Keith Brandt
Mathematics, Computer Science, and Physics
Rockhurst University
Kansas City MO, 64110
keith.brandt@rockhurst.edu

Margaret Richey
Department of Computer Science
University of Wisconsin
Madison WI, 53706
richeym@cs.wisc.edu

ABSTRACT

Mathematical induction is a proof technique used throughout mathematics, and recursion is a programming concept frequently used in computer science. This note will explore the parallel between induction proofs and recursive programs by providing several example problems that lead to an induction proof and a corresponding recursive program. We feel that students who are exposed to this parallel will gain a deeper understanding of both topics.

Key Words: mathematical induction, recursion, algorithms.

RAPTOR: Introducing Programming to Non-Majors with Flowcharts

Martin C. Carlisle, Terry A. Wilson[1], Jeffrey W. Humphries, Steven M. Hadfield
United States Air Force Academy
Department of Computer Science
2354 Fairchild Dr, Suite 6G149
USAFA, CO 80840-6234
{Martin.Carlisle,Jeffrey.Humphries,Steven.Hadfield}@usafa.af.mil

ABSTRACT

When students are learning to develop algorithms, they very often spend more time dealing with issues of syntax than solving the problem. Additionally, the textual nature of most programming environments works against the learning style of the majority of students. RAPTOR is a flowchart-based programming environment, designed specifically to help students visualize their algorithms and avoid syntactic baggage. RAPTOR programs are created visually and executed visually by tracing the execution through the flowchart. Required syntax is kept to a minimum. Students preferred using flowcharts to express

[*] Work performed while this author was a member of the faculty at the US Air Force Academy. Current address: ATR Technology Division AFRL/SNAR; Wright Patterson Air Force Base, Ohio; Terry.Wilson@wpafb.af.mil.

their algorithms, and were more successful creating algorithms using RAPTOR than using a traditional language or writing flowcharts without RAPTOR.

Keywords: Flowcharts, Visual Programming, Programming Environments, Problem Solving.

Virtual Reality for Small Colleges

Daniel C. Cliburn
Assistant Professor
Department of Mathematics and Computer Science
Hanover College
P.O. Box 890
Hanover, IN 47243
(812) 866-7286
cliburn@hanover.edu

Abstract

The popularity of Virtual Reality (VR) display systems has increased dramatically in recent years. VR applications are characterized by a virtual world into which users are immersed, providing interactivity and sensory feedback [1]. These superior sensory environments make virtual reality a popular medium for many types of applications, ranging from exploration of severe thunderstorm and tornado data [2], to architecture from ancient times [5]. Proponents of virtual reality feel that the VR experience enhances learning, since participants utilize more of their senses than with typical computer applications. VR systems generally provide displays to users in stereo projection that give a sense of 3D viewing. Traditionally, the cost of implementing virtual reality environment has been very high, limiting VR to only those institutions with large research budgets. However, with recent advances in commodity hardware capabilities and the reduction in cost of typical LCD projectors, a low cost VR system can now be constructed at any institution, even those with significant budget restrictions. This paper discusses our experiences with the construction of a stereo projection display environment at Hanover College, and some projects my students and I have done with this system. Several potential research projects in virtual reality, which could be done with little funding, are also proposed.

IMPACT OF OFFSHORE OUTSOURCING ON CS/IS CURRICULA

Ernest Ferguson
Northwest Missouri State University
2280 Colden Hall

800 University Drive
Maryville, MO 64468
ferg@mail.nwmissouri.edu

Keywords: computer science curriculum, offshore outsourcing, information technology and information systems curriculum.

Abstract

Reflection on the year 2003, historians might refer to the offshore outsourcing of information technology jobs as the megatrend of the year. As the IT industry matures, software is coming to be viewed as a commodity and computing as a utility. With custom software failing to deliver strategic advantage, the industry is rushing to seek the lowest labor costs for implementation, maintenance, support, and operations. This paper explores possible changes to computer science and information systems curriculum as a result of the outsourcing phenomenon.

ANTECEDENTS OF COMPUTER TECHNOLOGY USAGE: CONSIDERATIONS OF THE TECHNOLOGY ACCEPTANCE MODEL IN THE ACADEMIC ENVIRONMENT

Khaled Alshare
Professor of CIS
Accounting & Computer Information System Department
Emporia State University

Elizabeth Grandon
Professor of CIS
Accounting & Computer Information System Department
Emporia State University

Don Miller
Professor of Management
Business Administration & Education Department
Emporia State University

ABSTRACT

Several factors have to be taken into account in order to guarantee computer usage. In this study, we focused attention on some of these factors and created a theoretical model to better understand the relationship between computer literacy, attitudes, perceived ease of use, and perceived usefulness as determinants of computer usage. Additionally, we examined the impact of some external variables such as gender, income, and others on

computer literacy, perceived ease of use, perceived usefulness, and attitudes toward computers. We studied these relationships and investigated how gender and educational background mediate the hypothesized relationships. To validate the research model, we collected data from 166 students at a regional Midwest university. Finding support for the proposed model is of vital importance for organizations that can make better decisions when facing employee training issues and also for scholars and curriculum administrators. The results showed that gender, traditional vs. non-traditional students categories, educational background (business vs. non-business), classification (full-time vs. part-time) were not significant factors in affecting students' computer usage. However, income, self-reported measure of computer knowledge, perceived ease of use, perceived usefulness, computer literacy, and attitudes toward computers (positive vs. negative) were significant factors that impacted students' computer usage.

A TEAM TEACHING APPROACH TO INCLUSION OF A SECURITY COMPONENT IN IS AND CS CURRICULA

James Harris, PhD, CISSP
jdharris@pittstate.edu

Maeve Cummings, PhD
cummings@pittstate.edu

Felix Dreher, PhD
ffdreher@pittstate.edu

Kailash Chandra, PhD
kchandra@pittstate.edu

Department of Computer Science/Information Systems
Pittsburg State University
Pittsburg, KS 66762

ABSTRACT

Issues related to computer security and information assurance are important to everyone who uses information technology and these issues will have a substantial impact on the careers of students pursuing degrees in computer science and information systems. The onus, therefore, is on universities and colleges to include IT security in computer-related curricula. This presents a challenge especially for small departments where faculty members already have full teaching loads supporting service courses and other aspects of the degree programs. This paper describes a team teaching approach used to provide a course in computer security and information assurance for upper

level undergraduate students majoring in computer science or information systems. The course format, lab environment, and lab exercises are described. A survey of students shows that students were very receptive to this approach. The team teaching approach allowed the workload required for preparation of the new course to be distributed among several faculty members and it allowed each faculty member to focus on aspects of the course most closely related to their areas of interest.

THE IMPORTANCE OF PRESENTATION SKILLS IN THE CLASSROOM: STUDENTS AND INSTRUCTORS PERSPECTIVES

Khaled Alshare
Emporia State University
1200 Commercial Street-Campus Box 4057
Emporia, KS 66801
alsharek@emporia.edu

Nitham M. Hindi
Texas State University / San Marcos
101 Derrick Hall
601 University Drive
San Marcos, TX 78666
nh12@txstate.edu

ABSTRACT

Employers are demanding graduates with excellent communication (written, oral, and listening) skills. Thus, a student's presentation in the classroom becomes an important element in delivering positive learning experiences. This paper explored the role of students' presentation in the classroom from students and instructors' perspectives. The results of the study showed that forty-three percent of respondents indicated students were required to present in two classes per semester for an average of 10 minutes. Students and instructors agreed that critical objectives of presentations were to improve communication skills and to train students to talk to a group of people. However, the two groups differed on ranking the presentation evaluation. While instructors ranked "the content of presentation" in the first place, students ranked "the organization of the presentation" as their first choice. Both groups, however, agreed that "well-organized presentation" and "enjoyable content" were the two most important measures of presentation effectiveness. Students' responses showed no statistical significance between the grade assigned and the grade that should be assigned for presentation. While the majority of graduate students (70 percent) reported 11-20 minutes for presentation period, undergraduates

indicated 10 minutes (68 percent). Undergraduate students thought handouts were the most important visual aid compared to graduate students who preferred PowerPoint. The paper concluded by providing a checklist of items for effective and ineffective presentations.

Keywords: communication skills, presentation skills, and effective communication.

TESTING THE EFFECTIVENESS OF A SECURE CLASSROOM INTRANET

Nicholas Karlson
Department of Information Science
College of Information Science and Systems Engineering
University of Arkansas at Little Rock
2801 S. University Ave., Little Rock, Arkansas 72204-1099
E-mail: nekarlson@ualr.edu

ABSTRACT

A classroom intranet fosters student collaboration and community. In addition, an always-available intranet helps students remain actively involved with group work while outside of class. This paper discusses an example of a classroom intranet technology along with its associated underpinnings. A statistical analysis of data from an experiment designed to test the technology's effectiveness concludes the paper.

WHY MATRIX MANIPULATION SHOULD BE DONE IN C++

W. Douglas Maurer
Department of Computer Science
George Washington University
Washington, DC 20052
202-994-5921
maurer@gwu.edu

ABSTRACT

We show here how the technique of overloaded function call in C++ may be used in making matrix manipulations easier to understand, even when these involve specialized matrix representations. Such representations include sparse matrices, packed Boolean matrices, tridiagonal matrices, and arrays of pointers

to arrays; and these may start with m_{00} or with m_{11}. In all these cases, we may use the notation `m(i,j)` (with parentheses, not square brackets) to denote m_{ij}. The overloaded function call is available in C++ and in no other language, and this is why matrix manipulation should be done in C++. In many cases an auxiliary construction, that of the reference class, is necessary in order to allow matrix elements to be changed as well as used.

THE USE OF A MULTIMEDIA LESSON TO INCREASE NOVICE PROGRAMMERS' UNDERSTANDING OF PROGRAMMING ARRAY CONCEPTS

Jim McKeown
Dakota State University
309 East Hall
Madison, SD 57042
Jim.McKeown@dsu.edu

ABSTRACT

The methods used to teach introductory computer programming to college students are becoming outdated. The use of a multimedia lesson, delivered prior to classroom lectures or following classroom instruction was investigated for its effectiveness in developing the mental models of students. This was a quantitative study of the effects of a multimedia lesson on the mental models and conceptual knowledge of novice computer programming students in an introductory programming course. One group received a multimedia lesson prior to lectures. The other group received instruction prior to using the lesson. The order of the treatment made a significant difference on their pretest or posttest scores. Both groups saw significant gains in their posttest scores. Students in both groups also experienced significant improvement in their mental models of programming arrays.

Keywords: constructivism, programming, mental models, multimedia and instructional design.

PROGRAMMATICALLY TESTING CONCURRENCY IN J2EE WEB APPLICATIONS

Brian Mulanda and Bruce Mechtly

Department of Computer Information Sciences
Washburn University
Topeka, KS
brian.mulanda@washburn.edu, bruce.mechtly@washburn.edu

ABSTRACT

In this paper we present a Java GUI application and a corresponding web application that demonstrate data corruption in non thread-safe J2EE web applications. The application allows the user to create a number of threads that will cause nearly simultaneous connections to a web server, simulating use of the web server concurrently by multiple clients. Our web application contains deliberate design flaws so that it will show data corruption under even moderate concurrent use. We discuss the conditions under which corruption occurs and show how to make thread-safe web applications using servlets and JavaServerPages.

VIRTUAL SCHOLARLY COLLABORATION: A CASE STUDY

Bruce J. Neubauer
Pittsburg State University
223 Kelce College of Business
1701 S. Broadway
Pittsburg State University
Pittsburg, KS (USA) 66762-7530
E-mail: bneubaue@pittstate.edu

Gene A. Brewer
The University of Georgia
School of Public and International Affairs
303A Baldwin Hall
Athens, GA (USA) 30602-1615
E-mail: cmsbrew@uga.edu

ABSTRACT

Scholarly collaboration can certainly offer the potential to increase the productivity of scholars but can carry with it burdens and challenges not experienced by those who choose to write and publish individually. As scholarship itself becomes more complex the benefits of collaboration are likely to become more significant. Information technology can be used to facilitate scholarly collaboration over a distance in ways that were not available to scholars of previous generations. This is a case study, reporting and reflecting upon the experiences of authors in using information technology to facilitate their collaborative efforts. Parallels between scholarly writing and software programming are suggested.

Key words: collaboration, scholarship, video conferencing, instant messaging, case study

INTEGRATING OO CONCEPTS INTO A CS0 COURSE

Chuck Pheatt
Emporia State University
250A Science Hall
1200 Commercial
Emporia, KS 66801
pheattch@emporia.edu

ABSTRACT

This paper describes the use of field programmable integrated circuits (FPIC) in introducing object oriented (OO) programming concepts into a CS0 Course. Using a low cost device known as the OOPIC (Object-Oriented Programmable Integrated Circuits), students can easily control hardware circuitry while being exposed to OO programming concepts. The OOPIC device simplifies hardware programming by allowing students to use common programming languages (Visual Basic, Java, or C) and a simple development environment to create and test programs. In addition to introducing the use of an OO approach to problem solving, the device provides students an opportunity to deal with simple circuits.

Keywords: CS0, field programmable integrated circuits, OOPIC.

REASONS WOMEN PURSUE A COMPUTER SCIENCE CAREER: PERSPECTIVES OF WOMEN FROM A MID-SIZED INSTITUTION

Jennifer Kahle
Washburn University
1700 SW College Avenue
Topeka, KS 66621
jennifer.kahle@washburn.edu

Dr. Gary Schmidt
CIS Department
Washburn University
1700 SW College Avenue

Topeka, KS 66621
gary.schmidt@washburn.edu

ABSTRACT

There is not a substantial number of women in the computer science field. Even though computer science offers a great deal of professional opportunity, statistics show women have not been taking advantage of its many benefits. The purpose of this paper is to determine why women are entering the computer science field and to suggest ways to increase enrollment of female computer science majors in a specific category of learning institution. Questionnaires were given out asking open-ended questions about educational background, interest in the computer science field, and encouragement in pursuing a computer science career. The questionnaires were given out to women whom are in various stages of their computer science career. Responses were quantified into no more than four categories. Inferences were made based upon the information gathered from the responses.

Going Wireless:
The Emergence of Wireless Networks in Education

MS Leslie Moody
Washburn University
1700 SW College Avenue
Topeka, KS 66621
leslie.moody@washburn.edu

Dr. Gary Schmidt
CIS Department
Washburn University
1700 SW College Avenue
Topeka, KS 66621
gary.schmidt@washburn.edu

Abstract

This paper focuses on the emerging technology of wireless networks, and their use in education. This paper discusses some advantages of wireless networks, as well as some concerns that need to be addressed before implementing a wireless network. Wireless networks are being implemented in both K-12 institutions and post-secondary institutions, though it has been found that they are being used in a different manner. In K-12 settings they are being used primarily to integrate technology and its use into the classroom setting, replacing the standard computer lab that

is located outside the classroom. At the university level, the wireless network is utilized more as a means of convenience. The students and teachers are able to access resources for learning regardless of their location. They can use laptops and PDA's to expand their learning capabilities to any location, not just the classroom. The ideal wireless network needs to bring more technology into the classroom and in turn enhance the learning experience, as well as provide the opportunity for students to access resources outside the classroom, which would provide those who are interested a way to obtain knowledge any time they are willing and able.

APPLICATIONS SOFTWARE PROGRAMMING

James T. Streib
Department of Computer Science
Illinois College
Jacksonville, IL 62650
jtstreib@ic.edu

James W. Chaffee
Office of Educational Technology
Illinois College
Jacksonville, IL 62650
jchaffee@ic.edu

ABSTRACT

The purpose of this paper is to consider various aspects of applications software programming. In particular, programming in Visual Basic for Applications (VBA) is examined and various samples of code are given. The implementation of a service course for students not majoring in either computer science or management information systems is discussed. Lastly, the results of an informal survey are presented as part of the conclusion.

SOFTWARE ENGINEERING BASED ON THE TEAM SOFTWARE PROCESS WITH A REAL WORLD PROJECT

Nasser Tadayon
Department of Computer and Software Engineering
College of Engineering
Embry Riddle Aeronautical University
600 S. Clyde Morris Blvd, Daytona Beach FL 32114-3900

tadayonn@erau.edu

ABSTRACT

It can be observed the increasing demand for experienced professionals in development of software. On the other hand we see the decrease of interest in students and decline in student's enrollment in computer science and software engineering in most of the colleges and universities in the United States. The outburst of technology in our society demands software engineers to handle the exponential growth. Software plays very important, if not a critical, role in our daily life. It is one of the most rapidly growing fields in Engineering and technology. As software engineering educators, we bear the obligation of attracting and maintaining the interest of students by exploring and identifying initiatives that excite and draw them to the discipline. One tactical move for undergraduate students to understand the concepts and see themselves in their future role is having a realistic customer with real expectations in a software engineering course. There have been several papers exploring the same idea [1, 2, 6]. In this paper, the instructor describes the process and the structure in building a semi-realistic term project, using an external customer for a sophomore level introductory course in software engineering. The course is designed for relatively small groups of traditional and nontraditional students. The instructor assert his efforts in incorporating teamwork with diverse students as well as the challenging experiences that the students faced while working in the semi-real world industrial environment.

Keywords: TSP, software engineering, curriculum, real world project.

A COMPARISON OF UNDERGRADUATE STUDENT PERFORMANCE IN ONLINE AND TRADITIONAL COURSES

Dr. Gary Ury
Northwest Missouri State University

ABSTRACT

As online instruction becomes more prevalent at the college and university level, researchers are attempting to measure the success of these programs using a variety of methodologies, instruments, and sample sizes. There is a need for continued exploration and study to assure quality instruction. The purpose of this study was to compare course performance over time between online and traditional classroom students enrolled in a required management information systems course included in the business school's common professional component. The online delivery method was found to be effective, but performance, as measured by final course grades, showed a significantly

lower mean score than students enrolled in traditional sections of the course. Other findings indicated that traditional course designs can be adequately adapted to the online model and that collaborative course development can lead to a level of consistency in student performance within high demand courses.

---

nifty Assignments

A Software Engineering Code Obfuscation Project

Charles Ashbacher
Mount Mercy College

1330 Elmhurst Drive
Cedar Rapids, Iowa 52402

Charles Ashbacher Technologies
Box 294
Hiawatha, Iowa 52233

E-mail: cashbacher@yahoo.com

Abstract:
In the Spring term of 2003, I taught the upper level computer science major class CS 400 Software Engineering at Mount Mercy College for the first time. Rather than present a series of formulas of software creation metrics and diagrams of software development strategies, I decided to have the students build a significant and useful project. My choice was to have them build a code obfuscation program in the language Java. This choice of language was due to the many methods available to perform String processing, which is essentially what the project was. This paper is a report on that project as well as an announcement of the open source availability of the code.

Our first step was to write a series of coding specifications for the project, and cover enough of the Unified Modeling Language (UML) so that we had a common set of conditions and a common language used to represent the design. The class was split into two groups of five and the project went through four iterations. For each iteration, there were two separate parts and at the start, each group was given a description of the requirements for one part of the iteration. The group then wrote the specifications for that part and then each handed the other their specifications, so each group wrote code conforming to the other group's written specifications. At the end, each group was required to submit complete formal documentation for all their code and the groups submitted a working copy of the code obfuscator containing the code integrated from both teams. Over 100 pages of documentation were submitted and the final program is estimated to be over 10,000 lines of code, which includes internal documentation.

A Web Spider Project for an OOP Course

Jeffrey D. Martens
Department of Computer Science
Hood College
401 Rosemont Avenue
Frederick, MD 21701
jmartens@cs.hood.edu

Introduction
This programming assignment involves the incremental design and implementation of a web search tool as a semester-long project implemented in four phases. Further discussion, UML diagrams, and Java source code will be available on the author's web page[1].

First Phase: Display URL Text
For this phase, the students were required to write a Java program to obtain a URL from the user, download the page corresponding to that URL using Java's `URL` and `URLConnection` classes, report the number of bytes obtained via a simple character count and via `URLConnection.getContentLength()`, and explain any discrepancy in the sizes. The instructor's design consisted of a single class, `PageGetter.`

Second Phase: Display Links on a Page Alphabetically, and Verify Their Liveness
In this phase, students are asked to augment the first phase to find all the links on a page, alphabetize them, and verify that each is currently accessible. Each student defines and uses a regular expression to recognize HTML links.

Most students based their phase two solutions on their own phase one solutions, but they were also allowed to use the instructor's phase one as a starting point. The students were given this option for each phase. The instructor's solution involved minor modifications to `PageGetter` *and a new class,* `LinkVerifier.`

Third Phase: Honor Robot Exclusion
The robot exclusion protocol[2] allows system administrators to specify files and paths that are off-limits to automated search tools, and allows web page authors some control over whether robots may index or follow links on a particular page. Students are told to read each server's `robots.txt` file only once, forcing students to cache the information for each server.

The instructor's solution adds one class, `RoboChecker`, which is a subclass of `LinkVerifier`. `RoboChecker` may be thought of as a `LinkVerifier` that, for each link discovered, determines whether that link can be indexed or followed. Meanwhile, the instructor's `LinkVerifier`'s `main()` has been modified to support unit testing.

Fourth Phase: Depth- and Breadth-First Search
In the fourth phase, students were to provide the user with a choice of depth- or breadth-first search for a particular string beginning at a specified URL. To keep execution time and memory usage finite, the program is given a maximum number of web pages to visit.

The instructor's solution defined a `GraphSearcher`, which can be thought of as a `RoboChecker` that actually follows links. `GraphSearcher` itself is abstract, having two subclasses, differing primarily in how the links are stored: `DepthFirstSearcher` uses a queue, whereas `BreadthFirstSearcher` uses a stack.

Possible Variations

With the above project, the students end up with a tool that will search for a particular string starting at a specified URL. The resulting code, or the assignment itself, can easily be modified to be a tool that provides:

- a list of broken links on a page, or on a set of pages;
- a spider that limits its search to one server or a given set of servers;
- a tool that lists or downloads files of a particular type, e.g., JPEGs, found on a page;
- a tool that downloads a single specified URL.

References

1) `http://cs.hood.edu/~jmartens/ccsc/2004/centralPlains/`
2) `http://www.robotstxt.org/wc/exclusion.html`

Team Game Programming Assignment

Jim McKeown
Dakota State University
309 East Hall
Madison, SD 57042
Jim.McKeown@dsu.edu

ABSTRACT

Students in the Advanced Visual Basic Programming course must complete a group project to develop a computer game. While the topic does not fit exactly in a business programming course, for the students it is a good exercise in software development. Students enjoy the game aspect and the development, but from a learning perspective, the development process is more valuable. Most of the work and development must be completed on their own. They work as a team and become responsible, not only to themselves, but to the other members of the team as well. The students work in teams of three for the project. Student ideas for a computer game are submitted in advance. The games can range from simple card games like Solitaire and Black Jack, to parlor games like 3-D Tic-Tac-Toe and Chutes and Ladders, to games of chance like Craps and Roulette. The programs must have online help screens available, storage for winners and scores and an option to play against the computer. Each student writes a proposal and a description of the game they would like to

program. Students teams are drawn randomly and the student projects are selected randomly from the proposals submitted for that group. From there, one team member writes a complete set of rules for the game. Another writes the requirements for the program and a third creates the algorithm for the program. Each then critiques the work of the other team members and they submit their plan for approval. Once approved the group develops their software. Each team member must log their share of work on the project and each team member rates the other team members. The projects are scored and the students are graded.

Panel on Offshore Outsourcing: Impact on CS/IS Curriculum

Clifton Dunn
Computer and Information Sciences Dept.
Southwest Baptist University
Bolivar, MO 65613
417-328-1701
cdunn@sbuniv.edu

Mustafa Kamal
Computer Information Systems Dept.
Central Missouri State University
Warrensburg, MO Warrensburg, MO 64093
660-543-X4243
KAMAL@cmsu1.cmsu.edu

Ernest Ferguson (Moderator)
Dept. of Comp Science / Info Systems
Northwest Missouri State University
Maryville, MO 64468-6001
660-562-1551
ferg@mail.nwmissouri.edu

Sri Siva
Dept. of Comp Science / Info Systems
Northwest Missouri State University
Maryville, MO 64468-6001
660-562-1600
sri@mail.nwmissouri.edu

Keywords: curriculum, offshore outsourcing, offshoring.

INTRODUCTION

In November 2002 Forrester Research report [6] predicted that at least 3.3 million white-collar jobs and $136 billion in wages are expected to shift overseas by 2015, of which nearly one million would be IT-related. This was brought closer home when headlines in the *Kansas City Star* detailed the extensive layoffs of Sprint IT employees due to outsourcing [7]. According to Evans Data Corporation, 4.0% of outsourcing work was done offshore a year ago while 9.6% of outsourcing work is done offshore today [1]. The U.S. Bureau of Labor Statistics reported that 2.1% of U.S. computer and math workers were unemployed as of July 2000. That number had risen to 5.6% by July 2003 [1].

This panel will explore the current state of offshore outsourcing and its impact on computer science / information systems curricula.

CLIFTON DUNN

Overseas outsourcing is a hot topic in software engineering but has been festering for 15 years. Many companies looking for the short-term biggest bang for the buck turn to outsourcing. Good intentions by managers and stockholders are being manipulated by presenting a brighter picture than actually exists. The costs are usually so under estimated that the outsource decision is little more than a rubber stamp to a cost savings proposal. Many of the long-term costs of outsourcing are not directly measurable and get easily overlooked, either on purpose or accidentally, when comparing the alternatives. Things like post-release support, customer satisfaction, product education, documentation, overseas communication, project management, code design and review, plus cultural and language differences are just a few of the long term hidden costs for outsourcing. Additionally this form of code development will result in a close tie between the company and the outsourcer, which could affect the success or failure of both companies.

Even with these pitfalls outsourcing does have its place. If a project can be completely isolated, is not mission critical, does not directly interface with other internal or vendor code, and one is willing to entrust control of support to others, then the project may have a chance for outsourcing success. Smaller companies or startup companies may have a more legitimate business case for outsourcing since they probably do not have an established code base, have immediate financial constraints, or have a one-time need. One of the most frequently cited reasons for a large company to outsource is time-to-market. Outsourcing often frees up existing staff to work on more critical applications. The availability of qualified IT personnel is an issue that affects the outsource desire as well.

Given the changing workforce, colleges and universities need to adapt. There is demand for people who can manage outsourced projects, and colleges need to provide graduates that have some educational background in management of distributed projects. The company that is trying to outsource and the outsource company will both demand these graduates. At present most project management assumes that you are part of a local in-house team and have direct access to a group of people with the same goals and development structure as the project manager. Outsourcing works just the opposite; the development team may be half way around the globe, access can be limited and indirect, and most importantly the tools and development goals may be vastly different than would be used by a local project team.

3. ERNEST FERGUSON

It does not appear to be a question of "will offshore outsourcing continue as a practice," but rather how common will it be? Gartner Inc., an information technology research firm, calls the movement of tech-related jobs an "irreversible megatrend" [3]. According to Gartner, "by 2004 more than 80% of all U.S. companies will have considered shifting U.S. IT jobs overseas, while 40% of all U.S. organizations will have completed some type of pilot or will source IT services from non-U.S.-based service providers" [5].

A recent New York Times article stated that, "Low-skill jobs like coding are moving offshore and what's left in their place are more advanced project management jobs" [4].

This shift in jobs mandates a greater emphasis on communication skills both oral and written, project management integrated throughout program, development of business savvy, expanded treatment of risk assessment, security, and CMM. Computer science departments should encourage students to take a foreign language and a multicultural course.

MUSTAFA KAMAL

In an information-based economy, firm assets are primarily intangible. They consist of employee knowledge, database, R & D, ideas, etc. These intangible assets are what give a firm value, as opposed to tangible assets like plant and equipment in a manufacturing-based economy. But if US companies shift these intangible assets – knowledge jobs -- overseas, how much value is lost in USA national economy as well? A recent article in the Business Week (Feb 3, 2003) raised the above question and concerns that received much attention. The truth is that the rise of the global knowledge industry is so recent that most economists haven't begun to fathom the implications. For developing nations, the big beneficiaries will be those offering the speediest and cheapest telecom links, investor-friendly policies, and ample college graduates. In the West, it's far less clear who will be the big winners and losers.

There should be absolutely no doubt in anyone's mind that when a segment of IT works can be done by highly skilled IT workers for a few thousand dollars salary a year, the jobs will migrate overseas. How can US workers compete with equally qualified professionals in developing countries for a fraction of the salary? The following table shows average salary offered to an IT professional with 2-3 years of experience.

Average Monthly Salary in US dollars [8]			
USA	Canada	India	China
$4000	$3000	$700	$500

While the exact number of jobs lost in the US due to outsourcing is disputable, the number of jobs created overseas is colossal and its impact on the socio economic life is enormous. The following table shows the number of jobs created in some parts of the world due to outsourcing.

Country [2]	**Number Of Jobs Added**	**Projected Growth In Next 5 Years**	**Foreign Direct Investment**
China	3.5 million	3 to 3.5 times	$53 billion
Hong Kong/ Taiwan/ Macao	3.25 millions	3 to 3.5 times	Not known
India	650,000	3 -4 times	$6 billion

Outsourcing is not a reversible processes as less than 1% jobs come back to their original destination. What about future of all those college graduates who will be looking for a job? Can US students compete with the knowledge workers overseas? In general, average skills in areas of math, critical thinking, problem solving, geography, science, etc of US students are far below that of many of these students overseas that will compete with our knowledge workers.

In spite of growing popularity for outsourcing in IT sector among industries, the concept is under scrutiny by citizens and politicians alike. Recently Indiana governor has cancelled its 15 million worth outsourcing project with Tata International, an India based Information Technology service provider because it felt that the IT service provider wasn't planning to use US resources as agreed. In the state of New Jersey, certain senators are attempting to create legislation to ban state contracts form being outsourced to overseas companies.

Nasscom, of India defends that outsourcing has not caused layoffs but, on the contrary, has helped U.S. companies to avert them. U.S. banks, financial-services companies and insurance firms have saved $6 billion in the past four years by outsourcing work to India. During that time, part of the money saved via outsourcing has gone toward the addition of 125,000 jobs at those institutions. "Indian IT companies have [contributed} and continue to contribute to the U.S. economy by employing nearly 60,000 people in the U.S. in 2001," Nasscom states. "Nearly 170 Indian IT companies have physical establishments in the U.S."

SRI SIVA

The main reason companies decide to outsource is a financial one. When IBM on December 15th, 2003 announced that they are moving 4700 High Tech jobs overseas, shares were up on that day (source: www.cnn.com). This clearly indicates that even to the common shareholder, they are able to perceive the tremendous cost savings when a company decides to outsource work to offshore locations. This is just the beginning of the clear impact we see on the IT sector. What is the impact on students in the United States who are currently majoring in IT related fields? How do we as an institution still continue to sustain enrollment in our department?

REFERENCES

[1] *Baseline*, September 1, 2003. pp.28-29. http://common.ziffdavisinternet.com/download/0/2216/Baseline0901-p28-29.pdf.

[2] Einhorn, Bruce. The Quest for Asia's Outsourcing Crown, *Business Week online*, July 30, 2003.
[3] Johnson, Maryfran. "Riding The Offshore Wave" *Computerworld*, September 15, 2003, http://www.computerworld.com/managementtopics/outsourcing/story/0,10801,84860,00.html
[4] Kinetz, Erika. "Who Wins and Who Loses as Jobs Move Overseas?" *New York Times*, December 7, 2003.
[5] Leung, Linda. "Taking IT Offshore: Where Are The Hot Spots For Offshore Outsourcing?" *Network World*, August 25, 2003. http://www.nwfusion.com/careers/2003/0825man.html.
[6]. McManes, Chris. "H-1B And L-1 Visas Accelerate Offshore Outsourcing," http://www.ieeeusa.org/newspubs/features/070703.htm, July 7, 2003.
[7] "Sprint Will Outsource Tech Jobs, Local Work May Head Overseas," *Kansas City Star*, September 16, 2003, pp. A1.
[8] Vijayah, Jaikumar. US Firm Look North For Outsourcing Help, *Computer World* Feb. 25, 2002.

Tutorial: Teaching an Introductory Computer Graphics Course Using OpenGL

Daniel C. Cliburn
Assistant Professor
Department of Mathematics and Computer Science
Hanover College
P.O. Box 890
Hanover, IN 47243
(812) 866-7286
cliburn@hanover.edu

Abstract

In the past, introductory graphics courses focused traditionally on algorithms for drawing points, lines and polygons using low-level or system specific commands. This approach has become dated with the introduction of OpenGL and other high level 3D graphics application programming interfaces (API). OpenGL is an API that provides access to graphics hardware through functions common to 3D graphical applications. This tutorial focuses on using OpenGL to teach a junior/senior level elective course in computer graphics. Both OpenGL programming techniques and the design of a high level 3D graphics API such as OpenGL will be covered.

HCI In The Classroom

Daniel D. McCracken
Computer Science Department
City College of New York
137th and Convent Avenues
New York, NY 10031
mccracken@cs.ccny.cuny.edu

Human-Computer Interaction is a recognized field in the CS curriculum, as of Computing Curricula 2001. But what is it? Can you teach it without a formal background in the subject? This workshop addresses both questions, through a combination of lecture/demonstrations, group exercises, and discussion of how to fit HCI into an

already-crowded CS curriculum. Each participant may request a copy of User-Centered Website Development: A Human-Computer Interaction Approach, by Dan McCracken and Rosalee Wolfe (DePaul University). All will receive a CD containing instructor supplements, including a complete set of PowerPoint slides. NSF support is gratefully acknowledged.

COPYRIGHT, LAW AND ETHICS ON THE WEB – ISSUES FOR THE COMPUTING EDUCATOR

Dr. Rebecca H. Rutherfoord
Professor and Department Chair, Information Technology
School of Computing and Software Engineering
Southern Polytechnic State University
Marietta, GA 30060
770+528-7400
brutherf@spsu.edu

WORKSHOP/TUTORIAL ABSTRACT

Most educators are familiar with copyright and fair use practices. However, there are new considerations for copyright and fair use on the web. Copyright includes seven different categories, many of which are applicable to presentations and papers on the web. The seven categories are: 1) literary works; 2) musical works, including words; 3) dramatic works, including music; 4) pantomimes and choreographic works; 5) pictorial, graphic and sculptural works; 6) motion pictures and audiovisual works; and 7) sound recordings. Educators need to become aware of the implications of using any of these seven categories on the web. Other issues pertaining to law are patents, trademarks and trade secrets.

Computing educators need to become aware of the new laws pertaining to copyright. The 1996 Telecommunications Act and the 1998 Digital Millenium Act are important for their implications for copyright. The newest TEACH act has implications for on-line and web courses. Some of the newest information concerning both of these acts shows that there may be some contradiction between the two. In addition, computing educators need to be aware of laws pertaining to access, privacy and accessibility when providing course material on the web. These issues are extremely important for not only copyright issues, but also for personal privacy issues. How the emerging technologies affect how we teach and what we are doing with our materials is important for all educators to know.

This workshop will examine the various copyright law and ethical considerations, particularly as they pertain to the internet. Particular examples will be given using classroom scenarios. On-line documents, labs and other materials will be examined for use on the web pertaining to law, ethics and copyright issues. Participants will create various scenarios to use with their own classes. The participants will leave the workshop with a set of usable examples to use in various computer science, information technology or software engineering courses. These examples would be of possible ethical or law violations.

How to Incorporate the Social and Professional Issues Component of the ACM 2001 Computing Curriculum into a Computer Science Degree Program

John P. Buerck, Ph.D.
Computer Science Technology
Saint Louis University
Phone: (314) 977-2320
Email: *buerckjp@slu.edu*

TUTORIAL ABSTRACT

This tutorial will outline and describe two methods for incorporating the social and professional issues component of the ACM 2001 Computing Curriculum into a computer science degree program. The first method is to develop a new stand-alone course traditionally titled "Computing and Society". The second method is to develop modularized mini-lessons and activities that address the social, professional and ethical issues of computing professionals. These mini-lessons and activities can then be incorporated into existing courses required for Computer Science students.

TUTORIAL OUTLINE

I. Introduction

II. ACM Computing Curriculum 2001 – Social and Professional Issues
   a. The need to incorporate the study of social issues into the curriculum
   b. CSAB/ABET Accreditation requirements
   c. Core and elective topics
      i. History of computing
      ii. Social context of computing
      iii. Methods and tools of analysis
      iv. Professional and ethical responsibilities
      v. Risks and liabilities of computer-based systems
      vi. Intellectual property
      vii. Privacy and civil liberties
      viii. Computer crime
      ix. Economic issues in computing
      x. Philosophical frameworks

III. Ethics across the curriculum programs and centers

IV. Development of a new course titled "Computing and Society"
   a. Fund the development of the new course
   b. Obtain approval of the new course
   c. Address the issue of who is qualified to teach the courses
   d. Design course activities and assignments to meet the goals of the course
   e. Evaluation of the learning outcomes

V. Development of modularized mini-lessons and activities to be incorporated into existing courses
    a. Champion to develop or oversee development of modules
    b. Selection of courses to incorporate mini-lessons and activities
    c. Mentoring the teachers of the selected courses
    d. Evaluation of the learning outcomes

VI. Instructor resources
    a. Sample syllabus
    b. Current text books and selected readings
    c. Web resources
    d. Video resources
    e. Sample modules, cases and classroom activities

VII. Summary and Tutorial Evaluation