

SIGCSE 2012 SRC Undergraduate Student Research Abstracts

Evaluating Relevance Metrics for Search in Personalized Network Spaces

Anthony Bentancur, *University of North Carolina at Charlotte*

In Personalized Network Spaces (PNets) people have access to an abundant amount of that originates from people and devices embedded within the surrounding environment. Users and applications that utilize information in these settings require the ability to get information about the "here" and "now", exhibiting tight spatiotemporal integration. We propose Gander, a personalized search engine that allows users to perform search of the "here" and "now" in the "here" and "now"; in other words, Gander supports dynamic search of the rapidly changing environment without reliance on a centralized index or connection to the Internet. The contribution of this work is the creation and evaluation of Gander relevance metrics that describe desirable spatiotemporal properties of data, and are used to order the search results for presentation to the search user. In this work, I create two new relevance metrics and evaluate their effectiveness in ordering results by incorporating them into Gander queries that are executed over a PNets environment. The two relevance metrics that I designed and implemented rank results for a search query for amusement park rides in order of: 1) rides that the user has not ridden and 2) rides near a bench. The NDCG values from evaluating these two relevance metrics were around .70 and .80 respectively. Our results, measured in terms of Normalized Discounted Cumulative Gain, show that our spatiotemporal relevance metrics provide high degrees of utility to Gander users.

Exploring Text-Based Analysis of Test-Case Dependencies of Web Applications

Camille Cobb, *Washington and Lee University*

Test cases formulated from user sessions of a web application often depend on what happened in a previous session to the shared, persistent application state. When ordering test suites, these dependencies must be maintained to avoid predominantly executing error code. This work explores the feasibility of computing dependencies directly from the test cases. We propose a novel text-based analysis of the URL requests. Our exploratory study of URL requests to five deployed web applications suggests that resources contain textual clues useful for estimating test-case dependencies. The results provide a foundation to implement a test-case dependency model that uses a text-based analysis approach to identify dependencies and an arranger that uses the model to automatically order test suites.

Improving the Flexibility of JFLAP through Presentation of Explicit Formal Definitions and Symbol Alphabets

Julian Jenkins, *Duke University*

Java Formal Languages and Automata Package (JFLAP) is a software visualization suite focused on the canonical concepts of a course in Formal Languages and Automata (FLA) Theory. The software eases the construction, modification, and testing of FLA algorithms and machines, catalyzing learning. Recent additions to JFLAP include a formal definition component which highlights the elements that formally underlie a given automata or grammar. This includes a panel corresponding to any symbol alphabets associated with a formal definition which allows for more varied, complex definitions while retaining backwards compatibility. This new system makes explicit much of the work previously done implicitly by JFLAP, encouraging greater user interaction and providing more rapid, productive feedback.

Using Static Analysis to Evaluate Student Code

Alexandra Johnson, *Carnegie Mellon University*

The automated assessment of student code is beneficial both for the learners and for the course staff. Students benefit from the immediate feedback and reduction of metacognitive requirements of early programming assignments, while course staff reclaim hours of tedious grading. Although important, current implementations of assessment tend to be course grained and focus more on the functionality (or lack thereof) of a solution than the knowledge a student has demonstrated. In this work we evaluate the use of a static analysis technique, abstract syntax trees, combined with JUnit

testing to award partial credit to incorrect program solutions. Results show a significant improvement with the use of ASTs in automated program assessment.

Teaching Computer Science Concepts Through Educational Games Using Social Aspects Within Peers

Javier Olaya, *Kean University*

The demand of computer science is growing and unfortunately the early college students are avoiding this degree. Production of Computer Science degrees are declining and the consequences are low participation of under-represented groups. The largest group currently is women and this is a loss to the computer science field. The students that come straight to university with no experience in programming have more difficulty accomplishing computer science assignments. Therefore, it is necessary for these students to find ways to enhance their learning. This research assesses the effectiveness of learning computer science lessons through an educational game.

Cloud Filter

Mateus Oliveira *The College of New Jersey*

Cloud Filter is a research project conducted at the University of Calgary that serves to question the security capabilities of cloud services. The primary issue that we targeted is the lack of fine-grain filtering of data storage. Services, such as Amazon S3, limit users to only a handful of locations where data can be stored. We suspect that this is because these cloud services do not have the hardware capabilities to support this security mechanism to all of its users. To test our hypothesis, we created our own Cloud Filtering Linux Kernel Module to implement this location control management.

Metaphor: A Declarative Approach to Computational Art

Jaimie Sauls, *University of Arizona*

Computer science is misrepresented as a science and such misperceptions may contribute to students' negative views and reluctance to join this field of study. The Laboratory for Computer Science creates online lessons for high school students that introduce computing theories in an interactive way. This study examines how students' perceptions of computer science change upon completion of these labs. The focus is on the students' perspective of computer science and their place in the field irrespective of their identification with a specific minority group. Identifying whether the stigmas or stereotypes are present with the students that experience these lessons and whether a deeper knowledge of the underlying theories in computer science will change these views is the goal.

Modeling the Power Consumption of Computer Systems with Graphics Processing Units (GPUs)

Stephanie Schmidt, *Sonoma State University*

To help optimize computer systems' energy efficiency, researchers have developed models relating these systems' instantaneous power consumption to resource utilization metrics available in software. Existing power models assume the major consumers of dynamic power are the central processing unit (CPU), memory and disk. The increased use of graphics processing units (GPUs) for general-purpose computing compromises this assumption. Therefore, models that exclude the GPU may significantly mispredict power consumption. I built full-system power models for two GPU-equipped machines in order to test this hypothesis. This research is the first to quantify: the inaccuracy of traditional high-level system power models for GPU workloads; the benefits of including GPU metrics in a full-system power model when applied to both GPU and non-GPU workloads; and the sensitivity of the model to the specific choice of GPU metrics.

Four-dimensional mapping with mobile robotics and RGB-D sensors

Tanner Schmidt, *Duke University*

Mapping with mobile robots in both two and three dimensions has been an active area research and has produced many great results. However, the environments in which robots operate are often dynamic. Adding time as a fourth dimension to the mapping problem allows robotic systems to track changes in their surroundings. This can be accomplished via extension the octree data structure, by inserting a map between parent nodes and their children such that the subtree of a given node at any point in time can be accessed by selecting from the map the node that was valid at that time. This method is efficient in that it does not increase the depth of the octree. This data structure can then be queried to determine the state of the environment in the past or to analyze changes to the map.

Experimentally Exploring Algorithmic Descriptions of Three-Dimensional Geometry

Elizabeth Skiba, *SUNY Geneseo*

GUI based software is the traditional tool for describing complex geometry, however I hypothesize that algorithmic descriptions are superior for reasons including their ability to reuse pieces of geometry. I designed an experiment to compare three geometry description tools. Descriptions were generated manually, through the use of a GUI, and algorithmically. I specifically investigate their strengths and weaknesses when changing a pre-existing geometry. I measured a ratio of initial size of the description to size of a change to that description. Results suggest that algorithmic description has significant advantages over other methods of describing changes to geometry.

PICS: Platform Independent Cloud Scripting

Kelly Smith, *Slippery Rock University*

Previous work in minimizing the inherent difficulties of parallel programming has yielded three popular methods for automated, scaled parallelization: OpenMP, MPI, and CUDA. This research borrows much of its philosophy from these methods, adds parallel language constructs to JavaScript, and provides a general-purpose framework for spawning algorithms to a heterogeneous, platform-independent cloud of processing entities including mobile devices. Currently the cloud consists of clients on the following platforms: Windows, Linux, OSX, Android, and any modern web browser. Initial results have been encouraging, and this framework has proven to be effective in bringing parallel computing to untapped computing resources.

Modular Program Design and Verification in RESOLVE

Daniel Welch, *Denison University*

In 2003, Sir Charles Anthony Hoare issued a challenge to the computing community at large to construct a verifying compiler. To answer this challenge, a language named RESOLVE was developed to not only achieve the goal of verification, but also create a language supportive of modular, reuse-oriented, software design principles. To determine the developmental state of RESOLVE in relation to this vision, we implement Kruskal's minimum spanning tree algorithm as proof of concept. This algorithm, due to its reliance of multiple components, presents an opportunity to test RESOLVE's new verifier and compiler on a sufficiently complicated project.

Matching XML Schema Documents

William West, *The College of New Jersey*

With the dynamic nature of databases, there is a growing need to re-organize and combine sets of data. While the combination of two or more data sets is often done manually, this can become a costly and time-consuming process. Thus, automatic methods of database schema matching become necessary. XML is unique in its use of a tree-like

structure and strictly defined element, hence requiring its own matching algorithms to achieve optimal accuracy. In this paper, we focus on developing an XML schema matcher in Java. A schema matching framework is outlined in detail, using two standard XSD schemas as examples. We find that through the use of existing Java libraries and external APIs, a custom and fully functional XML schema matcher can be successfully developed in Java.