Queen's School of Computing

Seminar Series

Log Intelligence: Supporting Logging Decisions Using Development Knowledge

Logs are generated at run-time by logging statements that are deliberately added into the source code by developers. Logs generated during the execution play an essential role in field debugging and support activities of large software systems. These logs are not only for the convenience of developers and operators but have already become part of legal requirements. For example, the Sarbanes-Oxley Act of 2002 stipulates that the execution of telecommunication and financial applications must be logged. In recent years, many companies (e.g., IBM, BlackBerry and Microsoft) have started leveraging the rich knowledge in logs to support the development and operation of their large software systems. The broad usage of logs lead to the emergence of a new market for log analysis platforms (e.g., Splunk, XpoLog, and Logstash), which support collecting, storing, searching, and analyzing the large amounts of log data.

Although logs are widely used in practice, and their importance has been well-identified in prior software engineering research, logs are leveraged in an ad hoc manner (even in many of state of the art log analysis platforms). First of all, there exists no support for logging decisions. Logging decisions rather depend on developers' gut feelings. All too often, developers cannot find a correct place in the source code or they tend to log too much. Making it worse, developers often change logging statements without considering the needs of other stakeholders.

In order to support making logging decisions, we build classifiers and regression models in order to model the logging decisions. For example, we model whether a logging statement needs to be updated in a code commit and what logging level is appropriate chosen. We find that such classifiers and regression models can assist in making logging decisions with high accuracy. On the other hand, we manually study log-related issues, such as missing logging statements, from the issue tracking systems of open source software. Based on the manual study results, we developed an automated tool that detects evident log-related issues. Our tool can detect existing inappropriate logging statements and identify new issues that are later reported to developers.

June 8, 2017
2:30pm-3:30pm
Dupius 215
Light Refreshments

Weiyi Shang

Assistant Professor
Concordia University, Montreal

Weiyi Shang is an assistant professor in the Department of Computer Science and Software Engineering at Concordia University, Montreal. He has received his Ph.D. and M.Sc. degrees from Queen's University (Canada) and he obtained B.Eng. from Harbin Institute of Technology. His research interests include big data software engineering, software engineering for ultra-largescale systems, software log mining, empirical software engineering, and software performance engineering. His work has been published at premier venues such as ICSE, FSE, ASE, ICSME, MSR and WCRE, as well as in major journals such as TSE, EMSE, JSS, JSEP and SCP. His work has won premium awards, such as SIGSOFT Distinguished paper award at ICSE 2013 and best paper award at WCRE 2011. His industrial experience includes helping improve quality and performance of ultra-large-scale systems in BlackBerry. Early tools and techniques developed by him are already integrated into products used by millions of users worldwide. Contact him at shang@encs.concordia.ca; http://users.encs.concordia.ca/~shang.