

10-1-2002

Student Develop Real-World Computer Information Systems

Charles C. Tappert
Pace University

Follow this and additional works at: http://digitalcommons.pace.edu/csis_tech_reports

Recommended Citation

Tappert, Charles C., "Student Develop Real-World Computer Information Systems" (2002). *CSIS Technical Reports*. Paper 5.
http://digitalcommons.pace.edu/csis_tech_reports/5

This Article is brought to you for free and open access by the Ivan G. Seidenberg School of Computer Science and Information Systems at DigitalCommons@Pace. It has been accepted for inclusion in CSIS Technical Reports by an authorized administrator of DigitalCommons@Pace. For more information, please contact rracelis@pace.edu.

T E C H N I C A L R E P O R T

Number 182, October 2002

Students Develop Real-World Computer Information Systems

Charles C. Tappert

Charles C. Tappert is Professor of Computer Science at Pace University. He holds a B.S. in Engineering Sciences from Swarthmore College, holds both an M.S. and Ph.D. in Electrical Engineering from Cornell University, and was a Fulbright Scholar. He worked at IBM for 26 years, mostly at the T.J. Watson Research Center, on speech recognition and pen computing. He has over 100 publications: journal articles, book chapters, conference papers, patents, and technical disclosures. While at IBM, he taught part-time as an adjunct at Pace, SUNY Purchase, and North Carolina State University. He taught full-time at the U.S. Military Academy at West Point for seven years before accepting his current position at Pace in 2000.

At Pace Dr. Tappert has been involved primarily in the Hudson Valley Center for Emerging Technologies and in the development of the Doctorate of Professional Studies in Computing, for which he is the Associate Program Chairperson. His research interests include pattern recognition, pen computing and voice applications, graphics, algorithms, artificial intelligence, human-computer interaction, and e-commerce.

Students Develop Real-World Computer Information Systems

Charles C. Tappert

School of Computer Science and Information Systems

Pace University

Abstract: *In the academic year 2001-2, Pace University CSIS students served the community by developing real-world computer information systems for actual customers. We describe the general use of team projects in CSIS at Pace University, the real-world projects from this academic year, the benefits of including real-world projects in the curriculum, and their cost in terms of the instructor load and the project development infrastructure. We found that the benefits of our method of structuring and conducting the real-world projects far outweigh the cost of providing them. The projects provide valuable systems for the customers, allow the students to develop technical and value skills, foster interdisciplinary collaboration, encourage student involvement in the university and local communities, support student and faculty research, enhance relationships between the university and local technology companies, and increase national recognition of the university.*

Introduction

In the academic year 2001-2, the School of Computer Science and Information Systems (CSIS) at Pace University used real-world computer information systems (CIS) as team projects in two sets of courses: the two-course capstone sequence in Software Engineering, CS615-616 [18, 19], and an elective course in Pervasive Computing Systems, CS631Q [20], in the Master of Science in Computer Science program. Although the students in both these courses develop CIS for actual customers, the emphasis of the project work in these two courses is somewhat different. In the software engineering sequence, the customers are referred to as clients. The clients are the source of the project requirements, and the teams focus on developing CIS that meet the real needs of the clients. In the pervasive computing course, student invention teams, called E-Teams, the “E” standing for “entrepreneurship,” work with customers to build innovative real-world CIS that use pervasive computing techniques and that demonstrate the feasibility of potential commercial products. In that course, the customers are referred to as mentors since they are typically the knowledge sources for the CIS. In this paper, however, the term “customer” will be used for both client and mentor in describing 18 student projects.

Developing real-world CIS for real customers in the senior capstone software engineering sequence provides an excellent setting for teaching students team and value [9] skills by following a human-centered development process. While it is clear that students of computing must learn technical skills, current thinking is that they must also learn “value skills” of how to interact with customers to provide them with the promised value-producing systems. Dertouzos makes a strong case for human-centered computing in which technology adapts to people rather than the other way around [12]. In a series of articles [7-11], Denning discusses similar concerns in describing the discipline of computing and how computing professionals should cultivate value-generating

relationships with customers in order to distinguish themselves from technicians. In one of these articles [9], Denning and Dunham describe six categories of value skills: coordination (of requests, offers, promises, etc.); customer relations; commitment management; teamwork; lifelong learning; and business and entrepreneurship skills.

Pace University's president, David A. Caputo, in his State of the University Address [4] encouraged community involvement and stressed his desire to strengthen Pace's image and identity. Our method of structuring and conducting the work on the real-world systems for real customers does this and more. The real-world projects provide valuable systems for the customers, allow the students to develop technical and value skills, utilize student-centered learning, foster interdisciplinary collaboration, encourage student involvement in the university and local communities, support student and faculty research, enhance relationships between the university and local technology companies, and increase national recognition of the university.

The following sections describe the general use of team projects in CSIS at Pace University, the real-world projects from the 2001-2 academic year, the benefits of including real-world projects in the CSIS curriculum, their cost in terms of the instructor load and the project development infrastructure, and conclusions.

Team Projects in CSIS at Pace University

The Computer Science department at Pace University has frequently used team projects to provide students with the educational experience of working together in teams, similar to what is done in industry, in order to design, build, and test computer information systems. For several years prior to 2001-2 the CS615-616 Software Engineering sequence had each student team develop as its project an "Order Processing System." In 2000-2001 the student teams used file input to develop their system prototypes during the first semester, and reengineered their systems and added a Web interface in the second. Since the inception of our Doctor of Professional Studies in Computing (D.P.S.) program [24] three years ago, students in that program have also created an order processing system for their first-year project. For these "canned" projects the instructor has played a dual role, acting as both project manager and customer. While these projects provide reasonable educational experiences, they do not give students an opportunity to interact with actual customers or to develop real-world systems that can be used by and provide value to the customer.

A project team typically consists of 4 to 5 students: an Architect-Designer, one or two Implementers, a Quality Officer, and a team Coordinator-Liaison [29]. For small teams several team member functions can be combined. At least one team member, usually the Coordinator-Liaison, should be a good communicator for customer and instructor interactions. Ideally, once the project is underway, teams should meet once a week in addition to project work time. Team members should work together and coordinate activities as a unit – for example, communication, including e-mail, between the team and the customer should be done as a group or should be coordinated by the team leader. The same students should work together on a project over the two-semester sequence; however, in rare circumstances, teams may be reconstituted over the course of the academic year due to problems with student class schedules, personality conflicts, unforeseen changes in the needs of the project, or other factors. In recent years, Java has

been the preferred language at Pace University for projects that require programming. Although the requirements for the projects come from the customers, the course instructor is the “boss,” or “Chief Executive Officer,” of the project teams – that is, the person who makes all the major decisions and adjudicates personnel problems.

For the CS615-616 sequence this past academic year, most teams delivered a prototype system that performed the basic required functions to their customer at the end of the first semester. This is usually possible since, according to the 80-20 rule, 80 percent of most computing development tasks can be completed in 20 percent of the time [29]. A complete, high-quality system was delivered at the end of the second semester.

Real-World Projects in 2001-2

Using real-world projects with real customers in computing courses is not new; for example, the University of Southern California’s Center for Software Engineering has been doing this for years in their capstone two-semester software engineering sequence [16]. We believe, however, that our program in the School of CSIS at Pace University is unique because it focuses on value skills, serves the community, and is organized on a large scale with supporting project infrastructure.

Eighteen projects were completed during the 2001-2 academic year (Table 1).

Project	Customer
Online Course Survey System	Barbara Pennipede (Assessment/Pace) Dr. Allen Stix and Chris Longo (CSIS/Pace)
Project Group Assignment System	Dr. Tappert (CSIS/Pace)
Test Item Reliability Analyzer	Dr. Stuart Varden (CSIS/Pace)
Genealogy Web Application	Dr. Tappert (CSIS/Pace)
Dean’s Communication System	Fran O’Gara and Dean Susan M. Merritt (CSIS/Pace)
PC Maintenance/Tracking System	Mike Sidaras, Bill Bernhey, Bernice Houle (CSIS/Pace)
DPS External Website	Dr. Fred Grossman, Dr. David Sachs, Tom Lombardi, and Chris Longo (CSIS/Pace)
Experimental Handwriting System	Dr. Sung-Hyuk Cha (CSIS/Pace)
Rare Coin Grading System	Rick Bassett (DPS/Pace) and Dr. Sung-Hyuk Cha (CSIS/Pace)
User Verification System	Dr. Sung-Hyuk Cha (CSIS/Pace)
Nurse Information System (NIS)	Barbara Thomas (Nursing/Pace) and Jean Coppola (DoIT/Pace)
NIS Wireless Extension	Barbara Thomas (Nursing/Pace) and Jean Coppola (DoIT/Pace)
Emergency Pre-Hospital Care Communication System	Rich Chasinoff and Dan Blum (Northern Westchester Hospital) Dr. Joseph Pastore (Business/Pace)
Medical Vital Sign Wearable	Dr. Tappert (CSIS/Pace) and Dr. Hussein Hanafi (IBM Research)
VoiceXML Application Development Facility	Darshan Desai and Shreenath Laxman (CSIS/Pace) Consulting with Speech Group (IBM Research)
VoiceXML Applications	Jean Coppola (DoIT/Pace), Shreenath Laxman (CSIS/Pace) Dr. Maryalice Citera (Psychology/SUNY New Paltz)
Multimodal Voice/InkXML System	Dr. Zouheir Trabelsi (Visiting Fulbright Scholar/Tunisia) Consulting with Pen Technologies Group (IBM Research)
Cluster/Grid Computing System	Dr. Tappert (CSIS/Pace) and Jean Coppola (DoIT/Pace)

Table 1. The eighteen team projects in 2001-2.

Ten were Web interfaces to backend databases, four were medical applications, three concerned VoiceXML or InkXML, and one was a cluster and grid computing system. The customers were from CSIS; from the Lienhard School of Nursing; from the Lubin School of Business; from the Office of Planning, Assessment, Research, and Academic Support; from the Department of Information Technology (DoIT); D.P.S. students; a visiting Fulbright scholar; and from outside organizations such as Northern Westchester Hospital, IBM, and the Psychology Department at SUNY New Paltz.

Student teams in each of these courses were required to write technical papers for possible publication that describe their systems. Eight of these papers, referenced herein, were presented at this year's Mid-Atlantic Student Workshop on Programming Languages and Systems, and PDF versions of the papers are available from the workshop's Website [15].

Web interface to backend database client/server systems

Ten of the eighteen systems developed this year by the student teams consisted of a Web interface to a backend database using the client/server architecture. The simpler ones of these simply allowed users to enter information to be stored in the database and retrieved and formatted the data for appropriate viewing. The more complex ones performed some calculations on or processing of the data to obtain useful information, and in some cases the processing was substantial.

The **Online Course Survey System** [28] was designed to automate Pace University's manual course-end survey system, in which students complete paper surveys to evaluate the courses they have taken. With the new system, students log in to take the survey through a Web interface, results are retained in a database, and survey statistics can be accessed from the Website. The opinion summaries generated by this system will not only give useful feedback to the administration and faculty but may also help the students to select their future courses. The system was tested by running in parallel both the old and the new survey systems on two large CSIS courses at the end of the spring semester. It was also used to obtain surveys for all online CSIS courses, courses for which it was previously not possible to obtain survey results, since the students in these classes do not physically attend class and therefore cannot be given a paper survey to fill out. These pilot studies showed the effectiveness of the system, and it is anticipated that the whole university will soon move to an online system for its course-end surveys. This team also developed surveys to evaluate the D.P.S. dissertation process.

The **Project Group Assignment System** [13] is a Web-based, interactive system that creates teams (groups) for project work in industry or academia. The automated system allows project managers and candidate team members to enter information through a Web interface into a database, and then allows an algorithm to create the teams. This system addresses problems that arise in the usual, manual team-creation process, which can be tedious, time consuming, perceived as unfair, and may not get a good mix of expertise. In the automated system, the project managers enter the project information: suggested team size; customer information; and parameter weights for team candidates' project preferences, availability, geography, and experience. Potential team members securely log in to the system and complete a survey form that records their project preferences, preferred availability times and locations for team meetings, and their academic and/or professional experience. The project manager can then execute a

Java program that groups the candidates into teams and can reassign team members where necessary to handle canceled projects or team member dropouts. Team members and program managers can view the resulting teams through the Web interface. Since an exhaustive search to obtain the optimal solution is not possible for more than a small number of projects, the algorithm uses a greedy search approach to obtain a near-optimal solution similar to a previous stand-alone program [30]. The system was successfully tested in the software engineering course by comparing the results of the automated system to the original manual grouping of 45 students into 12 teams, and we anticipate using the system to group students into project teams in future offerings of the software engineering course.

The **Test Item Reliability Analyzer** allows instructors to perform statistical analysis on multiple-answer tests. The measures provided by the system give the instructor greater insight into the relative preparedness of the students as well as the validity of individual test questions. The system computes five item (test question) measures: item difficulty (the percentage of students answering the item correctly), item variance (a measure of the variability of the student responses), item discrimination (a measure of the discrimination power of a test item), item correlation (a measure of the correlation between the item and the whole test), and item distraction (incorrect alternatives for a test item that are never chosen and should therefore be removed or replaced). It also computes an aggregate analysis indicator of overall test quality and accuracy. Such objective measures of the questions are valuable, since the things that separate good items from bad items are best not left to the subjective interpretation of the test constructor. A review of preexisting software programs that handled the customer's specific requirements (including flat file submission, remote access, and specifically requested analyses) showed how this system is an improvement over current work in the area. There are virtually no systems available, either publicly or privately, that offer these services to education professionals. The one similar system found, called ITEMAL [22], is proprietary to the University of Delaware, was never intended for public distribution, and is not very flexible or user-friendly. The new system successfully analyzed the quizzes used in the software engineering course and indicated where improvements can be made.

The **Genealogy Web Application** [2] is an example of developing a Web-based application that improves on an existing commercial product – the shrink-wrapped, stand-alone software application packages for the Microsoft Windows Operating System environment. With the stand-alone genealogy systems, only one family member can research, input, and publish his family tree and thereby undertake the tedious and never-ending task of updating and maintaining it, a particularly burdensome task for large family trees. In contrast, the Web-based application distributes this burden by allowing several family members to update and maintain their branch of the family tree and to do so remotely from anywhere through a Web interface. Furthermore, all family members can view the family tree information on the Web. To reduce the possibility of several family members' changing the data, write and change permission is granted to certain family members and to specified portions of the family tree. For privacy, viewing access is also limited to family members by a password.

The **Dean's Communication System** assists the Dean of the School of CSIS in her communications with faculty, students, and others at Pace University. It consists of a

database backend with both an external and an internal Web interface. The external “Communications from the Dean” site is accessible by all, and the internal one is accessible only by those having access permission. All communication items are stored in the database for tracking and for historical record keeping.

The **PC Maintenance/Tracking System** assists in the maintenance, support, and tracking of PCs in CSIS at Pace University. Information is captured when a PC is reported as having a problem or needing an upgrade (information such as the date, the PC ID, the PC’s location, the nature of problem or upgrade), when the problem is fixed (what repairs were made), when software is installed (the software, the installer), when a PC is moved, etc.

The **DPS External Website** team developed a new Website for the Doctor of Professional Studies in Computing (D.P.S.) program [24]. This site improves on the old site in terms of elegance, clarity, and ease of navigation. It has a backend database and a link to a back office interface that allows administrators without knowledge of HTML to change or update the look and content of the Website. The site also allows visitors to request information and to submit an application online.

An **Experimental Handwriting System** was developed to investigate the detection of handwriting forgery. The team collected handwriting samples from ten subjects who wrote a set of words in their natural style and attempted forgeries of the other nine subjects’ handwriting. These handwriting samples were digitally scanned and stored in an image database. A Web-based quiz was developed to gather statistical data from users trying to distinguish between the original and the forged handwriting samples. Novices and certified document examiners (CDEs) were asked to take the online quiz. Although many students took the quiz as novices, only a few CDEs were willing to do so, and the results were inconclusive. The implications of this and similar studies may have bearing on the admissibility of handwriting testimony in U.S. courts. The database of handwriting samples was also used in a study of handwriting forgery detection by machine [5].

The **Rare Coin Grading System** [3] was designed to automate the rare-coin-grading process. In the numismatic community today, the official grading of coins is accomplished by human inspection. This, by its very nature, lends itself to inconsistent results that can misrepresent the actual value of a coin by thousands of dollars; there is also the danger of misrepresentation for personal gain. Therefore, an automated computer grading system should be a welcome unbiased alternative. To demonstrate the potential utility of such a system, the team began to develop a system that grades, appraises, and authenticates rare coins by comparing them to preappraised coins stored in a database. First, a Web interface was structured as an educational quiz environment in which experts and novices can try their hand at grading coins. Second, the team developed a rudimentary machine grader by extracting three histograms (hue, saturation, and brightness) and an edge detection function from the digitally scanned color image of the coin to be graded, and assigning that coin the grade of the pregraded coin that it most closely matched. Preliminary results of the machine process appear reasonable and indicate the promise of the approach. Although these four coin features alone do not suffice to optimally determine a final grade, they are an important step toward a potentially accurate and valuable system. The team also made good use of object-oriented class methods to construct, manipulate, and display the histograms.

The **User Verification System** [1] was developed to investigate combining several biometrics in order to develop a high-confidence user verification system. A user verification system can be embedded into any system that requires high security and can be developed using various combinations of different modalities; this project used four: face, fingerprint, handwriting, and voice. Each of the four verification systems involves techniques of pattern recognition to compare signal samples from the input device with those from the database, and each can result in type I (valid user rejected) and type II (invalid user accepted) errors. Combining these four different verification techniques should achieve a high level of confidence. Initial work on this problem focused primarily on face recognition and resulted in a database of digitized photos and preliminary studies of verification by face image.

Medical application systems

Four systems involved medical applications. The **Nurse Information System (NIS)** [25, 26] physical assessment application walks a student nurse through a physical assessment. A legacy system on a proprietary device running an obsolete operating system was ported to a Java 2 Micro Edition (J2ME) implementation. The application was completely redesigned and rewritten using object-oriented design techniques to create portable classes for reuse and for use in future applications. A small prototype was built and scaled up to a full application on Palm personal digital assistants (PDAs) running the PalmOS. A C++ conduit was also written to allow the prototype to transfer data from the Palm handheld devices to a PC for storage and further evaluation.

The same students who developed the above-described NIS in the software engineering course extended that system with an **NIS Wireless Extension** in the pervasive computing course. They provided the earlier system with a wireless extension through the use of Java servlets on a Web server to allow the data to be e-mailed as an attachment to a user-specified e-mail address [25, 26]. This ensures that the NIS has a cross-platform future and can migrate from one device to another to take full advantage of the J2ME networking features. This project team also investigated the viability of commercializing the system.

A preliminary study of an **Emergency Pre-Hospital Care Communication System** was conducted for Northern Westchester Hospital [27]. Many hospital emergency departments are exploring new communication and information technologies that will assist them in providing higher quality of care by improving the speed of flow, the consistency, and the accuracy of information shared among all the parties involved in an emergency response team. This study describes some currently available or on-the-horizon communication and information technologies that may be appropriate for use in the emergency services field, discusses three communication systems currently being used by other hospitals, and provides three alternative approaches to obtaining the type of emergency communication system that will best fit the needs of Northern Westchester Hospital's emergency department. The study found that the particular solution to be chosen should ultimately depend on the time frame allotted for the project, the funds budgeted, and the number and skills of the staff members responsible for implementing the system.

The **Medical Vital Sign Wearable** computer is a prototype electronic system that monitors a patient's vital signs continuously using new non-invasive methods. The vital

signs analyzed in this study were the heart rate and breath rate. If such analysis of a patient's immediate general health were monitored remotely, it could assist medical support personnel. For example, each patient could be equipped with a medical status monitor that constantly reads blood pressure, pulse, respiration, and blood oxygen level, and a built-in expert system could determine when the patient's condition warrants attention and, if necessary, automatically transmit an alarm. This work extends that of a West Point student project in which a liquid-filled sensor pad placed close to the skin captured a signal that was processed by digital filtering to extract the heart and breath sounds and by performing Fourier analysis to separate heart and breathing signals for display or subsequent heart and breath rate determination [32]. After the semester ended, we became aware of a new, similar product, the LifeShirt, which is a lightweight shirt with embedded sensors that measure respiratory function, heart rate, posture, and activity level, with optional peripheral devices to measure blood pressure and blood oxygen saturation [23].

VoiceXML and InkXML systems

Three projects involved VoiceXML and/or InkXML. The **VoiceXML Application Development Facility** extended earlier work [31] to create a VoiceXML lab facility. Tools were created to facilitate VoiceXML application development, and several small applications were used to test the system. The lab will provide a gateway to develop voice applications, the interface for developing applications together with a template library, the capability to enable novices to use the code library to learn about and to build advanced applications, and will facilitate both novice and expert developers in deploying multiplatform applications. The laboratory hardware consists of a Cisco router for Voice Over IP, the IBM Voice Server (provided by IBM), a Local Area Network (LAN), the Public Switched Telephone Network (PSTN), and a Firewall for the Voice Server. A minimal prototype of the facility is now operational, and upon completion of the facility, the architecture of the VoiceXML studio should permit the registration, development, and deployment of an application. The studio essentially enables the user to load his/her application URL that can be referenced by the IBM Voice Server.

A second project team worked on **VoiceXML Applications**. The primary one, a voice-enabled Web-based absentee system [14], was implemented on the TellMe voice portal [21]. It was tested by the class of software engineering students; students who intended to miss a class called the VoiceXML telephone number and were led through an automated dialog to record in a database his or her name, the date and time of the call, the courseID, and the date he or she would miss class. Although it is preferred that students make the absentee call before class so that the instructor is informed prior to class time, they can also place the call after the class was missed to help the instructor maintain an accurate record of absences. The system provides the instructor and other administrators with a permanent record of absences that can be accessed and displayed in various forms through a Web interface. A more useful application of such a system would be for the university, or for any reasonably sized organization, to record absences or work time missed by its employees. Rather than calling a secretary, employees would call the system to report their absences, and the employer could then access this information through the Web interface; other departments (or systems) needing the information, such as payroll, could also access the database. In fact, we are targeting this application for

installation on the VoiceXML equipment donated this spring to Pace University by Ascent Computing Group, Inc. [17].

This team also analyzed an application that involved taking surveys by phone, which was requested by a psychology professor at SUNY New Paltz. This application is different from the usual VoiceXML applications because it requires the server to initiate the calls rather than receive them. In this case, we would like the server to place telephone calls to either a list of phone numbers or to a specified number of random phone numbers in a given location (telephone exchange). Then, when the person receiving the call answers, the application dialogue conducts the survey and stores the responses in a database. Although this application was not implemented because voice portals like TellMe only allow application developers to design applications that receive calls, it should be possible to develop such applications on the university's new Ascent hardware system.

A third XML system, the **Multimodal Voice/InkXML System**, combined voice and pen input using the standard voice and ink XML formats [33, 34]. Just as a standard VoiceXML format that facilitates the development of voice applications has been accepted, so the acceptance of a standard InkXML format to facilitate pen applications is anticipated once it is developed. The student team developed a multimodal interface architecture that combines standardized voice and ink formats to facilitate the creation of robust and efficient multimodal systems, particularly for noisy mobile environments. By providing mutual disambiguation of input signals and superior error handling, this architecture should broaden the spectrum of users to the general population, including permanently and temporarily disabled users. Integration of VoiceXML and InkXML provides a standard data format to facilitate Web-based development and content delivery. Diverse applications ranging from complex data entry and text editing applications to Web transactions can be implemented on this system, and a prototype platform and sample dialogues were developed.

Cluster and grid computing system

One team proposed its own project, the **Cluster/Grid Computing System**, and developed a high performance computing facility by co-opting idle and/or underutilized computer workstations to create a virtual computing pool [6]. While the idea is not new, it is gaining greater acceptance in academic institutions that possess enough raw computing power to do distributive computing without the need to buy a high-end dedicated computer. During the first semester, the team constructed a cluster computer consisting of five PCs and demonstrated a substantial speed improvement of the cluster over a single PC performing the calculations of a standard ray-tracing problem in computer graphics. For the second semester, the team developed a grid computer for molecular calculations by choosing a computation-intensive molecular simulation package and running tests in the student computer-lab facility that is highly used by students during peak hours. Here, depending on the nature of the simulation with respect to the number of atomic interactions that need to be calculated in real time, it was demonstrated that the more compute nodes attached to the simulation the less intrusive the job was to the user of any workstation in the grid.

Benefits of Real-World Projects

In this section we describe the various benefits of having the computer science students develop real-world systems for real customers. Overall, these projects result in a beneficial outcome for all concerned.

Customers receive valuable systems

The customers benefit from the systems created for them by the students. In some cases, this amounts to a monetary savings, because the customers do not have to outsource the projects. The university itself can greatly benefit by having small projects completed by students. Not only is there a monetary benefit, but also students often do a better job because they are more motivated and may have a better understanding of the university's needs.

In other cases, the customers receive systems they might not obtain under ordinary circumstances. Sometimes this is simply because funds for the project were not in the budget. For other situations, such as for the research-oriented systems, there is often no standard procedure for getting the system constructed.

Students develop technology, team, and professional skills

Developing real-world systems for real customers is a stellar real-world learning experience for the students, as individual technologists, as team members, and as maturing professionals in computing. Individually, the students learn technology skills, critical thinking skills, and how to be innovative and creative. They learn the technology skills necessary to design, build, and test real-world computer information systems. Through project reviews and team presentations, the students also learn about the various technologies used in other projects, and they especially appreciated the exposure to the projects involving cutting-edge technology and research.

Working in teams, the students learn fair-mindedness, intellectual humility, intellectual integrity, and the ability to work with others to produce useful systems and to take responsibility for them. Because most of the students are employed full time in various areas of computing, they bring their knowledge and expertise to bear in their project work, and by exchanging information, they learn from each other in this student-centered learning environment. The students also benefit from interacting closely with fellow students from diverse cultures and with diverse technological skills.

As maturing professionals, the students learn how to act in the computing field not only as technologists but also as value providers. By working with real customers in developing their project systems and focusing on human-centered computing, the students learn important value skills. This learning paradigm fosters lifelong habits for learning and the application of critical thinking and value skills.

Finally, these real-world experiences better prepare the students for the job market. Companies desire and are more willing to hire computing professionals having value skills in addition to having technology skills. Several students have secured jobs that were clearly based on these experiences, some due to the particular project-related areas of technology and some due to the acquired team and customer-related "soft" skills.

Promotes interdisciplinary collaboration and Pace and local community involvement

These projects involved two communities, the university community and the local community external to the university. Within the university community, many of the projects involved interdisciplinary collaboration with other departments, such as the medical projects with the nursing school, the project with the assessment department, and the involvement with the business school and DoIT. The work with the SUNY New Paltz psychology professor extended our collaboration to the greater university community.

The projects also extended into the local community, with the projects involving Northern Westchester Hospital and IBM, to provide the students with off-campus experiences and to foster an extended community for learning and growth. With project customers coming from the local community as well as from the university community, the project work brought to light how knowledge of computing and information technology connects with personal and social responsibility. Interestingly, we found that having projects that dealt with both the Pace community and the local community led to a blurring of the boundary between those communities.

Furthers student and faculty research

Seven of the systems developed by the students are related to student and faculty computer science research that is ongoing and expected to continue. Three of these concern pattern recognition: the Experimental Handwriting System, the User Verification System, and the Rare Coin Grading System; and the first two of these are related to security and personal identification. Three concern XML: the VoiceXML Application Development Facility, the VoiceXML Applications, and the Multimodal Voice/InkXML System. Finally, one concerns wearable computing devices: the Medical Vital Sign Wearable computer.

Seven other systems are research-related but are not related to ongoing student-faculty research. Two concern wireless technology: the NIS Wireless Extension and the Emergency Pre-Hospital Care Communication System. Two concern education-related research and were of short-term interest to the faculty customers: the Project Group Assignment System and the Test Item Reliability Analyzer. The two nurse information systems concern handheld computers and the J2ME, J2SE, and J2EE computer languages and, although not representing ongoing research by CSIS students and faculty, are of continued interest to the nursing school as a developing application that, due to its cutting-edge nature, should to be of interest to next year's students as a possible continuation project. Finally, one deals with supercomputing (short-term student interest): the Cluster/Grid Computing System.

Therefore, it is clear that these real-world projects for real customers profoundly foster and further student and faculty research.

Enhances relationships between the university and local technology companies

Some projects drew on the intellectual capital pool of local companies. For example, three projects benefited from collaboration with IBM scientists: the VoiceXML facility team received a copy of the IBM Voice Server from the IBM Speech group, the vital-sign wearable computer team received advice and use of an electronics lab from an IBM scientist, and the Voice/InkXML team interacted with the IBM Pen Technologies group.

Furthermore, the IBM Pen Technologies group is interested in an extended collaborative relationship to pursue research in areas of mutual interest and to have students test their handwriting research systems.

Increases national recognition of the university

Many of the benefits indicated above strengthen the university's external image and identity. Students can also present papers on their project work at technical conferences, and those working on cutting-edge technologies can enter competitions. This is already happening – for example, students and faculty have presented papers related to these systems at conferences (see references), and several students won prizes in IBM's VoiceXML contest. Thus, the student development of real-world systems is an example of what Pace University's president, David A. Caputo, refers to as "programs that will make Pace more synonymous with innovation, value and academic strength" [4].

Cost of Real-World Projects

Although far outweighed by the benefits, the cost of using real-world projects in academic courses is significant. There is an increased load on the instructors to solicit and interact with customers, to set up the projects, to work with the computer support personnel to set up the project development infrastructure, and to monitor the process of developing the systems. There is also significant computing infrastructure required to facilitate the development work of the students' systems.

Setting up projects, forming teams, and assigning projects

The first task is to come up with the projects, and this process is conducted primarily during the summer prior to the beginning of the school year. For this year, eight projects came from those interested in developing systems to further their research (CSIS faculty, sometimes in collaboration with local technology companies, and, in one case, with a D.P.S. student), one came from the instructor's course need (Project Group Assignment System), one came from the students' interest (Cluster/Grid Computing System), and the remaining eight came from contacts with and e-mail solicitations to faculty of CSIS and other departments.

The instructor then works with each of the customers to size and shape the project to be an appropriate systems development experience for the students, and this results in a written project description that is posted on the "projects" page of the course Website. For customers not familiar with computing systems, the instructor often performs most of this task independently. The inducement for the customers' involvement with the students throughout the year is the anticipation of receiving a useful system, although they are warned that the primary purpose of the projects is to provide the students with a good educational experience and that not all projects are successfully completed.

The next task is to form the student teams and assign each team to a project. This year the students formed their own teams, the teams listed their project preferences, and the instructor assigned teams to projects based almost entirely on their preferences. There are significant problems with this approach, such as the tendency of students to choose group members based solely on friendship and that the assignment of projects can be perceived as unfair by students who did not obtain their first choice. For this reason,

one of the projects this year was to develop a group assignment system to automatically assign students to project teams based on compatible project interests, compatible time schedules and geography (work and home location) for group meetings, and the diversity and appropriateness of students' computing and leadership experience; this system will be used next year to automatically group students into teams and assign the teams projects. Another method to augment this process, which has been used successfully at West Point, is to conduct a "customer fair" in which students and customers meet and interact, typically during an hour's class time in an auditorium or large room where each customer is set up at a table and the student teams can circulate and discuss the projects with the customers.¹ This method also allows the customers to list their preferences of the student teams that they would like to work with, and this information could serve as additional input to an automated system.

Systems development process

The development of the computer information systems requires a systematic approach by the students and proceeds through an evolutionary process model consisting of analysis, design, build, and test phases that will be described only briefly here. The first phase requires the students to work closely with their customer to understand what is desired, usually by developing use cases and use scenarios about how the system will be used; to plan the project; and to perform a risk analysis, all culminating in a written requirements document. The second phase involves designing the system using various analysis and design tools, and this work varies greatly depending on the type of system. For example, a database backend requires database design methodologies, whereas a Web interface requires interface and Web design engineering. The third phase is the actual construction of the system and, depending on the project, usually requires the use of one or more computer software languages and/or database implementation. The fourth phase is the testing of the system to ensure that it meets the customer's requirements and that all of its functions operate correctly.

In the software engineering sequence, the students were required to complete a working system prototype with the core components of the required functionality by the end of the first semester. Because the first semester of the course focuses on the traditional systems development methodologies and the second semester on the object-oriented ones, the students reengineered their systems using object-oriented techniques during the second semester to culminate in a robust and complete system. For the pervasive computing course, the systems were completed in one semester, which was possible because of the course's greater emphasis on project work and because most of the students were also in the software engineering course and therefore were familiar with systems development methodologies.

Project development infrastructure

Since most of the projects used the client/server architecture, a development server called Utopia, a Pentium II 266 MHz NT machine, was set up for the students to develop their systems. The software installed included Cold Fusion, PHP (Hypertext Preprocessor), Active Server Pages (ASP), MySQL, Tomcat, and Microsoft Office. This server was

¹ Author's personal experience teaching a similar two-semester capstone course at the United States Military Academy at West Point.

independent and separate from the other CSIS servers so that students could not corrupt data or interfere with operations on CSIS production servers. A directory was set up on the server for each project, and students were told to restrict their access to their project's directory. The variety of database-related software used by these client/server systems is summarized below in Table 2.

Project	Scripting software	Database software
Online Survey System	ASP	Microsoft Access
Project Group Assignment System	Cold Fusion	Microsoft Access
Test Item Reliability Analyzer	PHP	MySQL
Genealogy Web Application	Cold Fusion	MySQL
Dean's Communication System	Cold Fusion	Microsoft Access
PC Maintenance/Tracking System	Cold Fusion	Microsoft Access
DPS External Website	PHP	MySQL
Experimental Handwriting System	ASP	Microsoft Access
Rare Coin Grading System	ASP	Microsoft Access
User Verification System	ASP	Microsoft Access
VoiceXML Applications	PHP	MySQL

Table 2. Systems involving a Web interface to a backend database.

This table includes the ten client/server systems described earlier as well as the VoiceXML absentee system that also used a backend database and a Web interface, the latter for displaying the results and as an alternative to the telephone input. As indicated in the table, the student teams used three scripting languages (Cold Fusion, PHP, and ASP) to communicate with the databases and two database systems (Microsoft Access and MySQL). Several of these projects also used Java-related software, such as Java servlets, and Tomcat was installed to handle the processing of that code.

In addition to the above-mentioned client/server hardware infrastructure, the Utopia server was maintained by CSIS technical support personnel and sometimes required frequent attention. Another server was made available for the VoiceXML facility project and six old Pentium I and II PCs, network cards, and network cables were made available for the cluster computer project. Also, toward the end of the spring semester, the Online Survey System was moved to a production server to provide critical survey information to appropriate personnel from CSIS and from the Office of Planning, Assessment, Research, and Academic Support.

The development server was set up to assist the students in their systems development work because most students do not have access to such equipment or to the software installed on it, and the development environment grew over the year as software was added in response to student requests. Nevertheless, some students preferred to use their own development environment, and this was encouraged so as not to limit them by the development environment that we had set up.

Suggested improvements to the project development infrastructure

Although the development server was extremely helpful for the development of the student project systems, it was heavily utilized, especially toward the end of the spring semester, and had to be rebooted frequently during times of heavy use.

The project development infrastructure will be enhanced next year to reach a higher level of professionalism in project development. We will install the various scripting language and database software on several PCs in the computing lab so that students can test their systems before uploading them to the development server. This will reduce the frequent uploads and thus the overall load on the development server.

The Utopia development server will be upgraded. A Unix server could be added – a Solaris-Oracle configuration would be ideal, or a Linux-MySQL-PHP-Perl solution would be of minimal cost. Student access will be limited to their project's directory on the development server. A production server will be added toward the middle of the spring semester to house the final systems and to give the students practice in moving their system from one environment to another. We must also develop procedures for handling other issues relating to putting student project systems into production. One issue, for example, is system maintenance after the students complete the course; presumably, the faculty member or his/her students should maintain research-oriented systems developed for them, and university departments should similarly maintain their systems.

Conclusions

In the academic year 2001-2, Pace University CSIS students developed real-world computer information systems for actual customers as their course team projects. Developing real-world systems provides an excellent setting for teaching students the technological skills of the discipline, team-related skills, and value skills by following a human-centered development process. It also provides a vehicle for fostering interdisciplinary collaboration, encouraging student involvement in the university and local communities, furthering student and faculty research, enhancing relationships between the university and local technology companies, and increasing national recognition of the university.

Acknowledgements

The students and I would like to thank the project customers (see Table 1) for their willingness to interact with the student teams. We thank Professor Barbara Thomas (Nursing), Jean Coppola (DoIT), Professors Sung-Hyuk Cha, Bel Raggad, and David Sachs (CSIS), and Professor Joe Pastore (Business) for their excellent guest lectures in the Pervasive Computing Systems course. We thank the Hudson Valley Center for Emerging Technologies for funding for the VoiceXML equipment. We thank the CSIS technical support personnel for providing and maintaining the client/server infrastructure and for providing the other computing equipment; in particular we'd like to thank Tom Lombardi for setting up, maintaining, and frequently rebooting the Utopia server, and for helping the students with coding difficulties; Michael Sidaras for finding the server for the VoiceXML facility project and the old PCs for the cluster computer; and Matthew

Poli for overseeing this support. We also thank graduate assistant Naresh Trilok for setting up an online forum for student interaction on project questions, for providing PHP and Cold Fusion coding examples, and for helping students with coding problems. I would like to thank the students for their enthusiasm and effort in working closely with their customers to develop and deliver outstanding project systems, as well as Bernice Houle, Allen Stix, Zubin Wadia, Louise Kleinbaum, and others for reading early drafts of this paper and offering valuable suggestions. Finally, I thank Dean Susan M. Merritt for providing the opportunity and support for this endeavor.

References

- [1] W. Baker, A. Evans, L. Jordan, and S. Pethe, "User verification system," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [2] C. Barbosa, N. Pandey, O. Pavlenko, P. Cuning, and S. Pramod, "A Web-based genealogy application system," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [3] R. Bassett, P. Gallivan, X. Gao, E. Heinen, A. Sakalaspur, "Development of an automated coin grader: a progress report," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [4] D.A. Caputo, "State of the university address 2002," http://appserv.pace.edu/execute/page.cfm?doc_id=2906.
- [5] S.-H. Cha and C.C. Tappert, "Automatic detection of handwriting forgery," to be presented at the *8th Int. Workshop Frontiers Handwriting Recognition (IWFHR-8)*, Niagara, Canada, August 2002.
- [6] C. Clarke, L. Marino, and R. Pachigolla, "Creating a virtual computing facility: emulating grid services reference model," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [7] P.J. Denning, "Who are we?" *Communications of the ACM*, Vol. 44, No. 2, February 2001, pp. 15-19.
- [8] P.J. Denning, "Crossing the chasm," *Communications of the ACM*, Vol. 44, No. 4, April 2001, pp. 21-25.
- [9] P.J. Denning and R. Dunham, "The core of the third-wave professional," *Communications of the ACM*, Vol. 44, No. 11, November 2001, pp. 21-25.
- [10] P.J. Denning, "Internet time out," *Communications of the ACM*, Vol. 45, No. 3, March 2002, pp. 15-18.
- [11] P.J. Denning, "Flatlined," *Communications of the ACM*, Vol. 45, No. 6, June 2002, pp. 15-19.
- [12] M. Dertouzos, *The Unfinished Revolution*, Harper, 2001.
- [13] K. Doyle, S. Kroha, A. Palchowdhury, and W. Xu, "Project group assignment system," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [14] P. Gallivan, Q. Hong, L. Jordan, E. Li, G. Mathew, Y. Mulyani, and P. Visokey, "VoiceXML absentee system," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [15] <http://csis.pace.edu/csis/masplas/> (MASPLAS'02 Website).
- [16] <http://sunset.usc.edu/index.html> (University of Southern California Center for Software Engineering).
- [17] <http://www.ascent-inc.com/> (Ascent Computing Group, Inc.).
- [18] <http://www.csis.pace.edu/~ctappert/cs615-01/> (CS615 Website).
- [19] <http://www.csis.pace.edu/~ctappert/cs616-02/> (CS616 Website).
- [20] <http://www.csis.pace.edu/~ctappert/cs631-02/> (CS631Q Website).
- [21] <http://www.tellme.com/> (TellMe voice portal).

- [22] <http://www.udel.edu/topics/testscan/itemal.html> (ITEMAL system description).
- [23] <http://www.vivometrics.com/ce/> (VivoMetrics).
- [24] S.M. Merritt, F. Grossman, C. Tappert, J. Bergin, H. Blum, R. Frank, D. Sachs, A. Stix, and S. Varden, "The Doctor of Professional Studies in Computing: An Innovative Professional Doctoral Program," *Proc. 19th Annual Information Systems Educational Conference, ISECON* 2001, November 2001.
- [25] S. Palmer, N. Panchee, J. Sullivan, K. Thabet, and S. Westgard, "Migrating an application to Java2 Micro Edition: from port to portability," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [26] S. Palmer, N. Panchee, J. Sullivan, K. Thabet, and S. Westgard, "Porting an application to J2ME, J2SE and J2EE: Experiences with end to end Java programming," CSIS Technical Report 175, May 2002.
- [27] H.G. Park, J.M. Pastore, Jr., and C.C. Tappert, "Wireless technologies in pre-hospital communications: an analysis for Northern Westchester Hospital," CSIS Technical Report 176, May 2002.
- [28] S. Pethe, W. Baker, N. Brown, R. Hennings, S. Misra, "Online course opinion survey system," *Proc. 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, April 2002.
- [29] R.S. Pressman, *Software Engineering*, 5th Edition, McGraw-Hill, 2001.
- [30] M.A. Redmond, "A computer program to aid assignment of student project groups," *Proc. 32nd SIGCSE Technical Symposium on CS Education*, pp. 134-138, 2001.
- [31] C. Tappert, S. Laxman, and D. Desai, "VoiceXML Research Facility," Pace University Research Day, May 2001.
- [32] C.C. Tappert, A.S. Ruocco, K.A. Langdorf, F.J. Mabry, K.J. Heineman, T.A. Brick, D.M. Cross, S.V. Pellissier, and R.C. Kaste, "Military applications of wearable computers and augmented reality," Chapter 20, pp. 625-647, in *Fundamentals of Wearable Computers and Augmented Reality*, ed. W. Barfield and T. Caudell, Lawrence Erlbaum, 2001.
- [33] Z. Trabelsi, C. Tappert, and D. Desai, "Integrated VoiceXML and InkXML gateway," *Proc. AVIOS Speech Expo 2002*, San Jose, CA, May 2002.
- [34] Z. Trabelsi, S.-H. Cha, D. Desai, and C. Tappert, "A Voice and Ink XML Multimodal Architecture for Mobile e-Commerce Systems," to be presented at the 2nd ACM Int. Workshop on Mobile Commerce 2002, Atlanta, GA, September 2002.



School of Computer Science and Information Systems
Pace University
Technical Report Series

EDITORIAL BOARD

Editor:

Allen Stix, Computer Science, Pace--Westchester

Associate Editors:

Connie Knapp, Information Systems, Pace--New York

Susan M. Merritt, Dean, SCSIS--Pace

Members:

Howard S. Blum, Computer Science, Pace--New York

Donald M. Booker, Information Systems, Pace--New York

M. Judith Caouette, Office Information Systems, Pace--Westchester

Nicholas J. DeLillo, Mathematics and Computer Science, Manhattan College

Fred Grossman, Information Systems, Pace--New York

Fran Goertzel Gustavson, Information Systems, Pace--Westchester

Joseph F. Malerba, Computer Science, Pace--Westchester

John S. Mallozzi, Computer Information Sciences, Iona College

John C. Molluzzo, Information Systems, Pace--New York

Narayan S. Murthy, Computer Science, Pace--New York

Catherine Ricardo, Computer Information Sciences, Iona College

Sylvester Tuohy, Computer Science, Pace--Westchester

C. T. Zahn, Computer Science, Pace--Westchester

The School of Computer Science and Information Systems, through the Technical Report Series, provides members of the community an opportunity to disseminate the results of their research by publishing monographs, working papers, and tutorials. *Technical Reports* is a place where scholarly striving is respected.

All preprints and recent reprints are requested and accepted. New manuscripts are read by two members of the editorial board; the editor decides upon publication. Authors, please note that production is Xerographic from the pages you have submitted. Statements of policy and mission may be found in issues #29 (April 1990) and #34 (September 1990).

Please direct submissions as well as requests for single copies to:

Allen Stix
School of CS & IS - Suite 412 Graduate Center
Pace University
1 Martine Avenue
White Plains, NY 10606-1932

