SIGCSE 2012 SRC Graduate Student Research Abstracts

Deduced Social Networks for Educational Portal

Monika Akbar, *Virginia Tech*

Educational portals mostly rely on resources. Combining an online community and analyzing and harnessing the user trends can provide users with more information on how others succeed in the vast information space. The AlgoViz Portal collects metadata on Algorithm Visualizations and provides community support. We show how logs can be used to discover latent relationships between users, deducing an implicit social network. Clustering of the log data allows us to find different page-viewing patterns, giving us an idea about the different groups of users and their information seeking behavior.

Student Adherence to Test-First Programming in the Classroom

Kevin Buffardi, *Virginia Tech*

Virginia Tech employed Web-CAT, an automated testing tool, for several years in CS1 and CS2 classes to introduce and reinforce Test-Driven Design (TDD). Our concern over some students not accepting or adhering to test-first principles drew attention to the need to identify these students' behaviors and the consequential impact on their programs. Our research used Web-CAT data to analyze snapshots of students' work-in-progress to identify testing and development patterns of those not adhering to TDD. Analysis also focused on the relationship between TDD adherence and several measurements of code quality. Based on the results, we drew inferences on possible strategies to improve acceptance and adherence to TDD.

Computing Computational Thinking

Kyu Han Koh, *University of Colorado at Boulder*

Visual programming with game/simulation creation is one of common approaches to raise computer science interests in K-12 education. Several research papers indicate that motivational benefits of visual programming learning are successfully brought to computer science education, but still it is not clear what kinds of knowledge students have actually learned through making games and/or simulations. In this research, a method to analyze the semantic meaning of visual programming is provided to support educational benefits of visual programming in computer science education.

CipherXRay: Exposing Cryptographic Operations and Secrets from Binay Execution

Xin Li, *George Mason University*

We focus on the technical feasibility of exposing the cryptographic operations and secrets from the execution of a potentially obfuscated binary executable. We build CipherXRay upon one of the defining characteristics of all good cryptographic algorithms - the avalanche effect. CipherXRay is able to not only reliably identify the cryptographic operations, but also accurately pinpoint the location, size and boundary of the input buffer, the output buffer and the cryptographic key buffer. We have evaluated CipherXRay with OpenSSL, KeePassX, Kraken malware and a number of third party programs with built-in compression and checksum. Our results demonstrate that current implementations of cryptographic algorithms achieve virtually no secrecy if their execution can be monitored.

An Analysis of Optimizing Power-Performance on Multi-core CPUs

Vijayalakshmi Saravanan, *Ryerson University*

Our thesis proposal is to investigate the optimum power/performance of multi-core processors. This state-of-the-art report is examined in two different architectures such as multi-thread and multi-core in terms of power and performance gains: (a) to find the optimum number of processor cores in such architectures based on instructions per cycle (IPC) and (b) to present an efficient buffered LR (Left-Right) algorithm for pipeline stall reduction

which reduces the power consumption and maximizes the CPU utilization (i.e., throughput). We describe the experimental setup and simulators required to verify the results and then propose a benchmark for the design of multi-core processors.

Dynamic Test Input Generation for Database Applications to Achieve High Mutation Score

Tanmoy Sarkar, *Iowa State University*

Traditionally automatic generation of test inputs focuses on achieving high block or branch coverage and then assessing the adequacy of the generated test inputs is done by mutation testing. It is a technique where typical programming errors are replicated by modifying actual program and then checking whether (or to what extent) a test suite can distinguish between the actual program and its modified version. The modified versions are referred to as the mutants. Mutation testing of test suits for database applications presents an interesting and important challenge. Database applications consists of complex programming constructs as in typical applications and also consists of embedded SQL queries in various application paths. Mutants in such applications can be realized by not only modifying original program but also by modifying the embedded SQL queries. The objective of mutation testing in database applications, therefore, to check whether a given test suit can distinguish the actual application from its mutants. While mutation testing for typical program and for SQL queries exist, research in the field of mutation testing for database applications which combines both these aspects is still in a state of infancy. In this work, we propose to address the problem of mutation testing for database application by developing an automated test case generation technique which not only will provide high degree of branch coverage but also will achieve high SQL mutation score.

Using Refactoring to Achieve Quality Without A Name

M. Edward Wilborne, III, *North Carolina State University*

Mapping Christopher Alexanderís fifteen principles of good design to Martin Fowlerís behavior preserving code transformations, known as Refactoring, may result in code that has Alexander's “quality without a name”. This research will provide programmers a guide to improving the quality of object oriented code without the program resorting to making random refactorings.