Title: Refinement Techniques in Mining Software Behavior

Speaker: Mr Zuo Zhiqiang

Date/Time: 6 January 2015, Tuesday, 10:00 AM to 11:30 AM

Venue: MR1, COM1-03-19

Supervisor: Dr Khoo Siau Cheng, Associate Professor, School of Computing

Abstract:

Mining software behavior has been well studied to assist in numerous software engineering tasks for the past two decades. Two kinds of studies which received much attention are specification mining and statistical debugging. To tackle the lack of precise and complete specifications, specification mining is proposed to automatically infer software behavior from the execution traces as specifications. In order to support debugging activities, researchers have developed various statistical debugging approaches (e.g., statistical bug isolation and bug signature mining) which commonly collect two groups of execution traces and employ statistical techniques to discover the discriminative element as the bug cause or signature.

Among the execution traces analyzed by both specification mining and statistical debugging, there exist a significant number of useless elements. Mining directly over the raw execution traces wastes many computing resources and possibly produces meaningless results due to the meaningless elements. To enhance the efficiency and/or effectiveness of software behavior mining, refinement techniques are required to remove unwanted elements from raw execution traces. However, currently there is a lack of systematic refinement techniques for both software behavior mining studies. This dissertation presents a specific systematic refinement technique for each of the above two studies.

For specification mining, we propose a semantics-directed specification mining framework which exploits a user-specified semantic analysis to filter out the semantically irrelevant events from the execution traces before mining. Consequently, specifications mined are all semantically significant, and mining becomes far more efficient. Based on the framework, we present a particular dataflow sensitive specification mining system where dataflow semantics is taken into consideration. The experimental results show that our approach can generate high-quality specifications and scale well to real-world programs. Moreover, the mined specifications can practically help program understanding and bug detection.
For statistical debugging, we devise a novel hierarchical instrumentation (HI) technique to refine the execution traces. Based on HI, we effectively prune away unnecessary predicates instrumented and analyzed, and thus greatly reduce the overhead of statistical debugging. Firstly, we apply the HI technique to predicated bug signature mining (MPS) and propose an approach via HI called HIMPS. The empirical study shows that HIMPS can achieve around 40% to 60% saving in disk storage space usage, time and peak memory consumption compared with MPS while producing the same results. Secondly, we investigate the adoption of HI to cooperative bug isolation for field failures and propose an iterative approach via HI. The experimental results validate that our approach not only saves much instrumentation effort, but also sharply reduces the end-user’s runtime overhead.