects anomalies. We evaluated our interactive code review approach *critics* through user studies in industry. The results show that human subjects using *critics* answer questions about systematic changes more correctly with more time saving during code review tasks, in comparison to the baseline use of Eclipse diff. The technology was evaluated by professional software engineers at Salesforce.com during Tianyi Zhang’s summer internship in 2014 and all participants said they would like the tool to be integrated to their code review environment. This attests to the fact that our tool scales to an industry-scale project and can be easily adopted by professional engineers. This work was published in ICSE 2015 with an acceptance rate of 18.5% and a tool demonstration paper at FSE 2014. ICSE and FSE are flagship conferences in software engineering.

**Clone Removal Refactoring:** When developers perform similar changes to different contexts, this systematic editing may present an opportunity to extract common code from multiple methods. In collaboration with my graduate students Na Meng and Lisa Hua and Dr. Kathryn McKinley at Microsoft Research, I investigated a new refactoring approach that takes systematic edits as inputs and automatically performs a generalization task that factors common code and parameterizes any differences. To our knowledge, our work is the most advanced refactoring algorithm for automated clone removal. By applying this tool to real-world systematic edits, we found that clone removal is feasible only in the half of the cases, and therefore both automated clone removal and automated systematic editing are necessary. This work was published in ICSE 2015 with an acceptance rate of 18.5%.

**Test Transplantation for Code Clones:** Code clones are common in software. When applying edits to clones, developers often find it difficult to ensure the correctness of similar edits. Existing techniques check syntactic consistency but do not help examine the behavioral differences between clones. The problem is exacerbated when some clones are tested, while their counterparts are not. To address this issue, my PhD student Tianyi Zhang and I are designing a differential testing technique to identify behavioral differences between clones via automated code transplantation. This work has the potential to significantly improve software correctness, because up to almost a quarter of software systems consists of code clones from somewhere else but the cost of developing test cases is high.

**Services and Visibility**

I am highly visible within my broader research community and I have extensively served the community through various leadership roles. I was a General Chair for MSR 2016–13th International Conference on Mining Software Repositories. MSR is the fastest growing, sub-area conference in software engineering, and it is the most attended ICSE co-located event with 150 to 200 attendees each year. I am serving as a steering committee member for MSR conferences (2016-2018). Together with Professor Darko Marinov at UIUC, I am serving as a Program Co-Chair for Visions and Reflections Paper Track at FSE 2016–24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering.

I was on the Program Board of ICSE 2016, am serving on the Program Board of ICSE 2017, and have been already invited to serve on the Program Board of ICSE 2018. ICSE follows the program board/program committee model, where program committee members are responsible for reviews, while the program board members serve both roles of writing peer reviews and managing the review process, similar to associate editors in journals. There are roughly 30 board members each year. ICSE is the premier conference in software engineering, and continuous invitation to ICSE program boards is a recognition of my status in the software engineering community and the quality of my research work.

In terms of editorial services, I was invited to be an Associate Editor for IEEE Transactions on Software Engineering (2016-current). TSE is a leading journal in the area of software engineering. I am also an Associate Editor for Journal of Empirical Software Engineering (2015-current) and Journal of Information and Science Technology (2014-current).