CCSC Consortium for Computing Sciences in Colleges

Southeastern Region

22st Annual Southeastern Conference

Student Research Contest

Extended Abstracts

November 7 and 8, 2008
Augusta State University
Augusta, Georgia

Student Research Contest

Extended Abstracts

Compare Object Equality Using Equivalence Operator

Jennifer Brown
Converse College
Spartanburg, SC
Jennifer.Brown@converse.edu
Faculty Sponsor: Jonathan Qiao, Dept. of Math, CS & Physics

Most Java textbooks explain differences between scalar variables and objects from the very beginning. When it comes to the equality comparison, the difference is further emphasized that equivalence operator == can only be used to compare scalar variables, not objects. For object equality comparison, method *equals* in class Object has to be overridden and needs to be called. This is true in general but comes at a price because of the overhead of the method call. This can be easily seen in the much better performance of the equality comparison of *int* variables than that of *Integer* objects. If a class has many significent fields (in terms of equality comparison) and some of them may themselves are objecs, the equality comparison can be even more expensive. Since object equality comparison frequently occurs in object-oriented programming, the added cost can hurt the performance. The goal of this work is to explore when and how in Java to speed up object equality comparison and hence to improve program efficiency.

In Java, an object is manipulated through its reference. If a and b are two objects of class A, a and b are merely the references to the two objects, though they are treated as objects. Therefore, $a == b$ does not compare the equality of the two objects. It evaluates to be *false* even if the two objects hold the same value and *true* only when a and b point to the same object. Since operator == is merely to make an integer equality comparison, it is a lot more efficient than a call to method *equals*, which may recursively involve more *equals* calls. Therefore, the key here is to find when two objects are equal to each other if and only if they are the same object.

It is guaranteed in Java that for every string literal, there is only one copy stored in a JVM [1]. For example, consider two *String* objects *s1*, *s2* and *s3* created as following:

String s1 = "Hello world", s2 = "Hello world";
String s3 = new String ("Hello world");

Then, *s1 == s2*, *s1 == "Hello world"* and *s2 == "Hello world"* all evaluate to *true.* However, *s1 == s3* is *false* while *s1.equals(s3)* returns *true.* This is because both *s1* and *s2* are references to the same String object *"Hello world"* in the constant pool, while *s3* is a reference to a copy of *"Hello world"* in the heap. Therefore, if we know two String objects are string literals in Java, operator == instead of method *equals* should be used for the equality comparison to substantially speed up the execution. A common example of this can be seen in many client-server and Web applications, where server side needs to find whether errors occur. There is a String object, *error*, with the initail value "". When an error occurs, an error message of string literal is assigned to *error.* At the end of each operation, the server needs to determine how to respond to the client based on whether *error* is a empty string, which can be tested by *error == ""* instead of *error.equals("").*

Interestingly, class String also provides method *intern()* which returns the String constant of the same content in the constant pool. Using the above example, s3.intern() == s2 will evaluate to *true.* This can facilitate string search process if all String objects in a container are known to be literals. Suppose *names* is an array of string literals and *person* is a String object, which

int/==	Integer/equals	Student/equals	String Literal/==	String/equals
3.4	16.6	73.5	41.2	270.0

Table 1: Execution time in milliseconds of 5 search procedures.

may not be a string literal. When we want to search the array for *person*, assigning *person* by *person.intern()* would allow operator == to be used in place of method *equals* in the loop for a performance gain.

The second scenario is when a class is defined in a way that no objects of this class can be created by client. Consider the following simple example,

```
public class A {
    public final static A a1 = new A ( ... ),  a2 = new A ( ... );
    private A( ... ) { ... } // This is the only constructor
}
```

Since the class has only one private constructor, it cannot be extended and clients have no way to instantiate the objects. In fact, *a1* and *a2* will be the only two *A* instances, which will exist once class *A* is initialized, though there may be many more references to these two objects. For example, *A.a1* and *A.a2* might be passed to other methods and assigned to other references (*A a = A.a1.*) As a consequence, the equality comparison of the instances of *A* can be conducted by applying == to their references regardless how many references of *A* exist. A typical use of this is in the Java typesafe enum programming[2]. When there is only one public instance of the class, it is called singleton, which represents some intrinsically unique system component.

Package java.lang has a class, *Class*, which allows the type of objects (and other information) be found at run time. This is especially important when polymorphism is used. Consider a container of a number of objects all belonging to a parent class. In case we want to find whether two objects in the container, *a* and *b*, belong to the same class, we could apply operator == to *a.getClass()* and *b.getClass()*. This is because a JVM creates one and only one *Class* object for each class it loads [3]. When *a* and *b* belong to the same class, they must point to the same object. It can also happen when we want to find whether a passed object, *o*, belongs to the class we assume, such as FileOutputStream. Again, *Class.forName("java.io.FileOutputStream") == (Class)o.getClass()* does the job without a call to *equals*.

Table 1 gives experiment results. The first experiment is to search from three arrays of the same size, 1,000,000, but of different types, *int*, *Integer* and *Student* (which has 8 fields, all of primitive types or *String*,) respectively. The first three columns of the table give the execution time of three searches. The second experiment is to repeat the search from an array of 9 string literals for 1,000,000 times. The fourth column gives the execution time when the search key is a string literal while it is not for the fifth column.

The future work will be focused on the second scenario, that is when we should forbid external object creation so that operator == can be applied to the equality comparison.

References

[1] Gosling, James, Joy, B., Steele, G., Bracha, G.: The Java Language Specification, Second Edition. Addison-Wesley (2000)

[2] Bloch, J.: Effective Java - Programming Language Guide. Addison-Wesley (2001)

[3] Eckel, B.: Thinking in Java, 4th Edition. Prentice Hall (2006)

"RFID Technology Used in Percussion Music Education and Music Therapy"

Shelby Dawes
Georgia College and State University
Milledgeville, Georgia
Shelby_dawes@ecats.gcsu.edu
Faculty Sponsor: Dr. Gita C. Williams Phelps, Department of Information Technology and Marketing

I created a preliminary program in Java called DrumBeats that could be used in the field of music education. This program utilizes RFID sensors hidden within a drum that interact with the software and provide instant feedback to a music student on whether or not they played the drum on the right beats. Programs like DrumBeats can ultimately improve the music student's experience by providing more instant feedback during self-guided practice, and by providing a private unintimidating environment to facilitate learning. At this time, technology is being used extensively in music education, and music educators recognize that in order to keep their lessons relevant to students with a high level of computer literacy, they must utilize computer technology [1]. Popular video games such as "Rock Band" could also have a useful application in music education if modified to fit musical lesson plans and equipped with more realistic instruments. With some modification, interactive programs such as this project's prototype could help students learn to play instruments via online instruction.

The main program of DrumBeats interfaces with an RFID sensor hidden in the top of the drum and a small 0.5" RFID tag in the tip of a drumstick. When the drumstick comes within 3 inches of the RFID sensor, the program knows that the drum was hit and the visual display of the drum stick changes to give the user feedback that their drum beat was registered. Both RFID devices were purchased by Georgia College and State University at http://www.phidgets.com.

To handle the exact timing of the drum beats and to serve as a metronome for the user, I used a Swing Timer object. This Timer triggered events every 10th of a second (100 milliseconds). The beats of the song were stored in an Access database, and when a beat matched the timing of the metronome, a short drum beat sound played. This provided an audible guide that the user could follow along with while the song was playing. For the software testing, I set up one song, "The March of the Siamese Children" from the musical The King and I.

The visual display in this program is still in development. Right now the user sees a white bar on the screen similar to a sheet of music, with vertical black bars where the beats are to be played. A small yellow dot jumps on to each line from left to right as the beats occur to serve as a visual guide, but this guide is in its preliminary stages and is still not intuitive for the user. The program requires more fluid animations for the user to truly see as well as hear when the next beat is to be played. At the bottom of the screen, I created a red drum and drumstick which changed colors every time the user hit the drum, but in this program's final stage I would like to incorporate actual animation of the drumstick moving up and down.

References

1. Choksy, Lois. Abramson, Robert M. Gillespie, Avon E. Woods, David. York, Frank. Teaching Music in the Twenty-First Century. Second Edition. Prentice-Hall, 2001.

Can PowerShell be Used to Teach Computer Science?

Michael England
Wingate University
Wingate, NC
maengland@wingate.edu
Faculty Sponsor: Dr. E. Kent Palmer, Math and Computer Science

This research sought to determine ways to use Windows PowerShell to teach computer concepts. Windows PowerShell was selected, because it was an interesting new tool with little published research available. The researchers looked for where that tool could be used in the computing curricula. Each of the disciplines identified by the Joint Task Force for Computing Curricula was examined for potential applications. These disciplines included: computer science (CS), information systems (IS), software engineering (SE), computer engineering (CE), and information technology (IT) [1].
Preliminary review of the software and the five curricula indicated that PowerShell could best be used for teaching computer science (CS) or information technology (IT). This preliminary review involved researching information online for Computing Curricula and PowerShell that was mostly found on the ACM website, the Microsoft website, and blogs that were written about PowerShell.
The researchers developed a methodology to test whether the use of PowerShell in computer science instruction could result in the learning outcomes defined in the Joint Task Forces model curricula for CS and IT. This methodology was developed by doing research on various forms of evaluation such as usability inspection and heuristic evaluation and then deciding what method would be best to use to evaluate PowerShell. Using this methodology, the researchers tested a number of potential PowerShell assignments to determine their suitability for use in instruction.

The Potential of PowerShell:

Windows PowerShell is a powerful tool for system administration that allows administrators to automate routine tasks. PowerShell provides administrators a convenient interface to work with logs and WMI. It provides a command line interface that enables users to run a variety of scripts. Since Microsoft began developing PowerShell in 2003, only a small amount of research has been published on its uses, particularly on its use for instruction. Microsoft included Windows PowerShell in Microsoft Server 2008 because they intend PowerShell to replace the command prompt on computers that use Windows.
Windows PowerShell makes use of cmdlets (command-lets). Cmdlets are similar to commands (such as dir and copy) found in other shells. PowerShell shell functionality is easy to extend since anyone can create his or her own cmdlets. PowerShell also makes extensive use of aliases and automation. Windows PowerShell extensively uses functions in its programming. These functions allow users to develop code for different computer actions. PowerShell also is capable of working with registry and environmental variables.
Unlike other shell languages, PowerShell is object oriented. PowerShell makes use of providers – some supplied with the product and others created by users. PowerShell providers create an interface that allows users to access different types of stored data in a consistent manner. PowerShell also makes use of .NET objects.

Methodology:

The initial review of PowerShell indicated it had potential for teaching computer concepts found in the CS and IT curricula. The potential of PowerShell was strongest in IT, so much of the research was focused on the learning outcomes defined for the IT curricula. A form was developed based on the learning outcomes defined for the IT curricula [2], some key topics included in the CS curricula [3], and recent literature published on learning theory [4,5,6]. The researchers ranked the exercises ability to meet each learning outcome using a scale of 1-7.
The researchers used the form to collect data from a number of PowerShell exercises. The researchers read chapters from a number of PowerShell books and performed any included exercise. If the book did not include exercises, the researchers created assignments based on the chapter content. For comparison, the researchers also did several assignments using Linux shell scripting, but not enough Linux assignments were completed for a full comparison of the two shell languages.

Results:

The researchers added the rankings for the learning outcomes determined by performing various exercises. The total scores on learning outcomes were then compared to determine the top five learning outcomes for Programming Fundamentals and for System Administration. Even for learning outcomes in the top five, the average ranking was only around 4.5.

Conclusion:

From the results, it was concluded that Windows PowerShell is not the best tool to teach Computer Science. Although the researchers had hoped that PowerShell could be used for teaching operating systems, PowerShell did not rank high enough on learning outcomes to justify its use in operating system courses. Additional research might identify other places that PowerShell could be used, such as applied system administration courses at a community college. Additional research could also compare shell scripting in PowerShell with Linux.

References:

[1] The Joint Task Force for Computing Curricula 2005 (2005). Computing Curricula 2005: The Overview Report, Retrieved from http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf
[2] The Joint Task Force for Computing Curricula (2008). Computing Curricula Information Technology Volume. *Journal*. Retrieved from http://campus.acm.org/public/comments/IT%20Curriculum%20Draft%20-%20May%202008.pdf
[3] The Joint Task Force for Computing Curricula 2001 (2001). Computing Science: ACM, Retrieved from http://www.acm.org/education/education/education/curric_vols/cc2001.pdf
[4] Angelo, T. A., & Cross, K. P. (1993). *Classroom Assessment Techniques: A Handbook for College Teachers* (2nd ed.). San Franciso: Jossey Bass Publishers.
[5] Fitzpatrick, J. L., Sanders, J. R., & Worthen, B. R. (2004). *Program Evaluation: Alternative Approaches and Practical Guidelines* (3rd ed.). Boston: Allyn and Bacon.
[6] Weimer, M. (2002). *Learner-Centered Teaching: Five Key Changes to Practice*. San Francisco: Jossey-Bass.

Faculty Reference: E. Kent Palmer

Using Only Bumper Sensors to Map and Navigate an Environment

Alan M. George
Undergraduate Student Majoring in Computer Science and Mathematics
Florida Southern College, Lakeland, FL

Introduction

The topic of a robot mapping an unknown environment and navigating through the environment using the map has been studied for over two decades. Most of the published papers on this topic have involved sophisticated robots with elaborate sensors [1, 2]. However, only a limited amount of research has been performed on the other end of the spectrum. The goal of this research is to create an algorithm and software prototype that allows a robot to map and navigate through an environment using only bumper sensors.

Materials and Restrictions

The robot used in this research was the iRobot *Create*, a relatively inexpensive robot that has a limited variety of sensors. The only sensors used in this research were the *Create's* built-in bumper sensors. The bumpers were utilized to detect obstacles in the environment. The operational environment for the research consisted of an indoor location with an almost-level, almost-smooth surface.

Research Problems and Solutions

A robot with only bumpers for sensors is similar to a blindfolded person. While a person who is not blindfolded has the ability to continually look around a room to see the location of objects, someone who is blindfolded is only aware of objects as the objects are touched. The longer period of time that has passed since the person has touched a particular obstacle, the less certain the person may be of the exact location of the object. This scenario motivates the three issues of interest to this research: localization and object discovery using only bumper sensors, application of the discovery data to develop a global map of the environment, and navigation using the global map.

To explore an unknown area containing multiple obstacles, the robot needs a localization algorithm to enable it to efficiently navigate through the area and identify the location of each obstacle it encounters. Since a robot with only bumper sensors cannot detect an object before contact, navigation with an algorithm designed for sonar sensors or laser range finders is not applicable to this scenario. To address this problem, a localization algorithm from the University of Michigan [1] was modified to represent the robot's local environment by placing a polar graph around the robot with a specific radius. The center of the polar graph is the center of the robot. When an object is hit, the location is saved on the graph, creating a radar-like representation of known obstacles around the robot. This graph can then be used by the robot to calculate where to turn next to avoid any known obstacles.

To map the environment as the robot explores, an algorithm was developed to use the output of the robot's localization algorithm to update a global map consisting of a two-dimensional grid represented as a matrix of cells. Using the points found in the localization algorithm, each cell's value represents the presence or absence of an object in that cell. Since the environment was discovered by a continually-moving robot using only bumper sensors, the amount of environmental error resulted in inaccuracies in the calculations of the location of obstacles encountered. This environmental error made it impossible to create an accurate map based purely on the direct mathematical calculations. To address this issue, fuzzy logic can be applied to the algorithm, providing each cell with a more descriptive value than just a true or false indication of whether an obstacle was encountered at this location. The result is a mapping algorithm that updates a global map consisting of a grid of cells whose data give the approximate location of obstacles.

To navigate through the environment using the map, the grid was converted into a graph. The vertices of the graph are the cells that do not contain an object. The edges of the graph are the horizontal and vertical lines that correspond with each edge that an object-free cell shares with another object-free cell. An A* algorithm can be applied to this graph to find an efficient path to the robot's desired destination.

Methods and Results

A software prototype and algorithm was developed for robotic localization and was used to demonstrate a robot's exploration of an environment by avoiding objects found within the radius of the polar graph. This Java prototype runs on both Linux and Windows platforms. Demonstrations of the algorithm and the prototype were successfully conducted several times in indoor locations with almost-level, almost-smooth surfaces. In addition, the mapping and navigation algorithms were designed to illustrate how a robot with limited sensors might map an unknown environment and then use the map for navigation. The results of this research indicate that it is possible for an inexpensive robot with only bumper sensors to successfully explore, map, and navigate through an unknown environment.

Future Research

It is expected that other students can add to this research by using the supporting documentation for the three algorithms and the software prototype. The localization algorithm can be expanded to implement a more intelligent process for deciding how to explore an area. The mapping algorithm can be expanded to include fuzzy logic to provide a more descriptive map. The navigation algorithm can be expanded to use other graph theory algorithms for shortest path or to create a graph from the map in a more efficient way. Ultimately all the algorithms can be combined to create a complete software system that will allow a simple, inexpensive robot to successfully explore, map, and navigate and unknown environment in real-time.

References

[1] Borenstein, J., and Y. Koren. "THE VECTOR FIELD HISTOGRAM - FAST OBSTACLE AVOIDANCE FOR MOBILE ROBOTS." IEEE Journal of Robotics and Automation 7 (1991): 278-88.

[2] S. Thrun, W. Burgard and D. Fox. "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping" . In *Proc. Intl. Conf. on Robotics and Automation*, San Francisco CA, May 2000.

Semantic Feedback Interface for Timing Analysis

William M. Graeber
Furman University
Greenville, South Carolina
william.graeber@furman.edu
Dr. Christopher A. Healy, Computer Science

Computer science research projects often span multiple generations of students. This was true for a timing analysis program under development at my university. A semantic web interface for live testing and historical performance reports was able to reduce the amount of time required to analyze the impact of changes to project code, and made the entire timing analysis program available online for demonstration purposes and collaboration. Additionally, the significance of the semantically correct interface was to provide a consistent interface among different browsers and impaired users, and unifying historical performance reports gave project designers a method to determine if design changes to the project required further investigation.

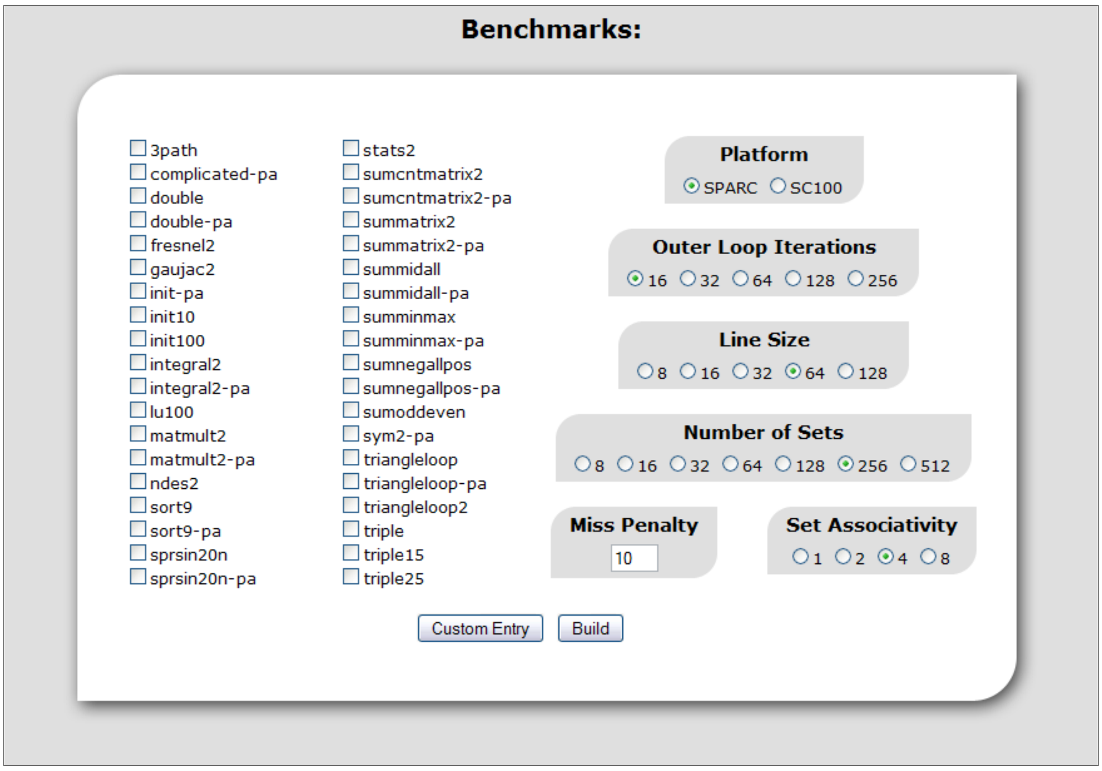
Related interfaces have been created by several institutions in order to achieve similar goals. Dodd et al. used a web interface to share simulation results of Order-to-Delivery systems and processes [1]. The Order-to-Deliver simulator was focused on enhancing basic client interactions with an existing application rather than placing emphasis on project development. Contrastingly, Kapadia et al. directed attention to the high availability of geographically dispersed locations through using an online interface [2]. The system created a central repository for over 50 university and commercial simulation packages. In addition to the work of Kapadia et al. and Dodd et al., Kwan-Lui Ma presented another benefit which is a key feature of the timing analyzer interface. He pointed out that a correctly designed interface may allow project developers to be geographically dispersed in addition to clients using a traditional web interface [3]. A larger developer scope allowed more people to effectively collaborate on a project with increased speed and efficacy. Each article showed a valuable quality about creating a web interface for an existing or new program, but the timing analyzer interface went further by combining and expanding upon them. Additionally, the interface allowed developers to work on a processor architecture that may not readily be available, and its design allowed for further data analysis programs to easily be implemented.

In this instance a series of shell scripts had previously been created to test the efficacy of a timing analysis program created to determine the worst case code scenarios. No convenient way existed for a new project member to begin committing changes without first spending a considerable amount of time learning the order and arguments of the scripts. Timing analysis was conducted on an extensive set of benchmark code, and manually compiling each one could easily become a tedious task. The interface automated the process and gave users a list of predefined benchmarks and options that were available such as outer loop iterations, line size, number of sets, miss penalty, and set associativity (Fig. 1). Also, the interface allowed a user to input and test a custom benchmark for rapid development. When a benchmark was executed with given parameters, a results page was displayed along with any previous data generated by that user.

The web interface made use of PHP and MySQL to provide the functionality required to dynamically compile and test scripts for benchmark code analysis. Several bash scripts were written to simplify the task of interfacing PHP with the code compilation and testing as well. The user-generated benchmark reports were stored in a SQL database after each execution to pool data collection. MySQL was chosen to archive the data due to its widespread adoption and to allow portability of the data set. The extensibility of SQL has already proven a beneficial design decision, as visual data analysis programs have already begun development.

The dynamic web technologies used have shown themselves to be very useful for merging complex actions into a lightweight environment in which developers can work. However, the choice of a web application as the interface also causes the project to require periodic maintenance in order to ensure its evolution continues with rapidly emerging web standards and technologies. For example, AJAX is a technology that would greatly benefit the interface and allow the data to be seamlessly manipulated or visualized. Furthermore, the server-side code would become much more modular and readable if conformed to an Model-View-Controller (MVC) framework. MVC frameworks allow developers to code in a more object-oriented fashion by further abstracting the pages (views) from code objects (models). By abstracting model processing from the HTML code and inserting controller tags pointing to logic for the specific page, separate developers may modify the client-side interactions and server-side logic in parallel.

Overall, the project was a huge success by greatly simplifying the testing and data collection of an ongoing university project on timing analysis. The timing analysis project has gained a wider audience for data collection due to the addition of its web interface. Moreover, it has also benefited from the ease of collaboration now available to distributed developers, increased data collection, and abstraction created for the further development of data visualization plugins.



☐ 3path
☐ complicated-pa
☐ double
☐ double-pa
☐ fresnel2
☐ gaujac2
☐ init-pa
☐ init10
☐ init100
☐ integral2
☐ integral2-pa
☐ lu100
☐ matmult2
☐ matmult2-pa
☐ ndes2
☐ sort9
☐ sort9-pa
☐ sprsin20n
☐ sprsin20n-pa
☐ stats2
☐ sumcntmatrix2
☐ sumcntmatrix2-pa
☐ summatrix2
☐ summatrix2-pa
☐ summidall
☐ summidall-pa
☐ summinmax
☐ summinmax-pa
☐ sumnegallpos
☐ sumnegallpos-pa
☐ sumoddeven
☐ sym2-pa
☐ triangleloop
☐ triangleloop-pa
☐ triangleloop2
☐ triple
☐ triple15
☐ triple25

Platform: [✓ SPARC] SC100

Outer Loop Iterations: [✓ 16] 32 64 128 256

Line Size: 8 16 32 [✓ 64] 128

Number of Sets: 8 16 32 64 128 [✓ 256] 512

Miss Penalty: 10

Set Associativity: 1 2 [✓ 4] 8

Fig. 1. Initialization page with benchmark list and build settings

References

[1] Dodd, C., Kumara, S. R. T., Lee, Y., Tang, K., Tew, J., and Yee, S. 2002. Simulation applications in the automotive industry: simulation anywhere any time: web-based simulation implementation for evaluating order-to-delivery systems and processes. In *Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers* (San Diego, California, December 08 – 11, 2002). Winter Simulation Conference. Winter Simulation Conference, 1251-1259.

[2] Kapadia, N. H., Fortes, J. A., and Lundstrom, M. S. 2000. The Purdue University network-computing hubs: running unmodified simulation tools via the WWW. *ACM Trans. Model. Comput. Simul.* 10, 1 (Jan. 2000), 39-57. DOI = http://doi.acm.org/10.1145/353735.353738

[3] Ma, K. 2007. Creating a collaborative space to share data, visualization, and knowledge. *SIGGRAPH Comput. Graph.* 41, 4 (Nov. 2007), 1-4. DOI = http://doi.acm.org/10.1145/1331098.1331105

RFID in Chemistry Education

Will Hawthorne (wilirius@gmail.com)

Georgia College & State University

Through the study of the history of computers in education and the many attempts at implementing computer aided instruction, I have come to the conclusion that interactive courseware may be the most successful in acclimating the K-12 educational environment to a more technological and student-centered type of instruction. As science scores in Georgia and across the country fail to keep up with the international standard, it becomes ever more apparent that we need to motivate students to be more interested in science, and interactive courseware which engages the students in hands-on learning and encourages exploration of the content may be the best way to do that. Maintaining interest is crucial in providing students with the best education, and I propose the design of an instructional system for the periodic table which eases the students into a deeper interest and understanding of chemistry. I plan to show the students the link between chemistry in their daily lives and their current curriculum.

An obvious solution to interfacing the hands-on approach of instruction with a computer program is through the use of radio frequency identification tags. RFID tags exchange their data with a RFID reader via radio waves, and the tags themselves can either embed minor amounts of data or be associated by their id with information via a database. The tags, which contain integrated circuits and an antenna for transmission, are small enough to be embedded within nearly anything.

For my RFID project I chose to embed the RFID tags in colored Styrofoam balls which in turn represent the atoms of various elements in the periodic table. Although unable to implant 117 Styrofoam balls with RFID tags, the ultimate objective is to have a large board laid out with the Styrofoam balls forming the structure of the periodic table, with each ball colored by its grouping in the periodic table. Then by grabbing a Styrofoam ball and waving it near the RFID reader, the program would pull up a picture and text explanation of the element and some fun facts about it as well. Users also have the option of mixing the elements together and discovering what compounds could be created using those elements, with information about the compounds available as well. The program obtained the information by associating the tag number obtained by the RFID reader to a database containing two tables: one for the elements and another for the compounds.

A brief survey with 30 participants was conducted on June 30th, 2008. The results were generally positive; with most users in agreement that such a program would be beneficial to very early chemistry curriculum. Things that people liked about the project: “Interactive and easy to use,” “the pictures used,” “cool concept and very easy to use,” “really easily marketable software/game,” and “hands on was good.” Criticisms on the project included comments on the graphical user interface, colors used within the program, the incomplete database, and lack of sound/video.

By going through the process of research, design, implementation, and obtaining user feedback concerning this program, I have learned much about the interaction between computers, humans, and education. Humans love to use computers, especially when the design is aesthetically pleasing as well as easy to use and understand. What humans do not like, especially when it comes to learning the early parts of chemistry, is long and complicated books. I consider it a high priority to bring a program like this to the K-12 curriculum in order to introduce students to the building blocks of life, the discovered elements of our known universe in a more fun and interactive manner.

Stochastic Timing Analysis of Programs with Multiple Paths

Paul Keim
Furman University
Greenvile, South Carolina
Paul.keim@furman.edu
Dr. Chris Healy, Computer Science

A real-time system is a computing system that operates on deadlines from event to system response, for example, an operating system, or a data collection system. When using real-time systems, the scheduler for the operating system has to know just how long each program will take. Failure for the scheduler to properly allocate resources can lead to anything from program failure, to total systems failure [1]. For example, if the real-time system in the car failed to react to the sudden and violent deceleration of the car, the airbags would not deploy, leading to possible harm. Therefore, it is important to know the Worst Case Execution Time (WCET) of a program or process. This can be determined in various ways.

A recent development in WCET analysis is the use of stochastic techniques. Bernat et al. first proposed this technique in 2002 [2]. They noted that traditional approaches for WCET analysis produce values which are very pessimistic if applied to today's processors. Therefore, they combined both measurement and analytical approaches into a model for computing probabilistic bounds on the execution time of the worst case path sections of code. Zolda [3] introduced the idea of letting the probability be variable for each path, but only considers the case of two paths, whereas this project focuses on cases with more than two paths.

In the analysis of these programs, they are decomposed into sections as follows: A program is comprised of functions, which are in turn comprised of loops, which are made up of instructions. Treating each loop as a node, an array of 100 values is created to show the distribution of the runtimes of each node. In a bottom-up manner, this distribution is determined for the program.

In this distribution, each element contains an execution time, while its position in the distribution represents the percentile of probability distribution for that runtime. As an example, for a program with a runtime ranging from 100 to 1,000 cycles, the 0th percentile (or the first element in the array) could represent 300 cycles, and the 99th percentile (The last element) could represent 600 cycles. Translated, this means that there is less than a 1% chance that the program will be under 300 cycles, and a 1% chance the program will be over 600 cycles.

Generating such a distribution for a program with one or two paths is quite simple: a program with one path can only run one possible way, thus its runtime will be a uniform distribution. A program with two paths is only slightly more complicated; by determining the probability of each path being taken, it is then a simple matter of finding the binomial probability distribution. However, once a program has three or more paths, determining the distribution becomes more complicated. Ordinarily, one considers all the paths together, then determining the probability, then finding the polynomial probability distribution. However, when programs start having more than five or six paths, this becomes inefficient. Therefore, we adapt the binomial probability formula into the algorithm outlined below:

Let $\{P_1, P_2, \ldots P_n\}$ be the set of paths. Each P_i has a probability of executing, p_i, and a probability distribution D_i. We create the overall probability distribution D as follows.

1. Use the binomial probability on D_1 and D_2 to initialize D.
2. For i = 3 to n, $D = D \oplus D_i$, where $\oplus$ is defined as follows:

Let $\oplus$ represent the binomial probability formula applied to the probability distributions of two paths. The distributions of these paths are weighted in the following manner:

a. Weight the number of elements. Suppose $p_1 < p_2$. Thus, D1 continues to contain 100 elements, and D_2 stretches to $(100{*}p_2/p_1)$ elements.
b. To create the extra elements for D_2, starting with the initial 100 elements, duplicate accordingly. For example, if needing 200 elements, duplicate each element. If needing 150 elements, duplicate every

other element.

The timing analyzer’s predicted distribution tends to be very broad because it considers the execution of any of the paths. Using a simulator to verify the results, differences between the projected distribution and the actual distribution (Figures 1 and 2, respectively) were noted. Note that the vertical scales in Figures 1 and 2 are different because the timing analyzer predicted a much wider distribution than was actually realized. It turns out that the absolute worst case path was not taken. Nevertheless, the peak of the probability distribution is in the same area as the simulated distribution, indicating that the prediction is close. One goal for the timing analyzer is to produce a more accurate upper bound. Other future work of this research includes expanding the approach to handle more complex structures, such as nested loops and programs with more than one loop.

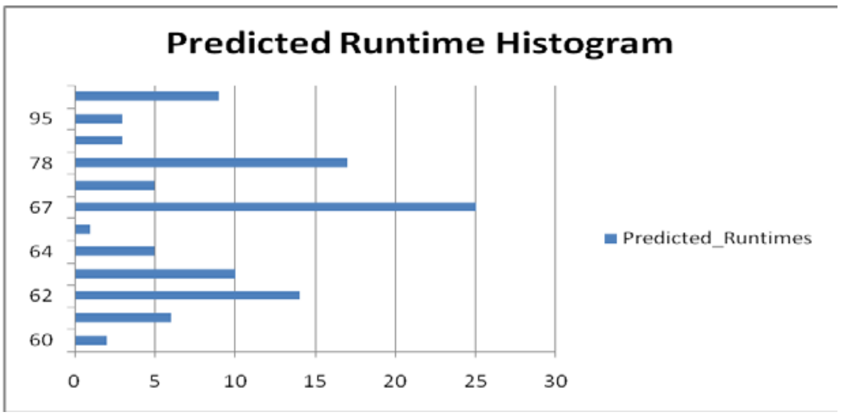


Figure 1: The x-axis represents the probability of the runtime, while the y-axes represents what percentage of the WCET that runtime is.

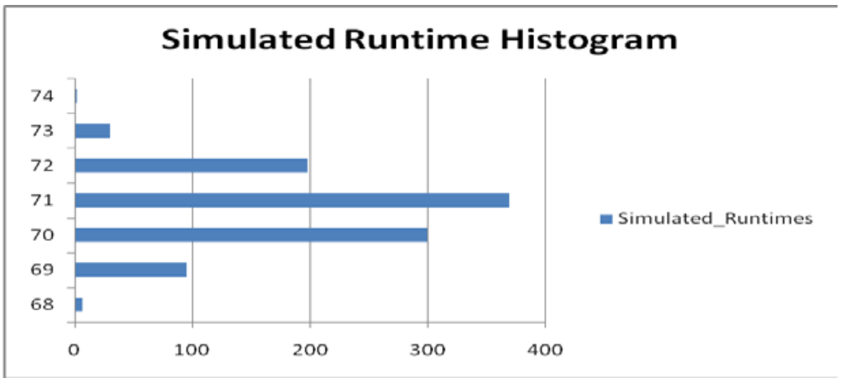


Figure 2: The x-axis represents the number of simulations, while the y-axis represents the percentage of the WCET.

References:

[1] A. Ermedahl, The worst-case execution-time problem—overview of methods and survey of tools. In *ACM Transactions on Embedded Computing Systems (TECS),* Vol. 7, issue 3, Article No. 36, April 2008.

[2] G. Bernat et al, WCET Analysis of Probabilistic Hard Real-Time Systems, in *Proceedings of the 23rd IEEE Real-Time Systems Symposium*, page 271, December 03-05, 2002

[3] M. Zolda. INFER: Interactivity Timing Profiles Based On Bayesian Network. In *Proceedings of the 8th International workshop on worst-case execution time (WCET) analysis*, pages 39-51, October 2008.

Intelligent Navigation System for Autonomous Sailboat

Bradley S. Neff
Computer Science
neffbs@wofford.edu
Wofford College

Mentors: Corey Ippolito, NASA Ames Research Center
Joseph Sloan, Wofford College

Abstract

There is a need for autonomous sailboats to be able to generate valid courses within polynomial time, and remain manageable within non-sparse environments, all while the sailboat is in motion. This can be accomplished by designing a three tiered system that would produce a waypoint tree within the known statespace of the sailboat, and add random points to the tree until an acceptable solution is found.

The statespace of the sailboat includes orientation, position, wind velocity and water current velocity. Possible unobstructed paths between points within the statespace would be provided by a linear controller calibrated to the sailboat. This controller also serves to sustain the boat between waypoints in the real-time portion of the system. The second tier consists of the Trajectory Management System, which sustains the sailboat on a given set of waypoints and handles any problems such as overshooting a waypoint.

The Trajectory Planner serves as the overarching intelligence for the system, and provides the best currently known path to the TMS, as well as continuously searching for improved paths. At the start, it is assumed that all obstacles in the statespace of the object are known, as well as the current position and the goal position. Random positions within the statespace of the object are then generated and connected to a previous node using as heuristic, developing a tree-like structure that will eventually span to the goal node, if it is possible.

Using Android to Promote Education in Mobile Device Development

Daniel Noler, Chris Baker
Mercer University
Macon, GA
danny_tcg@yahoo.com
Faculty Sponsor: Laurie White, Computer Science Department

In a rapidly developing world, cellular devices have quickly emerged as one of the fastest evolving technological fields. They have increased greatly in both popularity and complexity, requiring more advanced operating systems and applications to meet the demands of the consumer. Android is a software stack designed to meet these demands in an open source environment. The project is currently being developed and funded by the Open Handset Alliance, which includes companies such as Google and T-Mobile. Android includes an operating system, middleware, and key applications, as well as a Software Development Kit (SDK) for developers to create their own applications for the Android environment. Due to its open source license and tools provided, Android is an ideal platform for bringing the mobile market to the educational realm. This project is part of a greater effort to determine the uses of Android in the classroom through porting the operating system to hardware-supported devices.

At the time of the project, Android was unavailable in its full source code, as Google did not want to release the full source until an Android device was available on the market. However, it was still possible to port Android onto existing mobile devices. By using a compiled Linux kernel and applying the necessary Android patches, one can port Android to a real device and run a version of Android on the device. Android is very hardware specific when choosing a device to port to, and one must abide by the minimum hardware requirements. There have been many documented successful attempts at porting Android to real hardware. The full Android source was released on October 21, 2008.

The research and work involved in this study is designed to further a larger project. Headed by Dr. David Umphress and Dr. Richard Chapman of Auburn University, the overall project is designed to determine how Android can be used in education. Dr. Umphress first worked with Android in the spring of 2008, when he taught a course involving Android program development. The class proved successful based on student feedback, and the project proceeded to this project. The phase outlined here took over the summer of 2008 during an REU session at Auburn, and involved using Android as a means to educate on Linux kernel design and deployment.

This study involved porting Android onto two devices: a Nokia N800 and a Sharp Zaurus SL-C3200. These two devices were chosen because of their availability and hardware compatibility. Both devices meet the minimum hardware requirements and ran Android successfully, though neither device was fully functional given Android's limitations at the time. For example, the lack of an on-screen keyboard prevented the use of any text on the Nokia N800, and the lack of a Wi-Fi card in the Zaurus restricted Internet usage. Several subjects were covered during research required for porting, such as the architecture of Android, the anatomy of Android programs, and the use of open source tools. All development took place using Kubuntu Linux. Many of the tools used were also open source, such as OpenEmbedded and BitBake. Since Android is still under development, research was limited to the Internet as almost no published works on Android exist at the time. Therefore, most research was accomplished through the use of known Android information web sites provided by faculty accompanied by the use of search engines. Information on the Android platform itself was obtained from Google, which has created an in-depth manual for program development as well as message boards for communication with other Android enthusiasts.

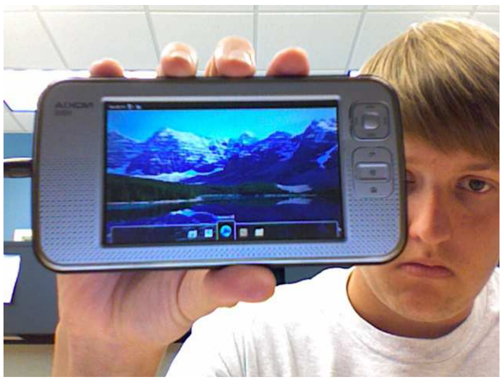


Figure 1: Chris Baker holding N800

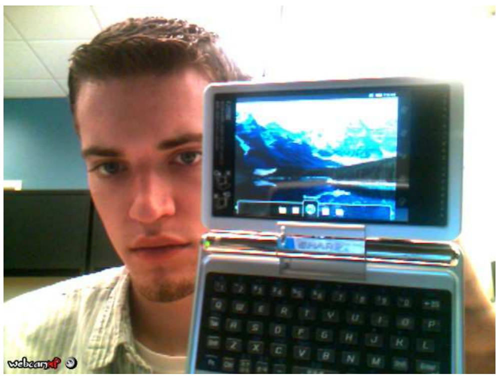


Figure 2: Danny Noler holding SL-C3200

There are a number of obstacles to overcome when porting Android. Other than the strict hardware requirements, one must also find all necessary patches online, and may even be required to create specific device drivers for hardware support. Most popular devices, however, have been hacked and most required files and patches are readily available over the Internet. Also, a number of tools may be required for the compilation of the Linux kernel, such as the GNU tool chain. Other tools may be needed for flashing ROM on particular mobile devices. However, the greatest obstacle is the learning curve. In order to fully appreciate Android porting and installation, one must gain an understanding of the Linux kernel, device driver programming, and a general knowledge of Linux command line tools. One of the major goals of this project was to determine if this obstacle could be used as a means for education. Based on our initial results, we believe that our work could be replicated in a classroom environment in order to educate others on these topics, especially as Android is available in its full source, which alleviates certain barriers present during this research. For example, a laptop was planned for use as an Android device. However, Android had not been compiled for the x86 architecture, though now it is possible.

Android is a very lucrative option for mobile device programming education. As it is further developed, more and more documentation will become available for new developers. Its open source nature provides the perfect environment for the educational realm, as there are no costs to create Android programs. Its use of Java allows for most beginning programmers to delve into the mobile world without a steep learning curve. This research project shows that, even in its incomplete state, Android is a very useful tool for mobile device development education, and also can be useful when learning about Linux and the open source tools available for it.

Presence in an Immersive Environment

Steven Nunnally
Roanoke College
Salem, VA
smnunnally@roanoke.edu
Faculty Sponsors: Dr. Stephen Hughes & Dr. Durell Bouchard (Computer Science)

Virtual Environments are immersive displays that are spaces that are not real, but may seem real to the user. These displays are great for interacting with a computer system in 3 dimensions, as the better Virtual Environments make the user feel much more a part of the environment. Presence and Virtual Environments are very important to researchers in the Virtual Reality field. Presence describes the level of immersion in the Virtual Environment, or how much the user feels a part of the Virtual Environment. Presence is currently not measurable objectively. Only subjective measures, like presence questionnaires, have been proven to work by comparing and ranking the tested Virtual Environments. An objective measurement is based on subconscious thought; therefore it would be reliable and would have the ability to compare Virtual Environments outside of the experiment based on other similar experiments.

Behavioral Realism is an objective measurement that judges the degree the user reacts to events on the Virtual Environment. Postural sway, a type of Behavioral Realism, is the theory that participants in an experiment will react to an event in a Virtual Environment with more magnitude if the Virtual Environment has a higher level of presence. The experiment outlined in this paper uses the reaction to centrifugal force as the event, and records the users sway against these centrifugal forces. This method would work for all Virtual Environments that are used for navigational purposes, opening up an objective comparison of presence among all of these environments. This would allow comparisons between any research worldwide that use this objective measurement, to find the Virtual Environment with the greatest presence. This would grant great advances in Virtual Environments, eventually allowing schools to use them for education, and home's to use them as an interactive interface for the internet.

Many other experiments before this experiment tried to prove that these objective measurements for presence would work to easily prove what level of presence the Virtual Environment is on. Using a video of a car racing around a race track, Freeman measured the sway of the person watching it. Freeman also used a questionnaire in order to correlate the sway with the questionnaire data, when they compared stereoscopic and monoscopic video (Freeman, Avons et al. 2000). Freeman concluded that their data showed weak support for the use of this behavioral realism in evaluating the separate displays. A follow up experiment worked with different display sizes to see if they were able to find similar correlations in the changes of the questionnaire and sway data. Again, the conclusions were weak, while the questionnaire data seemed to show what would be expected, that a larger screen increased presence (Freeman, Avons et al. 2001), since the objective measure did not correlate with this as strongly as they had hoped. Where his experiment failed, this experiment hopes to succeed by increasing the magnitude between the comparisons. The field of view change is a great difference, from 28 degrees for the small screen to 170 degrees for the big screen. Also, most researchers agree that the level of involvement is a strong factor in the level of presence, so the active participants should show a higher level of presence.

The experiment outlined in this project is very similar to Freeman's. The experiment is a 2 by 2 comparison. Every subject was administered the test on both the desktop display and the CAVE display, to set up a within subject comparison. This within subject should give good comparison results using the questionnaire, since this is a side by side comparison, giving the data a good basis when comparing the postural sway results. The control of the vehicle is a between subject display, so that participants are either active or passive. The questionnaire should still show a great difference since there are questions that ask the level of control the participant had over the actions on screen.

The participants' motion sway was recorded using 2 Wii sensor bars, as infrared lights, attached to the participant with Velcro, using suspenders to trace the shoulders and a head band to trace the head. The lights were recorded with a Wiimote, which passed the information in pixels to an infrared electronic whiteboard program which recorded the pixel coordinates of the 4 separate infrared lights. These coordinates are later used to analyze the postural sway.

The results from ten participants were very interesting for the research. The questionnaire data shows few results. In this case, the higher the results from the questionnaire the more present the user felt. The questionnaire data between big and small displays is not very different. The small display has a slightly better average. The average Active participant's rating is about 10 points higher than the Passive participant's, despite whether they were on the small or large screen.

The only conclusions from the behavioral realism results are that there are results there to be made. Using the method of analysis and graphing the researcher can decipher what movements are made in response to turns and what movements are shifts that have nothing to do with the information displayed in front of them. No conclusions can be made about this method of measuring presence objectively, although the results could be promising if some of these results became a little more defined with more subjects.

The future for this research is bright, but a few things need to change. Some of the methods must change to minimize problems with inconsistency in the IR recording program. Also the presence questionnaire needs to be changed so that an appropriate correlation can be made between small and large screen displays, simplifying the questionnaire might solve this problem. This preliminary test proves that an analysis of the data can be made, but a comparison of the data, not just visual comparison, must be developed to make better statistical data.

References

Freeman, J., S. E. Avons, et al. (2001). "Effects of Stereoscopic Presentation, Image Motion, and Screen Size on Subjective and Objective Corroborative Measures of Presence." Presence 10(3): 298-311.

Freeman, J., S. E. Avons, et al. (2000). "Using Behavioural Realism to Estimate Presence: A Study of the Utility of Postural Responses to Motion-Stimuli." Presence 9(2): 149-164.

Book Tracking Systems Based on RFID Technology in Libraries

Alexander Oplesnin, Georgia College and State University,
alexander_oplesnin@ecats.gcsu.edu

The field of my research conducted by Dr. Gita Phelps was the implementation of the passive RFID technology in library environment.

RFID Tag consists of an integrated circuit and an antenna combined to form a transponder. RFID tags collect the energy to operate from a Radio Frequency (RF) field emitted by a reader device; therefore they do not need a battery. When energized by a radio signal from a fixed position reader or handheld scanner, the tag returns the stored information in order that the item to which it is attached can be easily located.

Academic and public libraries challenge the problems in managing the collections assets and maintaining or improving service levels to patrons.

During the last twenty years, libraries have extended their services from simply lending the printings like books and periodicals to adding multi-media extensions stored on CDs and DVDs. Therewith, libraries nowadays provide a number of services to their visitors via the Internet. Along with the significant increase in resources provided by a library the concerns on the security of the collection assets and concerns on the services provided are also growing up. “Throughout this time, many municipalities and institutions who manage our library systems have sought ways to reduce operating budgets (cut staff) to offset the growing capital costs of the multi-media resources and computer hardware”.

The aim of my research was to find out the ways of how to speed up the operations performed by the library customers and as well the ones executed by its stuff.

I found out that the use of RFIDs compared to the use of magnetic stripes (the current method used in our library) is more fruitful in terms of the timing averagely spent on a common operation as check-in/check-out. In addition, the implementation of radio tags in libraries reveals lots of auxiliary features which were not available with the older methods, they are as follows:

- Frees staff to better service patrons
- Better space planning
- Increases membership rate
- Easy to use: books can be read in any orientation
- Reduces queuing time
- Provides user privacy
- Encourages users to come back

Description of the project:

Probably, each library user and each member of the library management could be given similar to the collection items’ one a tag for faster login procedure; in this case a user or

a librarian doesn’t have to deal with username and password entries, instead only one reading from his or her RFID is needed in order for the user to get in the library computer system. Information stored in a database may provide identification for an item, proof of ownership, original storage location, loan status and history. The data associated with every tag may be easily found within the database. The signal read from each tag will be the key to each corresponding record.
RFID tags have been specifically designed to be affixed into library media, including books, CDs, DVDs and tapes.
The tag itself is thin, flexible and thus can be attached to almost any surface.

The implementation of RFID in the library environment is obviously more fruitful than any other technology used for the same mentioned above purposes before because its faster in use, its contactless, and the probability of a radio tag being read incorrectly is very low.

TRIGGERS IN SQL

Poojakumari Patel
Kennesaw State University
Kennesaw, Georgia
ppatel46@students.kennesaw.edu
Advisors: Dr. Meg Murray, Computer Science
Dr. Mario Guimaraes, Computer Science

OBJECTIVE

This poster presents recent enhancements to the Triggers sub-module of the Animated Database Courseware (ADbC) project. Triggers are an integral part of Database Management Systems because they implement complex integrity constraints that cannot be done in a declarative form with the CREATE TABLE and ALTER TABLE commands. The Trigger sub-module can be accessed from the ADbC home page by choosing the Transactions Module and Trigger sub-module. The work is the result of an NSF CCLI funded project (#0717707).

BACKGROUND AND RELATED LITERATURE

Although there are many examples of triggers in database textbooks, very few supporting instructional materials to demonstrate these events are available. One example can be found at the Arizona State University, School of Computer Science and Engineering Department web site: http://www.eas.asu.edu/~advdb/. This program – accessed via the Curriculum Examples tab - allows the student to download or view a database example with triggers used in different environments. One limitation of this program is that it is static rather than interactive.

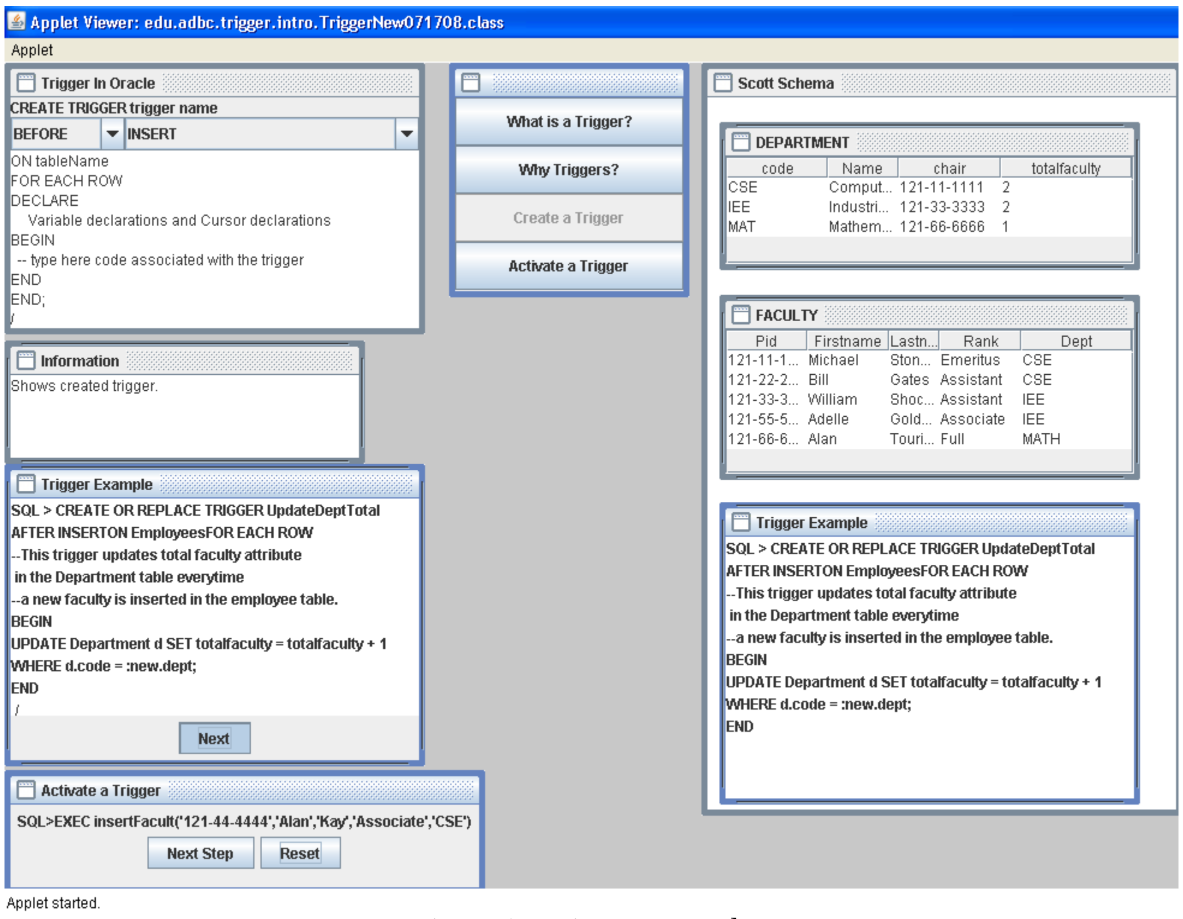
APPROACH AND UNIQUENESS

The Trigger sub-module was created using the Java programming language. It implements the AWT and SWING components of the Java library (http://java.sun.com/j2se/1.3/docs/api/). The animations were created with jdk 1.6 using the Eclipse development environment.

Java was a natural choice due to the need to make the animations accessible through the web. The development platform, Eclipse, and the versioning environment utilized, Subversion, were chosen by the students team developing the software. The goal was to implement a development environment similar to what is used in professional software development organizations. The development of the instructional software animations follows a process whereby students present their prototypes to other students and faculty at weekly meetings. Once the prototypes have been discussed and refined, they are tested with other students on an individual basis as well as used by instructors in the classroom environment.

RESULTS AND CONTRIBUTION

Recent modifications of the Trigger sub-module included redefining the button frame, adding a ‘Next Step’ function, defining a data schema frame, called the ‘Scott Schema,’, and adding an information frame. We are also working on developing the help menu to provide better support to users of the sub-module.



Applet Viewer: edu.adbc.trigger.intro.TriggerNew071708.class

Applet

Trigger In Oracle

```
CREATE TRIGGER trigger name
BEFORE  INSERT
ON tableName
FOR EACH ROW
DECLARE
   Variable declarations and Cursor declarations
BEGIN
 -- type here code associated with the trigger
END
END;
/
```

Information

Shows created trigger.

Trigger Example

```
SQL > CREATE OR REPLACE TRIGGER UpdateDeptTotal
AFTER INSERTON EmployeesFOR EACH ROW
--This trigger updates total faculty attribute
 in the Department table everytime
--a new faculty is inserted in the employee table.
BEGIN
UPDATE Department d SET totalfaculty = totalfaculty + 1
WHERE d.code = :new.dept;
END
 /
```

Activate a Trigger

```
SQL>EXEC insertFacult('121-44-4444','Alan','Kay','Associate','CSE')
```

Scott Schema

DEPARTMENT

code	Name	chair	totalfaculty
CSE	Comput...	121-11-1111	2
IEE	Industri...	121-33-3333	2
MAT	Mathem...	121-66-6666	1

FACULTY

Pid	Firstname	Lastn...	Rank	Dept
121-11-1...	Michael	Ston...	Emeritus	CSE
121-22-2...	Bill	Gates	Assistant	CSE
121-33-3...	William	Shoc...	Assistant	IEE
121-55-5...	Adelle	Gold...	Associate	IEE
121-66-6...	Alan	Touri...	Full	MATH

Trigger Example

```
SQL > CREATE OR REPLACE TRIGGER UpdateDeptTotal
AFTER INSERTON EmployeesFOR EACH ROW
--This trigger updates total faculty attribute
 in the Department table everytime
--a new faculty is inserted in the employee table.
BEGIN
UPDATE Department d SET totalfaculty = totalfaculty + 1
WHERE d.code = :new.dept;
END
```

Applet started.

Figure 1. Trigger Example

Figure 1 describes the actions that take place when a trigger is created and added to the database, as well as demonstrates what happens when a trigger is activated. The information window is updated every time the user clicks the *Next Step* button. The line of code that is executed is shown in the Triggers Example window and the modifications that occur are shown in the data schema window (in this particular example, the 'Scott Schema' window where data in the Department Table is modified).

Students have found this module useful to help them grasp the concept of triggers. This has been observed by testing the triggers on a one-on-one basis with students as well as reviewing end of semester course evaluations.

References:

[1] Dietrich, Suzanne W. and Urban, Susan D. An Advanced Course in Database Systems: Beyond Relational Databases. Available on-line at http://www.eas.asu.edu/~advdb/
[2] Guimaraes, Mario. "Database Courseware Functionality and Classroom Usage", ACM Mid South East Conference. November 2003, Gatlinburg, Tennessee.
[3] Guimaraes, Mario; "Database Courseware: an Update", Proceedings of 41st ACM-Southeast Conference, March , 2003, Savannah, GA.
[4] Guimaraes, Mario. "The Kennesaw Database Courseware (KDC) with an Evaluation Component". Proceedings of 42nd ACM Southeast Conference. April, 2004, Huntsville, AL.
[5] Murray, M. and Guimaraes, M. 2008. Animated database courseware: using animations to extend conceptual understanding of database concepts. *J. Comput. Small Coll.* 24, 2 (Dec. 2008)