DOCUMENT RESUME

ED 425 713 IR 019 201

AUTHOR	Smith, Peter, Ed.
TITLE	Association of Small Computer Users in Education: Proceedings of the ASCUE Summer Conference (31st, North Myrtle Beach, South Carolina, June 7-11, 1998).
INSTITUTION	Association of Small Computer Users in Education.
PUB DATE	1998-00-00
NOTE	222p.; For selected individual papers, see IR 019 202-224. For the 1997 proceedings, see ED 410 911.
PUB TYPE	Collected Works - Proceedings (021)
EDRS PRICE	MF01/PC09 Plus Postage.
DESCRIPTORS	Academic Libraries; Computer Mediated Communication; Computer Networks; *Computer Uses in Education; Distance Education; *Educational Technology; Futures (of Society); Higher Education; Information Networks; *Information Technology; Internet; Library Services; Material Development; Microcomputers; *Multimedia Instruction; Multimedia Materials; Navigation (Information Systems); Professional Associations; Technological Advancement; Telecommunications; Workshops; *World Wide Web
IDENTIFIERS	*Academic Computing; Association of Small Computer Users in Education; Connectivity; Web Sites

ABSTRACT

Topics addressed by the papers including in this proceedings include: multimedia in the classroom; World Wide Web site development; the evolution of academic library services; a Web-based literature course; development of a real-time intelligent network environment; serving grades over the Internet; e-mail over a Web browser; using technology to communicate, cooperate, and collaborate; information technology initiatives in higher education; using intelligent agents to assist educators; computer laboratory exercises; a student information technology support system; coordination of an academic computing help desk; Web-based parallel programming workshop for undergraduate study; a Web-based college information navigator; total quality management (TQM) in a computer lab; course preparation for distance education delivery; interactive course development; undergraduate data mining on the Web; multimedia training; computing trends in small liberal arts colleges; student-developed Web pages; automation of teacher-side Web page navigation for classroom presentations; accommodating the disabled; computer training for teachers; resource requirements in a client/server environment; Web-based campus directory; open platform software system for Internet interactive education; and teaching as instructional support. The volume also provides information about the Association of Small Computer Users in Education (ASCUE), a list of ASCUE board members, descriptions of pre-conference workshops, and a presenters index. Most papers contain references. (AEF)

ASCUE

ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION

"Continuing Second Quarter Century of Service"

Proceedings of the 1998 ASCUE Summer Conference

31st Annual Conference
June 7 - 11, 1998

North Myrtle Beach, South Carolina

Edited by Peter Smith, Saint Mary's College

ASCUE

ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION

"Continuing Second Quarter Century of Service"

Proceedings of the 1998 ASCUE Summer Conference

31st Annual Conference

June 7 - 11, 1998

North Myrtle Beach, South Carolina

Edited by Peter Smith, Saint Mary's College

Association of Small Computer Users in Education
"Continuing Second Quarter Century of Service"

Proceedings of the 1998 ASCUE Summer Conference
31st Annual conference
June 7-11, 1998
Myrtle Beach, South Carolina

ABOUT ASCUE

ASCUE, the Association of Small Computer Users in Education, is a group of people interested in small college computing issues. It is a blend of people from all over the country who use computers in their teaching, academic support, and administrative support functions. Begun in 1968 as CUETUG, The College and University Eleven Thirty User's Group, with an initial membership requirement of sharing at least one piece of software each year with other members, ASCUE has a strong tradition of bringing its members together to pool their resources to help each other. It no longer requires its members to share homegrown softwatre, nor does it have ties to a particular hardware platform. However, ASCUE continues the tradition of sharing through its national conference held every year in June, its conference proceedings, and its newsletter. ASCUE proudly affirms this tradition in its motto "Continuing Second Quarter Century of Service."

ASCUE's ASCUE-L LISTSERVER

Subscribe by sending the E-mail message SUBSCRIBE ASCUE-L yourname to listserv@gettysburg.edu. The listserv itself is ascue-l@gettysburg.edu. In order to stop the flooding of the listserv with unwanted solicitations, we have set it up so that you will have to send messages from the address you subscribed from.

NEED MORE INFORMATION?

Direct questions about the contents of the 1998 Conference to Bill Wilson, Program Chair, ASCUE '98, Director of Academic Computing, Gettysburg College, Gettysburg, PA 17325, 717-337-6933, wilson@gettysburg.edu. Web: http://www.gettysburg.edu/ascue

Association of Small Computer Users in Education

ASCUE 1997-1998 Board Members

(years remaining in office including current year)

PRESIDENT
Christine Schwartz (1 year)
Director of Computer Services
Ursuline College
Cleveland, OH 44124
216-449-4471
cschwart@main.ursuline.edu

PAST PRESIDENT
Carl Singer (1 year)
Computer Center
Depauw University
Greencastle, IN 46135
317-658-4294
singer@depauw.edu

PROGRAM CHAIR
Bill Wilson (1 year)
Academic Computing
Gettysburg College
Gettysburg, PA 17325
717-337-6933
wilson@gettysburg.edu

TREASURER
Thomas Pollack (1 year)
School of Business Administration
Duquesne University
706 Rockwell Hall
Pittsburgh, PA 15282
412-396-1639
pollack@duq2.cc.duq.edu

SECRETARY
Noreen Elliott (2 year)
Equifax Marketing Services
P.O. Box 1485
Cleveland, GA 30528
770-740-4708
nelliott@stc.net

NEWSLETTER EDITOR
Peter Smith (1 year)
Saint Mary's College
Notre Dame, IN 46556
219-284-4493
psmith@saintmarys.edu

LIBRARY/HISTORIAN
Jack Cundiff (1 year)
Horry-Georgetown Technical College
Box 1966
Conway, SC 29526
803-347-3186
cundiffj@.hor.tec.sc.edu

BOARD MEMBER AT LARGE
Richard Stewart (1 year)
Director of distance Learning
Lutheran Theological Seminary
7301 Germantown Avenue
Philadelphia, PA 19119
215-248-4616
richard.stewart.parti@ecunet.org

BOARD MEMBER AT LARGE
Rick Rodger (2 years)
Horry-Georgetown Technical College
Box 1966
Conway, SC 29526
803-347-3186
rodger@hor.tec.sc.us

Table of Contents

Keynote Speakers

Karen Leach has worked in business and education, performing efficiency studies, developing systems, and managing financial and other resources. She has been at Colgate University for thirteen years, first as Associate Provost, and as Chief Information Officer since 1993.

David Smallen has been the Director of Information Technology Services at Hamilton College for over twenty years and also directed its Office of Institutional Research. He has studied the IT environment extensively, especially for small colleges, written on the topic frequently, and provided consulting services.

The two speakers above will team up to give the first keynote address on Monday morning. Their topic is "IT's not Wizardry: The Collaboration of Dorothy and the Tin Man." They describe the talk as follows: Our campus networks are evolving from magical highways to critical intranets central to supporting our academic missions. As we move into the "yellow brick" world of 24 x 7 expectations, effective collaboration becomes essential. What are the critical success factors that make interinstitutional collaboration possible? In what ways can we forge partnerships with our colleagues to learn from and support each other in this challenging environment? How can we use the network to promote understanding and develop a common language about the practice and finance of IT? Karen and David will lead us down collaboration road stopping along the way to discuss the COSTS project, an international collaborative effort to identify exemplary practices and financial benchmarks for IT services in the educational environment.

Scott E. Siddall is a graduate of Case Western Reserve University in Cleveland, with a doctorate in oceanography from the School of Marine Science at the University of Miami (Florida). In addition to his duties as Director of Academic Computing at Kenyon College, Scott teaches the senior capstone course in Kenyon's Environmental Studies Concentration. Prior to coming to Kenyon in 1989, Scott was a member of the graduate faculty at the State University of New York at Stony Brook, and a member of the research faculty at the University of Miami. His publications include 18 papers in leading scientific journals, five book chapters, and several dozen invited presentations in both marine science and information technology. At Kenyon, he implemented a peer-reviewed incentives program for faculty, a successful helpdesk for support, a joint program with the library for instruction in information resources, a campus computer store and a campus-wide advisory council.

Scott will give the second keynote address on Tuesday morning. His topic is ""Partnering to Lower the Bar." He describes his talk as follows: We are surrounded by empowering technologies for information access and communication, many of which can redefine the teacher-learner relationships that we have cherished for decades. In this presentation, I will attempt to identify several important forces - social, technological, economic - which press us all to change our current practices, either as faculty, students or as technologists. I will discuss key issues from the perspective of each of these groups. I will examine the possibilities of broad partnerships as means for lowering the investment of time and effort required to effect genuine transformation of learning with technology. Several case studies will be reviewed in detail to illustrate the costs and benefits of varying styles of teaching and learning. I will encourage the audience to think less about "how" we use technology and more about "why" we look at technology to enhance learning.

Pre-conference Workshops

Pre-conference Workshop 1
Finding and Evaluating Information on the Internet
Presented by: Dennis Trinkle, History, DePauw University, Greencastle, IN.

This session will begin with a general discussion of the major search engines for finding information on the Internet--how each operates, what they contain, and the techniques for using them efficiently. This will be followed with methods to critically evaluate materials on the Internet, comparing the criteria and methods of assessing print sources with those of on-line materials. Other topics included: new intermediary sites that have taken as their charge pre-evaluating and organizing material for convenient scholarly access, strategies for teaching students how to critically evaluate materials on the Internet, individual and group exercises for teaching students to evaluate on-line materials. Hands-on experience with these topics will be integrated into the workshop.

Pre-Conference Workshop 4
A Little Java: An Introduction to Java and Javascript Web Applications
Presented by: Dennis Trinkle, DePauw University, Greencastle, IN.

This workshop will provide a basic introduction to Java and Javascript. It will not turn you into a commercial Java programer, but it will explore the similarities and differences between Java, Javascript and other programming and scripting languages such as C and Perl. The workshop will focus more specifically on how to integrate Javascript and Java applets into Web pages, and we will walk through the creation and integration of some simple JavaScript by way of example. Finally, we will discuss some of the excellent on-line tutorials and Java applet/Javascript libraries that provide excellent resources for those who want to build on the lessons of the workshop.

About the Presenter: Dennis A. Trinkle is an instructor in the history department at DePauw University. He is the Executive Director of the American Association for History and Computing and the co-author of The History Highway: A Guide to Internet Resources (M.E. Sharpe, 1997). He is also the editor of Writing, Teaching, and Researching History in the Electronic Age (M.E. Sharpe, 1998), and he has written articles on the practice of history in the electronic age for the Organization of American Historians Newsletter, the Organization of American Historians Council of Chairs Newsletter, University Currents, Slant, and the History Computer Review.

Pre-conference Workshop 2
Create Your Own Home Page With No Previous Experience
Presented by: Steve Anderson, University of South Carolina Sumpter.

Have you wanted to create and maintain your own home page? Has the learning curve of HTML has seemed too formidable to try to climb? Then this seminar is for you! All you need is the desire to learn an authoring software package which is free to educators. You will leave knowing that home page construction is not fraught with as much peril as you might have thought, but best of all, you will leave with a home page of which you can be proud and which was constructed using little more than basic word processing skills.

Major topics include:

* some history of HTML and how it looks in Notepad
* using an authoring environment (Netscape Communicator's "Composer" with Netscape for viewing) to construct WWW pages which will include simple text, links to existing sites (like your own school's site), bullet and numbered lists, formatted text via tables
* digital imaging including use of a digital camera which will be provided
* inserting a link to your e-mail account so others can "click you" a message
* locating useful materials on the WWW to use on your home page (see workshop 1 above)
* how to upload files you bring home to your server
* good and bad page design, when and where to use glitz
* using software like Word, Astound, Powerpoint to create image maps and forms

About the Presenter: Steve Anderson has been heavily involved in multimedia since 1992, after attending an ASCUE workshop. He has presented many papers at ASCUE and HENA which emphasize multimedia as a tool for teaching/learning as well as for presentation purposes.

Pre-Conference Workshop 3: ***Using Web Course in a Box to Get Faculty on the Web***
Presented by: Nancy Thibeault, Janet Hurn and Eric DeBrosse, Miami University, Middletown, OH.

At the Middletown Campus of Miami University we have had great success developing course materials for the Web using Web Course in A Box (WCB). WCB was developed at Virginia Commonwealth University by Bob Godwin-Jones and Sue Polyson and is free to people in higher education. WCB software is an integrated approach to enable instructors with minimal technical expertise to create and manage Web pages for a course, including the ability to:

* create course and faculty/student home pages without knowing HTML
* create threaded discussion forums with file attachments and archiving
* upload content from instructor's PC/Mac and manage links to all pages
* manage student access to pages; batch upload of student registration data
* create interactive quizzes and tests
* construct Lesson Pages containing elements such as graphics, text, headers, quizzes, discussion forums and linked files or Web sites.
* create on-line syllabus, class schedule, and announcements

All of this can be accomplished by simply completing a series of fill in forms. With WCB a faculty member creates a sophisticated and editable web course without learning HTML or the technical difficulties of publishing on the Internet. Miami University faculty and students have found WCB to be quite user friendly. We have over 20 faculty using WCB in art, physics, EDL, systems, math, BTE, anthropology, ENT, English, and chemistry courses and in the nursing resource center with great success. We have over 300 students using WCB. In this workshop, participants will set up their own WCB courses. We will demonstrate the software and its ease of use, show examples of its use, and provide hands on instruction.. We will bring our own server and provide technical advice on server set up.

About the Presenters:

Janet Hurn is is a Physics Instructor and Co-Coordinator of the Faculty Summer Institute on

Teaching with Technology at Miami University. She has also been co-cordinator of the annual technology fair and chair of the Retention Committee at the college. Her special area of interest is multimedia in Physics education.

Nancy Thibault is the Computer Services Manager and the other Co-Coordinator of the Faculty Summer Institute on Teaching with Technology and annual technology fair at Miami University. Her special area of interest is Web/Database programming.

Eric DeBrosse is a Senior Technical Services Specialist and UNIX system administrator at Miami University. He is also Webmaster and Web course in a Box administrator at the college. His special areas of interest are multimedia development and composing music.

Pre-ConferenceWorkshop 5
Managing in Difficult Times
Presented by: Carl Singer and Carol Smith, DePauw University, Greencastle, IN.

Developing effective support for information technology is a challenging and dynamic task. On one hand, we are constantly in demand by clients whose expectations change nearly every day because of influences outside of our control: advertising and commercialism, the Internet, colleagues from other campuses, and so on. At the same time, we are continually running to evolve our own professional and technical skills trying to stay one step ahead of our clients and to also provide leadership. Managing these demands while juggling institutional priorities is no small task, especially when working with a fixed set of staff, skills, equipment and budget.

Survival in this environment can be facilitated by developing the ability to be comfortable accepting ideas and approaches that may be counter to our traditional view of organization and management, the way we want it to be. At DePauw University, we have found that creating partnerships and teams in and outside of the information technology department has many benefits which aid in meeting demand and help us in information technology to provide critical leadership. However this approach also requires one to be open to new ideas and to be able to give up a certain amount of control. In this workshop, through guided activities and demonstrations, participants will acquire tools for developing strategies for managing information technology support within their own organizations through the use of partnerships and teams. The workshop is primarily for people responsible for information technology support, but the concepts and methods presented can be easily adapted to any type of organization, so other interested bystanders are encouraged to attend.

About the Presenters :

Carol Smith has been Coordinator of User Services and Computing Labs at DePauw University for the last four years. She spent six years at Indiana University in user support before coming to DePauw, where she is currently responsible for creating support mechanisms for all students, faculty and staff on campus. She is also part of a team which is developing a new faculty instructional support program at DePauw. Carol has presented papers and given tutorials on team-based methods for project management at two previous ASCUE Conferences.

Carl Singer has been Director of Academic Computing and Professor and Chair of Computer Science for many years. Along with regular duties, Carl has been actively involved in promoting effective and appropriate use of technology in teaching and learning. For the past several years Carl has successfully promoted the use of teams for problem solving and process inprovement. He has given workshops and consulted on team-based methods for project management at ASCUE Conferences and elsewhere.

Multimedia in the Classroom—A Pedagogical Change?

Stephen T. Anderson Sr.
Associate Professor
University of South Carolina Sumter
200 Miller Rd.
Sumpter, SC 29150
(803) 775-6341
stevea@sc.edu

Introduction

Over the last 12 years, I have used computer-based materials to change the way in which I taught material in math classes, quantitative methods classes, computer applications classes and computer programming classes. My efforts have evolved from acetate slides to multimedia on-line presentations. Some classes lent themselves more readily to the changes than others, but all experienced come change in the teaching/learning process. This paper will discuss the process and its implications.

The Early Days

My first attempt to supplement my lecture approach to teaching with computer based materials was in the mid 1980's. The first attempts simply utilized early presentation software to develop color acetate slides to be used while presenting computer literacy material an the introductory course. The slides were well received in that they were clear, well though out, and easy for students to read and take notes from. They also allowed me to better control the coverage in a class, especially when trying to guarantee that multiple sections of the same course did in fact cover essentially the same material. The largest change they produced was that I was forced to produce very complete and detailed notes on the material which translated into more preparation time, not including the actual production time to produce the materials. The production time itself was significant due to the relatively slow speed of the color plotters which were the only output devices available at the time to adequately produce color images which could be projected onto a screen. A major disadvantage of these early approaches was the immense amount of time it took to edit material. It directly paralleled the days of type written material where one change often meant recreating an entire slide of sequence of slides. This often meant that changes did not occur as needed, especially in the computer courses where change was a way of life. Eventually, projection devices were developed which allowed black and white images to be projected onto screens. This meant that slides could more easily be changed to accommodate the fast pace of innovation in some of the subject areas. Consistently though, the math courses did not provide fertile ground like the computer classes, especially the applications/literacy courses which changed so rapidly.

The Intermediate Years

As color projectors became less expensive, and as scanners made it possible to capture images from text books and photographs, and computers developed more horsepower at lower prices, the stage was set for a more complete transition to multimedia slide shows which never made it to printed form. In fact, many textbooks now came with supplemental multimedia presentations and/or multimedia resources to construct your own with the instructor's materials. By this time I had already taken the time to scan in most of the images in the text by hand and had constructed "shows" which included bullet list notes, the heavy use of digital images, and even some full motion video when storage constraints permitted them.

The students responded favorably for the most part. One constraint we faced with early projection technology was that we found it necessary to lower the lights in the classroom to such a level that it was easy for the often sleep-deprived students to doze off in the middle of class. While this is not exactly a new phenomenon in college classes, we found no need to exacerbate the problem by dimming the lights. To offset this mind-dulling effect, I would stroll through the class thinking that my close proximity would discourage many from taking a cat-nap, especially those students in 8 am sections. I found the method did in fact work and make me feel closer to the students. I also tried to vary the activities by turning on the lights, and switching to another activity such as a live web search to locate resource materials on a given topic we were discussing. The use of videos, which were packaged with the textbook, did present enough of a break in the action to keep the interest level up. Unfortunately (or maybe not), one motivator which always seems to work is to relate that certain of these topics are definitely going to be on the next pop quiz, exam, assignment, etc.

During this phase, things were going well, except I had noted an increased emphasis by students in making materials available on the web. Multimedia on the web was (and for many of us still is) very difficult to construct beyond the implantation of the actual slides in a static web page format. Also, I found that posting "the slides" on the web encouraged students, whose attendance was borderline to begin with, to skip class since they could "get what had happened in class off the web page." While I believe a properly constructed web page can aid in learning, I found myself having less time to construct a really good site in addition to developing new and revising old multimedia materials. Most software manufacturers will tell you that transferring them to a web site is a no-brainer, but I have found that only the lowest level of information (static slides for example) can be easily be transferred and much of the content is lost in the process. While these slides may be posted in advance, I have consistently found that the material is far to volatile to post, in advance, what actually is going to be shown in class since most of my editing occurs 1-2 days in advance of their showing. After the fact however, slides may be posted as shown. There is still some danger that a student will skip in the hopes that they can make up the material by simply looking at (more likely printing out) the slides. This lets them avoid what I believe is an important pedagogical tool of having students re-copy notes and fill them in while in the midst of the discussion, wherever it may lead. This does slow the class pace down a bit, but that may not be all bad either.

Current Phase

As I get better at the development of multimedia materials, I find myself transferring my attention back to the actual process that is occurring in class. For example, I have recently found myself slipping dangerously close to too much dependence on the shows to get the material across and not putting enough emphasis on the critical thinking skills I wish to continue to develop in my students. I remember, disdain, my bad experience using "canned" acetate textbook slides which put nearly everyone (including me) to sleep. I am now reconsidering my approach again as I mature in the use of this collection of media and I believe I will reconstruct the process utilizing more "discovery approach" methods when appropriate, even though they tend to let us cover less material in the same time frame.

Future Directions

While I still believe strongly in the use of multimedia materials inside and outside the classroom, I hesitate to suggest that it has caused a fundamental pedagogical change in the teaching/learning process. It has enabled us to put forth learning materials which are in fact more complete and apparently more enticing to the generation of student who has been brought up with M-TV. Whether it has effected a change in the way students learn is yet to be determined scientifically. It has definitely effected the way in which I prepare for classes in that I must continually revise my materials, especially in the literacy classes, to keep up with the frenetic change in that industry. I am now in the process of exploring more collaborative approaches to learning to use in conjunction with multimedia materials.

I am (as many of us are) currently implementing the collaborative technology necessary for live chat sessions and threaded web-based discussions. In particular I have just co-written a joint proposal, with the technical school next door, which will allow our students to interact with each other within USC Sumter as well as between USC Sumter and Central Carolina Technical College. Besides satisfying some of the buzzwords and in vogue phrases that are so neatly implanted in South Carolina's now famous "Performance Funding Indicators," these collaborative learning efforts may end up representing a more fundamental pedagogical change in the way learning takes place at the University of South Carolina Sumter.

Application of FrontPage 98 to the Development of Web Sites for the Science Division and the Center for the Advancement of Learning and Teaching (CALT) at Anne Arundel Community College

Bruce Bird
Physics Department
Anne Arundel Community College
101 College Parkway
Arnold, MD 21012
1-410-541-2262
blbird@mail.aacc.cc.md.us

Introduction

Over the last two years Anne Arundel Community College has undergone a rapid infusion of technology onto the campus. Last summer I was given the opportunity to develop computer support for faculty through our Center for the Advancement of Learning and Teaching (CALT). As part of that effort I developed a Web site for CALT (http://www.aacc.cc.md.us/calt)In the fall I also became the Science Division computer coordinator. Part of my responsibility as coordinator is to develop a Web site for the Science division (http://www.aacc.cc.md.us/science) and help science faculty produce their personal Web pages.

Previous to this, about two years ago, I set up my own Web site with a local ISP and so had enough experience with HTML to find it extremely time consuming for creating Web pages. In addition once I had placed a large number of links on my site I found it impossible to ensure that all links were unbroken. When Microsoft FrontPage 97 became available the three features that caught my attention were its elimination of the need to use HTML, its WYSIWYG word processor style interface, and its site management capabilities that includes automatic detection of broken links, both internal and external.

Choosing software and hardware

We have always been on a technology learning curve. The difference at the present time is the rapid simultaneous introduction of new technologies (computer, network, and software) with each of these technologies spawning several companies who all proclaim the superiority of their product. How can anyone find the path to technology nirvana? I have no absolute answers, only suggestions based on my experience stumbling along my own learning curve and trying to help others to move forward on theirs.

Some criteria I think important for selecting hardware and software are:

1. it needs to be easy to learn and use
2. cheaper is not always better but it helps
3. it should incorporate the "latest and greatest" technology advances
4. it has good support - both from the company that produces it, as well as from the Web community and the publishing industry

5. it has been recognized as an outstanding product by reviewers
6. it is compatible with existing hardware and software

Applying the above criteria to several available Web site development programs I decided to use Microsoft FrontPage 98.

Microsoft FrontPage 98

At the end of this article I have placed links to the Microsoft FrontPage 98 Homepage as well as several FrontPage 98 sites that provide useful information about this program. The Microsoft site will give you a detailed description of all the important features of this program.

The FrontPage 98 package comes with five major components:

1. FrontPage Explorer – used to create, change, maintain, and administer Web sites
2. FrontPage Editor – used to create, edit, and view Web pages using a WYSIWYG word processing interface
3. FrontPage Server Extensions – installed on your Web server to support certain FrontPage components
4. Microsoft Image Composer – an image editing program
5. Microsoft Personal Web Server – a fully functional Web server program

In order to use the more advanced features of FrontPage 98, the FrontPage 98 server extensions must be loaded on your Web server.

Some of the features of FrontPage 98 that I have found particularly useful in creating the CALT and Science Web sites are:

1. The word processor style interface with WYSISWYG has several icons and controls that are identical to those used in other programs that make up the Microsoft Office Suite.

2. Creating tables and forms is essential a click and drag or dialog box interaction with no need for the user to do HTML formatting or CGI programming.

3. Web hyperlinks can be inserted automatically in your Web page by using a dialog box and viewing the desired site in your Web browser, thus avoiding errors in typing URLs.

4. Text files, images (GIG or JPEG), hit counters, and search engines can be inserted in a Web page with just a single menu selection and a few choices from a dialog box.

5. Copy and paste can be used to move components quickly from one Web page to another or from another Microsoft application into the Web page and all links remain active.

6. Automatic link verification in FrontPage Explorer provides a way to easily check the status of all internal and external URLs.

In general we have found FrontPage 98 to be a useful and effective program for creating and managing Web sites. Some things to watch out for in using this program are:

1. We have not yet been able to successfully install the Microsoft Personal Web Server (MPWS) on any computer connected to the college network. Luckily, for our purposes we don't need it. What works for us is to install everything except the MPWS on individual faculty computers. We have not had any problem installing MPWS on computers that have a modem connection to an ISP.

2. Because of differences between Microsoft and Netscape browsers it is essential that Web pages created in FrontPage 98 be viewed with both browsers. I have sometimes found differences in the positioning of page elements, color differences, and unwanted text layering. These differences have been easily corrected by editing the original structure of the Web page.

Faculty Development

The rapid introduction of technology into a college necessitates an associated faculty professional development effort. The problems and approaches that various colleges have taken in this effort have been extensively discussed and links to some of these materials are listed at the end of this article.

Science Division Web Site

Building the Science Division site provided the opportunity to acquaint science faculty with the capabilities of FrontPage 98 and also increase their awareness of the nature and extent of science related sites on the Web. Approximately 30 science faculty and staff were introduced to FrontPage 97 and 98 through a two-hour, one-on-two session with the program. The FrontPage 98 program was loaded on a computer in the science building and was available to all science faculty. They were then asked to prepare their own personal Web pages. Most opted, however, to not use FrontPage 98 but instead to prepare material in a word processing program and give it to me to insert onto their personal Web page. A few chose to use templates prepared by our Media Production Services or to use pages they created on personal sites located on local ISPs. Except for two staff member none chose to use FrontPage 98 to create their own Web page. Several expressed interest in doing so but said lack of time was the main reason they didn't use it. Many faculty believe that if the program were available on their office computer so that they could directly edit their pages on the Web then they would be much more likely to use it.

A more successful approach that encourages faculty use of technology has been associated with the development of Web courses through the distance learning center of the college. The distance learning center (http://www.aacc.cc.md.us/diseduc/) has set up an On-Line Academy for faculty developing new Web courses. All faculty who participate in the On-Line Academy have FrontPage 98 installed on their office computer and receive a copy of the book, Microsoft FrontPage 98 At a Glance by Stephen L. Nelson, Microsoft Press, 1997. They then participate in two, three hour, computer workshops run by myself and Marshall Lucas, the college webmaster, using computer exercises for FrontPage 98 that I have developed. The workshop materials provide a list of procedures (detailed step-by step instructions) for carrying out common tasks. A copy of the workshop materials has been placed on the Web under restricted access for workshop participants. Marshall and I provide follow up support if needed. Two science faculty are presently developing Web courses and so far the combination of reduced teaching load, FrontPage 98 on their office computer, introductory workshops, text, available one-on-one support, and a well-defined goal appears to be successful.

CALT Web Site

The purpose of the CALT site is to serve as a teaching and learning resource for the college community. Although the primary audience is the faculty and staff at AACC, the public nature of the Web provides an opportunity to serve the local community as well as any visitor from the Web.

Recognizing that the intended audience occupies a broad spectrum of backgrounds and interests I have attempted to:

1. increase awareness of the types and extent of resources on the Web;

2. use technology to teach technology;

3. provide examples of the applications of software and hardware to teaching and learning.

Although search engines provide a way to locate information on the Web I believe it is still useful to provide focused, annotated list of links, particularly for persons new to the Web. These are provided on the CALT site by pages of links to tutorials, teaching, learning, educational technology, higher education, electronic publishing, trends, examples, how to ... , and CALT resources.

The increasing number of high quality tutorials available free on the Web provides a way to use technology to help people move up the technology learning curve. The tutorials section provides links to tutorials on the Internet, search engines, evaluating Web sites, browsers, Windows 95, Microsoft Word, and Microsoft PowerPoint.

While links are useful they are not enough. The rapid introduction of new software and hardware tools (particularly when these new tools are easier to use, have a shorter learning curve, and are more effective than previous technology for teaching applications) emphasizes the importance of making teachers aware of these new tools and what can be done with them. Tools such as FrontPage 98 are a good example of this. The time consuming use of HTML and CGI scripts to produce Web pages is replaced by essentially a word processing procedure. One of the best ways to convey what new tools can do is by means of examples. At the CALT site I have started to include some examples of what can be done with software (FrontPage 98) and hardware (Sony digital camera). This effort will continue along with examples of what can be done with new tools (Flash, Dynamite, Java Beans, Active X, dynamic HTML, etc)

Conclusions

Developing the CALT and Science sites along with previous experience presenting talks, workshops, and one-on-one tutoring with faculty on multimedia topics, authoring programs (Toolbook, Authorware, Director), and Web tools suggest that there is no one way to help faculty make the journey from anxiety to enjoyment (or at least comfort) in the use of technology in their teaching. Some general observations are:

1. faculty need time and appropriate resources;
2. whenever possible software must be on the faculty office computer;
3. workshops may be OK to get started but there needs to be a continuous program of support;
4. decide on the product (interactive teaching module, Web page, Web course, etc) before worrying about the software needed to produce the product. There is a lot to be said for just-in-time learning. It avoids a lot of time sitting in workshops learning material you will rarely use;
5. small projects that produce useful products are the best, at least when your starting up the learning curve;
6. concrete examples of applications of technology to teaching are more motivating than general discussions of educational technology;
7. faculty must be assured that when they develop technology dependent teaching materials that the teaching environment will allow these materials to be used.

Of course every college is unique in terms of mission, personnel, structure, and resources so the suggestions made in this article will have to be modified for each campus community.

Resources

Microsoft FrontPage

FrontPage 98 Home Page (http://www.microsoft.com/frontpage/default.htm)

Product Overview of FrontPage 98
(http://www.microsoft.com/frontpage/productinfo/overview.htm)

Mike's FrontPage FAQ Archive (http://w-ww.simplenet.com/frontpage/) "contains a collection of answers to FAQ's related to FrontPage and Web site development. These answers have been extracted from the newsgroup and other sources and whenever possible the credits to the contributor are maintained."

Support Area for Microsoft FrontPage (http://www.pmpcs.com/support/frontpage.htm) "These pages are intended to enhance the support already available for Microsoft FrontPage."

FrontPage User Tips by Keith Parnell's (http://www.frontpage.to/support/) provides tips on using Front Page 97 and 98.

CALT - FrontPage 98 Examples (http://www.aacc.cc.md.us/pubweb/) demonstrate some of the features of FrontPage 98. The CALT site is built using FrontPage 98.

Faculty Development

" Incentive Programs to Support the Use of Instructional Technology by Faculty at a Major Research University and a Leading Liberal Arts College", by Connie Vinita Dowell and Todd D. Kelley.
http://www.cause.org/information-resources/ir-library/html/cnc9660.html

"Faculty Training for Technology", Fax-Back surveys conducted through Syllabus magazine and SyllabusWeb http://www.syllabus.com/fb_05_28_97res.htm

"Abstracts of Suggested Readings on Information Technology in Higher Education", by Linda A. Suskie http://www.imir.iupui.edu/itreadings.htm

"Preparing Faculty for Instructional Technology: From Education to Development to Creative Independence", by Karen L. Smith
http://www.cause.org/information-resources/ir-library/html/cem9739.html

"The Professional Practices of Faculty and the Diffusion Of Computer Technologies in University Teaching", by Michelle Proulx and Brian Campbell.
http://www.sociology.org/vol002.003/proulx.article.1997.html

From One Generation to the Next: The Evolution of Library Services at Cumberland College

John Burch
Technical Services Librarian
Hagan Memorial Library
Cumberland College
821 Walnut Street
Williamsburg, KY 40769
Phone: 606-539-4160
jburch@cc.cumber.edu

BiblioFile, the First Generation

BiblioFile is a PC-based library automation system that Cumberland College purchased from The Library Corporation (TLC). The bibliographic data was stored on CD-ROMs that were created by The Library Corporation from MARC records provided by the librarians at Hagan Memorial Library on floppy disks. This way of updating bibliographic data was not satisfactory because books that were available on the shelves would not appear on the library's catalog until new CD-ROMs were created. This was the only major concern we had with the system. While this situation was annoying, it was not worth spending the amount of money that was necessary to get a different system. This all changed when the decision was made to create a campus network. Two faculty committees believed that a web compatible automated library system was a necessity for the campus network. The Library Corporation was at the time working on developing a web component for their catalog but their system was not available when Cumberland College was ready to build a campus network.

The Second Generation, SIRSI Unicorn

SIRSI Unicorn was the automated library system that was selected to be the new library automated system. It runs on a Sco platform that is loaded on a Pentium Pro Server with dual processors and 128 MB of RAM. Two of the librarians serve as system administrators of the system even though both of the librarians have limited computer skills. The money is not available to hire a qualified systems administrator for the library. There is some limited qualified assistance available on campus but the librarians felt it was in the best interest of the library to keep total control of the server. The decision to use Sco proved costly several months later when the campus network administrator decided to use a Windows NT server for the campus network. There has been some compatibility problems between the two servers that have necessitated a great deal of effort to straighten out.

SIRSI did enable us to have a secure database for the first time. With BiblioFile, there was one password that served as an override to the whole system. This was a problem because everyone, including the students, needed to know it. With SIRSI, every staff member has a unique password that grants them access to what they are cleared to use. For example, students do not have the ability in SIRSI to delete fines, which they could do using the old system. This level of security has enabled the staff to work effectively without the worry of someone corrupting their data.

Web Compatibility

As stated earlier, web compatibility was a must for the new system. SIRSI required the librarians to create a web page in order to provide an interface for the WebBot, which is the web version of Unicorn. It was decided that we should explore ways to provide information through the web page in order to address student complaints that had arisen through the years. The main complaint of course was that we were not open enough hours. With a staff of eight, four professionals and four support staff, we are stretched thin to be open the hours that we do operate and simply cannot extend our hours without hiring new employees. We decided to subscribe to full-text databases and make them available through the web page. This would allow the students to access information sources twenty-four hours a day and seven days a week. We have not had the system long enough to quantifiably examine student usage of the library but we seem to have fewer library users in the library but are providing more information services than ever before. The difference is that our patrons do not have to come to the library to access information but can do so from their offices or homes.

Hagan Memorial Library Cumberland College

Welcome to the Hagan Memorial Library Web Page! The purpose of this page is to both introduce you to the Cumberland College Library and its policies as well as to provide links to help fill your information needs.

Hagan Memorial Library

- Cumberland College Automated Catalog
- Cumberland College CD-ROM Resources
- First Search
- Proquest
- Britannica Online
- Congressional Quarterly and CQ Researcher
- Galenet (Including Contemporary Literary Criticism, Dictionary of Literary Biography, Gale Business Resources, Peterson's GradSearch)
- Well-Connected Health Resource

- JSTORJournals

- The Reference Desk

Hours

- Fall and Spring Semesters
  - 7:45 A.M. to 11:00 P.M. Monday-Thursday
  - 7:45 A.M. to 5:00 P.M. Friday
  - 9:00 A.M. to 5:00 P.M. Saturday
  - 2:00 P.M. to 11:00 P.M. Sunday
  - The library is usually closed during holiday and school breaks. Changes in hours are posted at the front entrance of the library.
- Summer School hours will be posted at the end of the Spring Semester.

Interlibrary Loan Forms

- Interlibrary Loan Request Form
- Interlibrary Loan Renewal Request Form

Library Policies

- Circulation Policies
- Copyright Policies
- General Policies
- Interlibrary Loan Policies

Staff of the Hagan Memorial Library with Telephone Numbers

- Ms. Becky Bledsoe .... Media Library Assistant ....ext. 4288
- Mr. Ron Bunger .... Instructional Media Librarian ....ext. 4428
- Mr. John Burch .... Technical Services Librarian ......ext. 4160
- Ms. Stacey Estes .... Interlibrary Loan Supervisor ......ext. 4161
- Ms. Hilda Price .... Circulation Supervisor ....ext. 4482
- Ms. Sherita Holder .... Acquisitions Assistant ....ext. 4161
- Ms. Ru Story-Huffman ...... Public Services Librarian ....ext. 4426
- Ms. Jan Wren .... Library Director ....ext. 4328

Cumberland College Departmental Resources

- Department of English
- Department of Mathematics, Physics, Geography, and Astronomy
  - Links to Other Sites for Mathematics
  - Links to Other Sites for Physics, Geography, and Astronomy
- Department of Religion and Philosophy

Links from the World Wide Web

Citing Electronic Resources

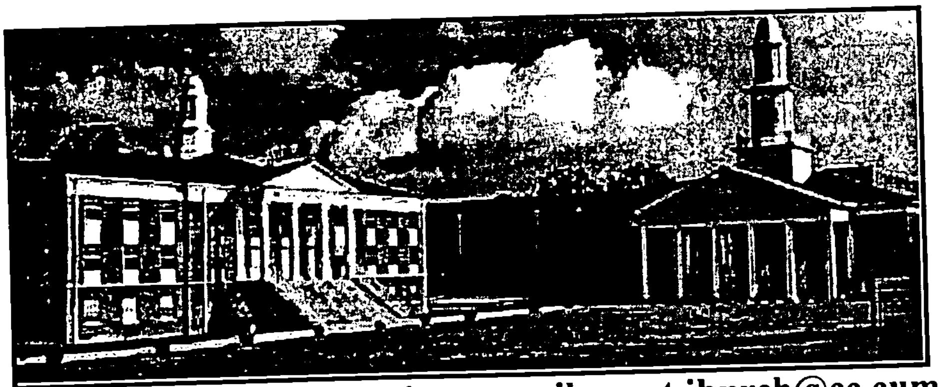
- A Brief Citation Guide for Internet Sources in History and the Humanities by Melvin E. Page
- APA Publication Manual Crib Sheet by Russ Dewey
- APA Style Citation Analysis with Examples
- Beyond the MLA Handbook: Documenting Electronic Sources on the Internet by Andrew Harnack and Gene Kleppinger
- Chicago or Turabian Style by the University of Wisconsin-Madison Writing Center
- Guide for Citing Electronic Information by K. Wagner
- MLA-Style Citations of Electronic Sources by Janice Walker

Links to Other Libraries

- Library of Congress
- Portico , The British Library's Online Information Server
- Stanford University Libraries & Academic Information Resources
- University of California / Berkeley
- University of Kentucky Libraries
- UTK LibLink University of Tennessee, Knoxville Libraries

Resources for Ready Reference

- Decisions of the U.S. Supreme Court Service provided by Cornell University.
- ECHO: EuroDicautom Translating Dictionary for the European Union
- The Internet Public Library (IPL) Ready Reference Collection
- LawCrawler
- PubMed (National Library of Medicine's Medline and other Databases
- Quick Reference University of Texas at Austin
- Ready Reference Using the Internet by Ellen Berne,Winsor School in Boston
- Statistical Abstracts of the United States
- Superintendent of Documents Home Page: GPO Access U.S. Gov. Printing Office
- The United States House of Representatives
- The United States Senate
- University of Michigan Documents Center (Government Documents)
- WebElements: The Web's Periodic Table
- Webliography : A Guide to Internet Resources



If you have comments or suggestions, email me at jburch@cc.cumber.edu

A Literature Course on the Web: Using the Intranet to teach The Novel?

David Field
Department of English
Depauw University
Greencastle, IN 46135
dfield@depauw.edu

Three years ago, Carl Singer (the Director of Academic Computing at DePauw and currently Past President of ASCUE) helped redesign a course (The Novel) that I have taught every year since I came to DePauw sixteen years ago. Carl created a program that integrated images, sounds, and text into the flow of the course. Using a computer and projector, I became able to make clear the context of the novels in intellectual history. Not only could students read about the arts and sciences at the time the novels were written; they could see the paintings (as well as maps and portraits of the artists) and hear the music (sometimes even the authors' voices). We had a timeline so that I could click on a date, allowing students to see other developments at that time--and to see the ways that each novel fit into the historical framework.

Even from the beginning, Carl and I both wanted a component of the class that students could browse outside of the classroom setting. As we worked on the in-class portion of the program, we both became aware that the "laboratory" component of the class would have to be different from the in-class aspect. The laboratory program would need to contain more explanatory text, and it might contribute more fully to the goals of the class if it included certain features that we might not discuss in class. To be specific, if the lab portion of the class dealt with certain major themes, then we would be free in class to let the discussion move away from these major issues and explore tangential issues that came up unexpectedly.

For the past year, I have worked on that laboratory component of the class, and DePauw has put in place a fiber-optic network and added computers so that all students have access to multimedia computers in laboratories and living units. The lab component thus became more than an element that students will work on at particular times when they gather in a computer lab--now they will be able to browse this multimedia site from their living units as well as other locations around campus.

During the time that I was constructing the site (and up until late August 1998), the site will be available on the Internet (http://www.depauw.edu/~dfield/frames.html). At the end of August, access will probably be restricted to the DePauw Intranet.

The site includes a three-part Introduction, consisting of the images I use during the first class as well as other images and references that go beyond the specific topics I cover in that class. The Introduction and other sections have links to a bibliography, a glossary of terms used frequently in the course, and sites that deal with the major themes of the course. In addition, there is a section devoted to each novel, with images of the authors, reproductions of paintings, sound files of music, and discussions of the scientific developments from the period. I have also created an annotated set of links to the literature, art, music, and scientific sites on the Internet related to the books we read.

The course site has a complete set of individualized assignments for the entire class. The students may go to the class list, click on their own names, and get an assignment which includes a link to a scholarly site on the Internet (for example, the Victorian Web, Pemberley--the Jane Austen site--or Zembla--the Nabokov site), where they have a reading assignment. After reading the assigned section, they must write a report on the article and give an oral report to the class. Right now, I do not have them use multimedia in their reports to the class, but I think that they will soon be able to use programs to compose reports in HTML, and I hope to implement such a project in the future.

I also plan to add an announcement page so that the class can check for any last-minute changes in assignments, reminders about class, and questions to consider before the class meets.

A Low-Cost Solution for Campus Connectivity

Chris Gibbs
Don Olive
Campbellsville University
con@campbellvil.edu

Abstract

The need for cross-platform connectivity is problematic for most small educational institutions; Linux and PPP afford a low-cost and effective solution by leveraging existing telephone network infrastructure. We will present a basic "HOWTO"' tutorial for the creation of a private internet service provider with all the standard inter/intranet services including web hosting, e-mail, firewalling, multiprotocol routing, and gatewaying, etc. Issues such as hardware requirements, users account administration, and resource control will be explored.

(Chris and Don's paper was not available at the time the Proceedings went to press. They will supply copies of their paper at their talk.)

Development of a Real-Time Intelligent Network Environment.

Anatoliy Gordonov: GORDONOV@POSTBOX.CSI.CUNY.EDU
Michael Kress: KRESS@POSTBOX.CSI.CUNY.EDU
Roberta Klibaner: KLIBANER@POSTBOX.CSI.CUNY.EDU
Computer Science Department
Staten Island College
City University of New York,
2800 Victory Blvd. 1N-215, SI, NY 10314
Tel. (718) 982 2850
Fax. (718) 982 2856

Introduction.

The traditional computer network offers instructors the ability to share data, applications, and files with their students. Students can create group projects by sharing code and ideas utilizing this network. E-mail, file transfer protocols, and the Internet can be found in most laboratory environments and can be used for classroom activities. In general, all of these network related activities may be described as "sharing and communication". Benefits of employing networks in the classrooms include student access to sharable resources, enhancing communication between student and professor, between students inside a working group, between working groups, etc. As a result of this enhanced communication, an instructor has to process more information during the class hours. The ability to quickly analyze incoming data and issue an appropriate response to the students may considerably enhance the quality of teaching. Obviously, the faster the response the better results it produces. Ideally, we want to achieve real-time help, where an instructor may offer assistance to students at the time at which the problem arises. Intervention at this time is most beneficial to a student because he/she is currently thinking about the problem and is highly motivated to solve it.

To provide real-time help the instructor needs to have the information about the students' performance. The solution may be found in introducing new intelligent functions which are incorporated into the classroom network environment and provide real-time evaluation of the students' performance. Based on this evaluation, the network filters out the information about the students who currently do not need help. It delivers to the instructor only the data about the students who really need help. This relieves the burden of continuous class monitoring and allows the instructor to concentrate on his/her responses. Moreover, this approach allows the instructor to offer help to some students even if these students, for some reason, did not seek assistance.

An intelligent network capable of monitoring each station and able to determine which questions or topics are causing difficulty would be a pedagogically valuable tool. By providing real-time performance analysis and dynamically reconfiguring the question package, the intelligent network would be able to customize the drill for each class [1,2,3]. Using the instructor supplied timings, students encountering difficulty can be offered a review of the material and step-by-step solutions. Students who have mastered the lesson can be offered more challenging work.

Model of an Educational Intelligent Network.

The educational intelligent network model helps us offer new network functions that can be very useful to an instructor in his/her efforts to evaluate students' performance. Provided in the real-time, this evaluation may become a beneficial source of information about both the students' performance and quality of the instructor's materials (tests, drills, etc.). As an example for our analysis in this paper, we consider specially designed drills. The main goal of these drills is to allow students to independently solve a number of problems based on current lecture material. Drill assignments reveal problems and misunderstandings which students may have. An assignment is presented to a student on his/her screen, and the solution is evaluated by a computer. The structure of a drill includes three variants of each assignment: regular, advanced, and simplified. By default, a drill starts on a regular level. If a student is lost, makes too many mistakes, and/or spends too much time to find the correct answer, he/she may be temporarily switched to a lower level; and then (may be with the instructor's help) return back to a regular level. The best students in the group may be switched to a more advanced level.

Let us consider a model of the network in which these drills are implemented. In general, any network may be presented as a set of nodes interconnected through a communication medium. Let $P = \{ P_i \}$ a set of network nodes where $i = 1,\dots,N$, N represents the number of the nodes in the network. We are considering a classroom equipped with N student stations and a teacher station. Assume that any node may be in one of two states:

- 0 corresponds to the situation when a student makes too many mistakes or spends too much time on some problems compared to a schedule assigned by the instructor.
- 1 means that the student's achievement is on schedule.

Let $F = \{ F_i \}$ - a set of boolean state functions. Each F_i corresponds to P_i . Let S(t) be a schedule function. S(t) defines the minimum number of problems a student is to solve at time t. S(t) is defined by the instructor before a drill starts.

$F_i(t_0) = \{0$, if $X_i(t_0) < S(t_0)$ or 1 if $X_i(t_0) \geq S(t_0)$ where $X_i(t_0)$ represents the number of problems solved at time t_0 by a student i $\}$.

At any moment, we have a state vector:

$F(t) = \{ F_1(t),\ F_2(t), \dots\ F_n(t)\}$ defines the state of the network at the moment t. It shows which nodes (students) are currently on schedule and which are not. Because F is a binary vector, the network will be split into two subsets of nodes: one - for the students who have normal performance and the other - for the students who are currently behind the schedule.

Given m performance evaluation points for which S(m) has been defined and n nodes, the dynamics of the student performance can be represented in the matrix (figure 1).

For each F_{ij} $1 \leq i \leq m$, the number of performance evaluation points, $1 \leq j \leq n$, the number of nodes.

Each line of this matrix corresponds to a particular moment and constitutes a snapshot of the network state. Each column shows the dynamics for every participating student during the drill.

Let us assume that the number of problems to be solved in a drill is also m. In this case, the schedule function S(t) may be specified at the moments $t_1, t_2, \dots t_m$, and it will show how many problems are to be solved by a moment t_i (see figure 2). X_j (t) represents the real-time performance of a student who works on the node # j. The comparison of S(t) and X_j (t) defines the state function of the node # j. The situation when $F_j = 0$ will be displayed to the instructor in order to notify him/her about the students who are currently behind the schedule. The instructor may provide some real-time help to a student or a group of the students related to the problem in question. One of the most efficient ways to provide the real-time help is to employ Analog/Digital Video Network (ADViNet) [2,5]. ADViNet allows the instructor to send images from teacher screen to any designated group of student screens in the classroom without disturbing the rest of the class.

In the above analysis, we deal with the information about a particular student (network node). It allows us to understand the individual student's problems and provide real-time help.

Unfortunately, the information about the individual student's performance does not enable us to provide more general in-depth analysis of the overall class performance.

Let us introduce variable Z which can take any value from the set $\{ Z_1, Z_2, \ldots Z_m \}$ where $Z_i = \sum^{n}_{j=1} F_{ij}$; $0 \leq i \leq m$. Z shows the overall performance of the class. $Z = Z_i$ displays how many students are on schedule at the moment t_i. If Z is too small, this would indicate that the work is too hard, or the material has not been thoroughly explained; if Z is too large, a more difficult problem set has to be supplied. Thus, the value of Z permits us to analyze the quality of the test or drill. An example of using variable Z for the evaluation of the overall class performance is shown in figure 3. There are several zones which should be considered separately. Zone A ($Z \leq Z_{MIN}$) shows that the current problem(s) is (are) too complicated for the majority of the students in the class, and real-time help from the instructor is needed. Zone B ($Z \geq Z_{MAX}$) reveals the opposite situation when the problem(s) is (are)too simple. In this case, the network can be automatically or manually switched by the instructor to the level with higher requirements. These several levels are specified in the information structure which defines a drill.

If we consider two consequent moments t_i and t_{i+1} and $Z_i - Z_{i+1} > \Delta Z_{MAX}$ (zone C), then the current problem is too hard for the class, and help is required. Although it may be the time schedule that should be changed.

If $Z_{i+1} - Z_i > \Delta Z_{MAX}$ (zone D), then a problem is probably too simple, and a higher level should be used. The algorithm of real-time performance analysis is shown in figure 4.

The algorithm of real-time performance evaluation is included in the application layer communication protocol. The goal of this protocol is to make the software for real-time class assistance more structural and standardize the "ground" for the future development. The incorporation of state computation and performance analysis in the protocol allows the network to derive control operations from the algorithmic steps of network subfunctions and make the necessary dynamic reconfiguration. The reconfiguration may involve automatic real-time hardware reconnection and set up and/or the alteration of data in the Management Information Base that will effect the network functioning. In other words, after the reconfiguration, the number of nodes involved in a session or the way (or order) of their communication may be changed [4].

Conclusion

We believe that the use of traditional computer networks in a classroom can provide real-time assistance to the instructor. In this paper, we tried to analyze one possible implementation of an intelligent network by incorporating intelligence into the application layer protocol. As an example, we have considered specially developed drills. The state of each network node is defined as a function of a student's performance during the course of the drill. The analysis of a state of the nodes in real-time permits algorithmic reconfiguration of the network resources. As a consequence of this reconfiguration, the instructor will monitor only those students who are behind the schedule. This, in turn, allows the instructor to concentrate on the analysis of the students' problems and the ability to provide real-time help rather than attempting to identify those students who are experiencing difficulty.

This paper introduces work in progress. In the future, we are going to extend intelligent network functions to other instructor's activities.

References

1. Anatoliy S. Gordonov, "Analog/digital networks for real-time video distribution with extra computational capabilities". Electronic Imaging Conference. EI'97. San Jose. CA. Febr. 97.

2. Gordonov, A., Kress, M.,"Educational Analog/Digital Video Networks", AACE ED-TELECOM Conference, Calgary, Canada, June 1997.

3. Gordonov, A., Kress, M., Carlin, M.,"Video Networks in the Electronic Classroom Environment and Analysis of Student Preferences in the Development of a Network-Based Video Distribution System", 1997 ASCUE Summer Conference, Myrtle Beach, South Carolina.

4. Ahamed, S. & Lawrence, V. (1997). Intelligent Broadband Multimedia Networks, Boston: KAP.

5. Gordonov, A., Kress, M., Blank, A., Moroh, M., "ComWeb: An Electronic Classroom for Teaching Computer Literacy", Computers and Education. An International Journal. Vol. 29, No.4, pp181-187, 1997

$$M = \begin{pmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \dots & \dots & \dots & \dots \\ F_{m1} & F_{m2} & \dots & F_{mn} \end{pmatrix}$$

Figure 1

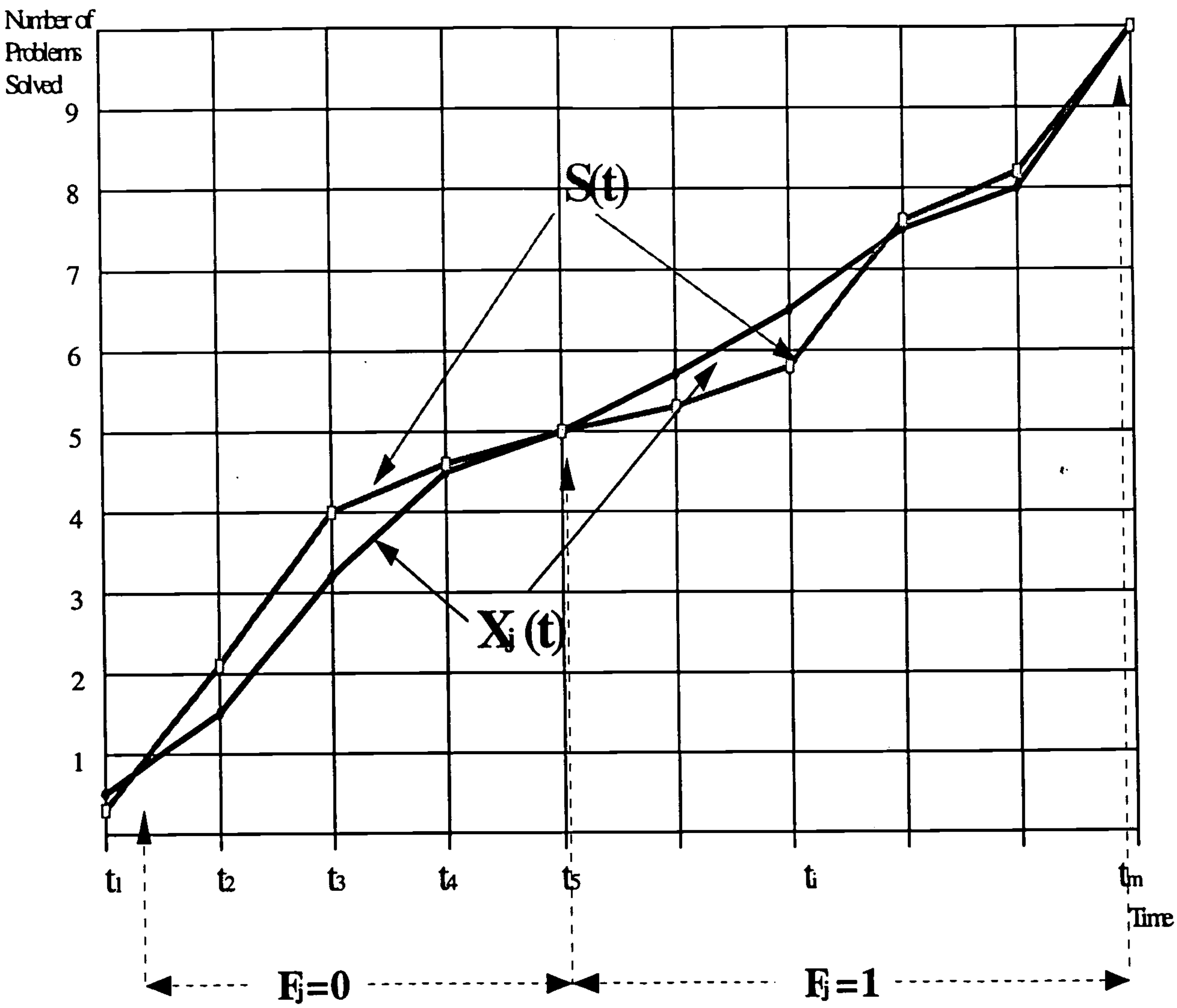


Figure 2. A Student's Performance

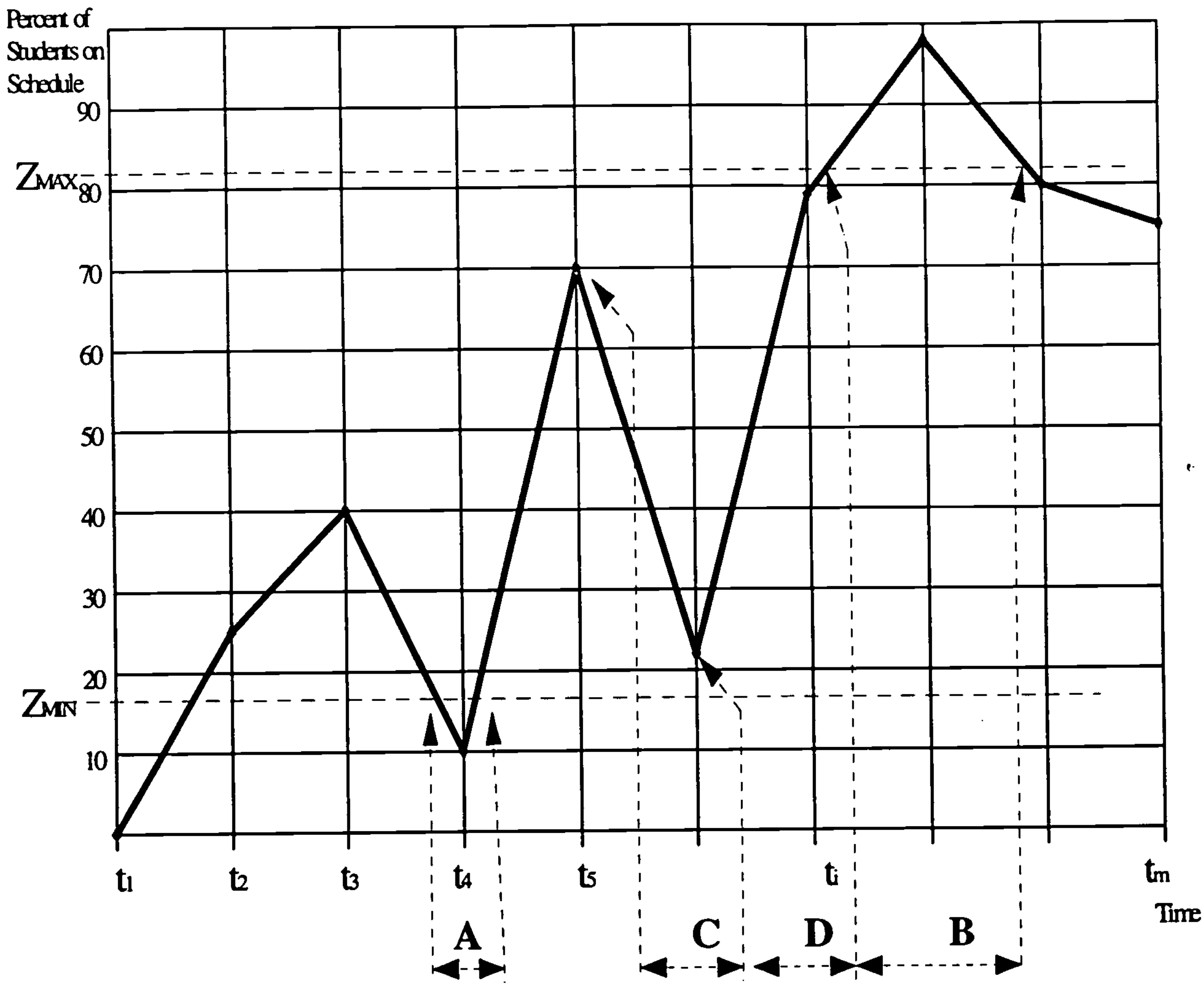


Figure 3. Overall class performance

Algorithm of Real-Time Performance Analysis

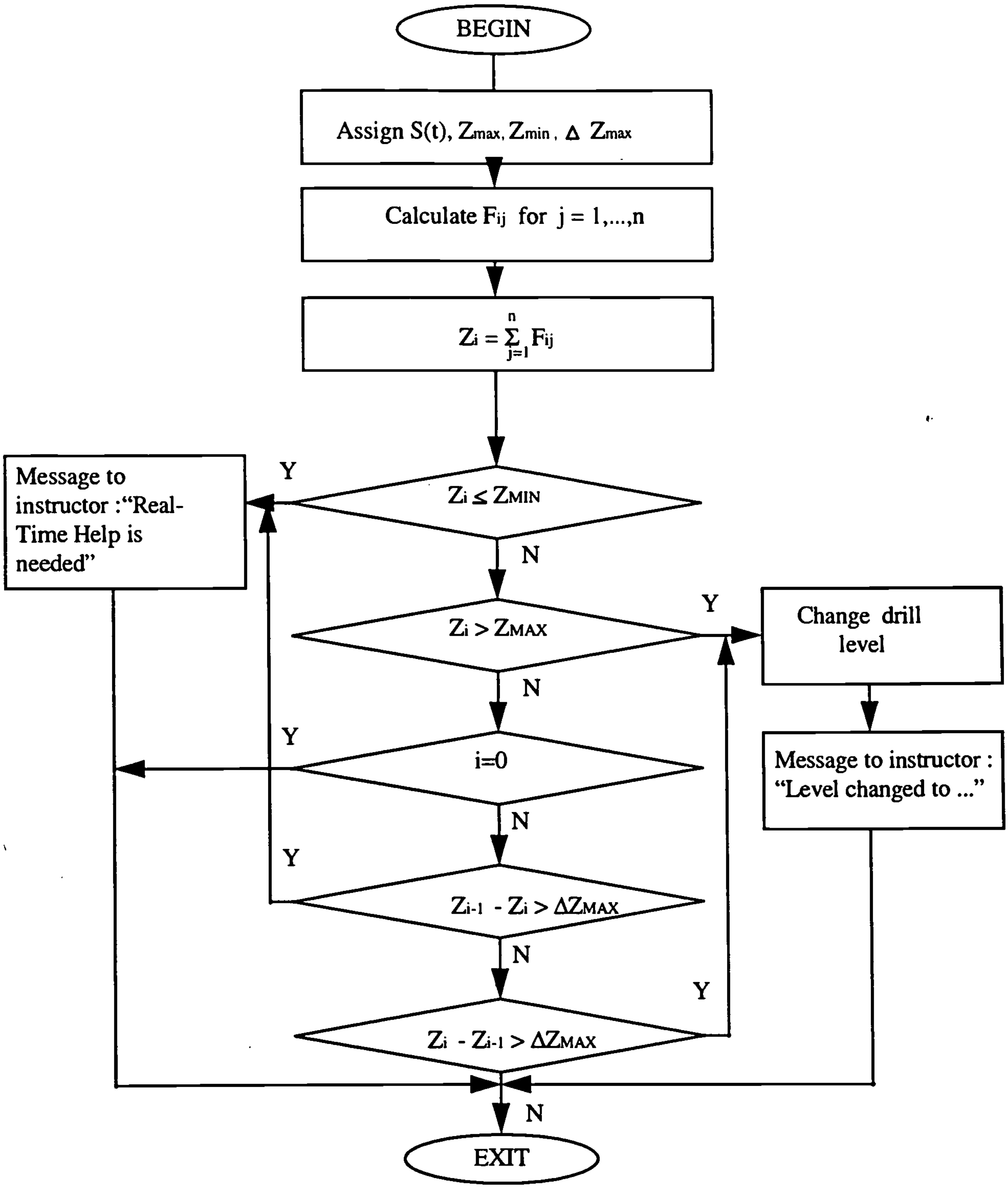


Figure 4

Serving Grades Over The Internet

Dr. James K. Harris
Professor of Computer Science
Francis Marion University
Florence, SC 29501
Email: harr@fmarion.edu

Introduction:

Many times bewildered students have entered my office claiming no knowledge of their grades and leave looking shocked. To keep this from happening (at least in my office), I have created a grade server, allowing students to access their grades over the Internet from my home page. Using a CGI program written in Visual Basic, the grades are read directly from an Excel spreadsheet and presented to the requester (after entering a password). The grade for each quiz, assignment, test, etc. is presented along with the class average and rank for each. This way the student knows exactly where they stand in the class which is especially useful when the grades are curved. In this paper, I demonstrate the grade server, show how it is implemented, and show how it can be modified to allow for different spreadsheet formats other than Excel.

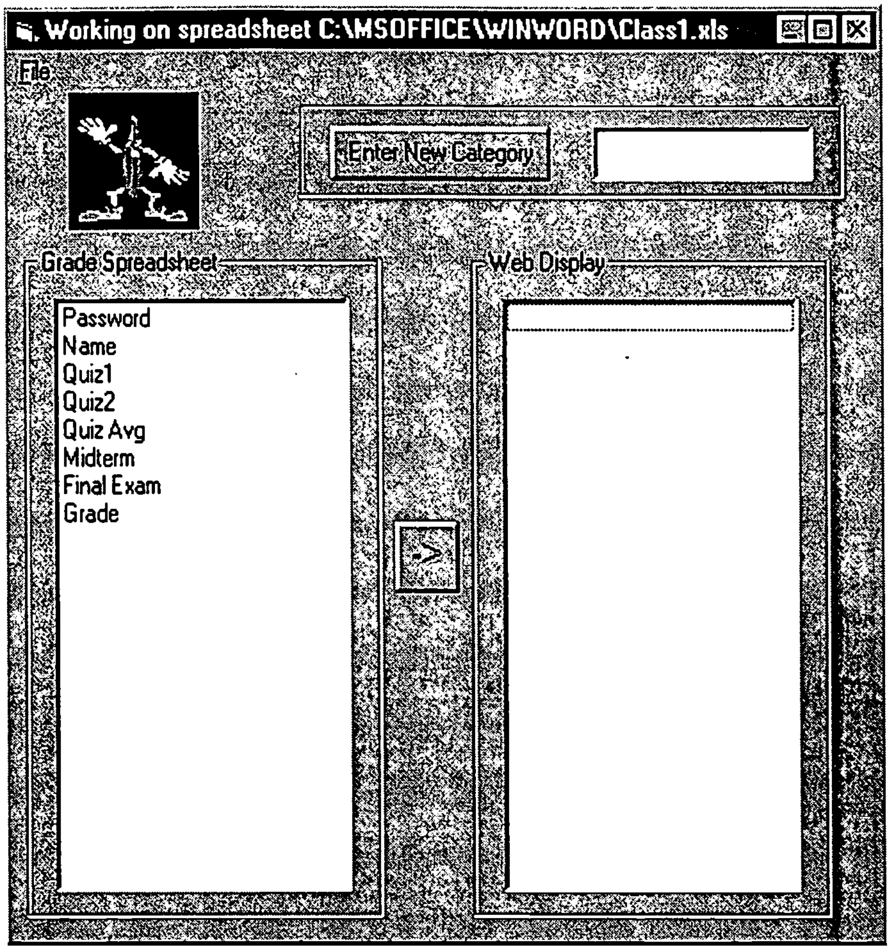
The Grade Server:

In this section, I explain how to use and set up the grade server (in case you would like to use it yourself). The software and instructions are downloadable from my web page at okra.fmarion.edu. First, the spreadsheet must be set up in a specific format. The spreadsheet below acts as a guide.

Password	Name	Quiz1	Quiz2	Quiz Avg	Midterm	Final	Grade
400801	Bauer,Linda						
764311	Black,Ben						
989519	Jones,Ralph						
698390	Smith,Deon						
	Average						

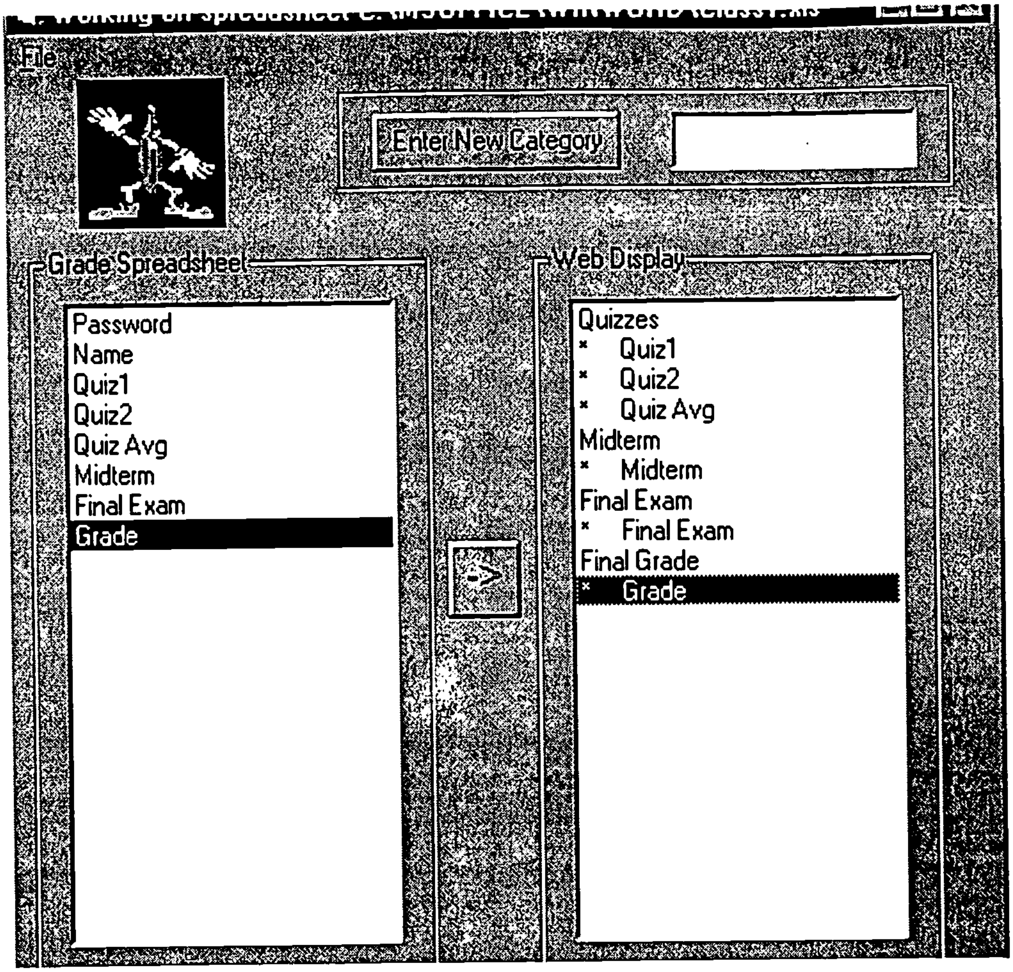
The headings should be in the first row. The first heading in column 1 should be "Password". This acts as the password for the students to access their grades. Any alphanumeric string can be used for the password, although I chose six digit numbers (which are strings!). The names should be entered lastname, firstname with no space after the comma. The last row in the name column should have the name "Average" to be used to show the average grade for each quiz, assignment, etc. Only those columns with an average are displayed on the grade server. The column headings that follow "Name" are of your choosing. The cells of the spreadsheet that contain the information you want displayed should be a named range and the name should be "Grades". The spreadsheet file can be saved anywhere on your hard drive under any file name.

After the template for the spreadsheet is created, the program "gradecfg.exe" is executed. The graphic below shows the program after loading the spreadsheet file.



The spreadsheet file is loaded into the configuration program (as shown above). You can choose which columns you want to display and categorize them. The graphic below shows the window after the choices are made.

BEST COPY AVAILABLE



The configuration file is saved into the directory \httpd\Cgi-Win. The filename should be either grades1.cfg, grades2.cfg, grades3.cfg, or grades4.cfg (there are 4 classes supported per server). A copy of the configuration file is shown below.

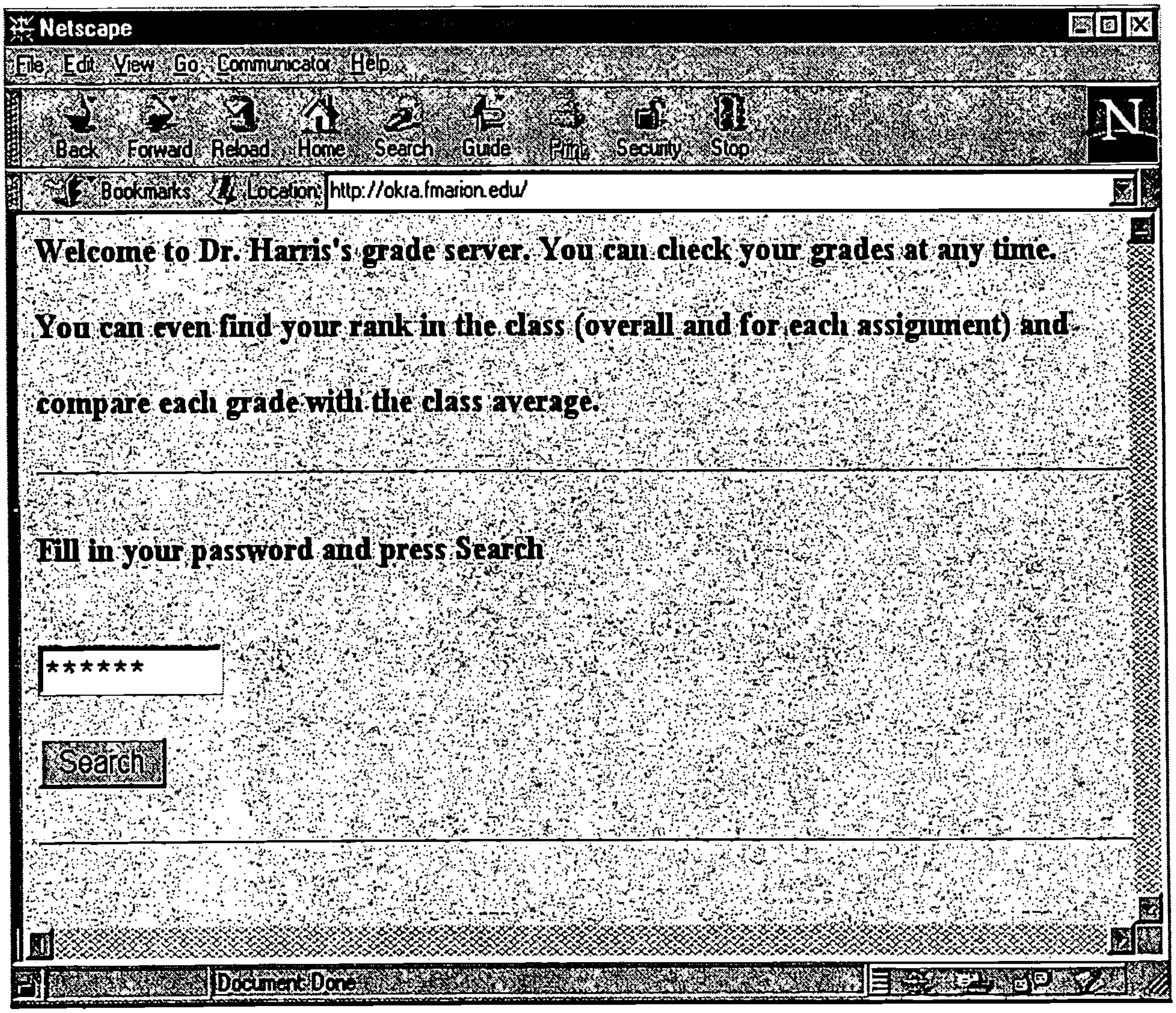
```
"C:\MSOFFICE\WINWORD\class1.xls"
"Excel 5.0;"
"Quizzes"
"*    Quiz1"
"*    Quiz2"
"*    Quiz Avg"
"Midterm"
"*    Midterm"
"Final Exam"
"*    Final Exam"
"Final Grade"
"*    Grade"
```

BEST COPY AVAILABLE

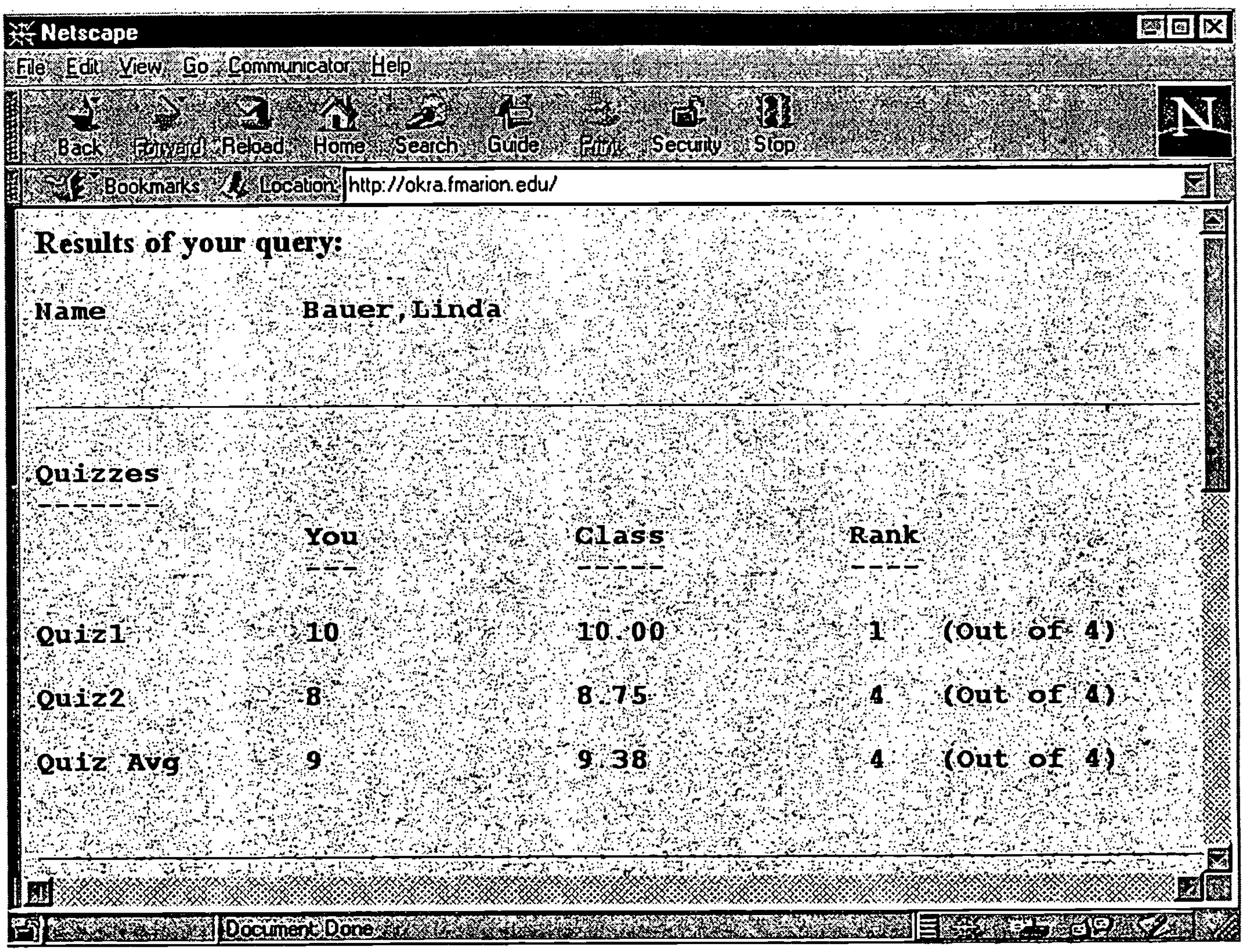
The line "Excel5.0" can be changed in the configuration file to "dBASE III;", "dBASE IV;", "dBASE 5.0;", "Paradox 3.x;", "Paradox 4.x;", "Paradox 5.x;", "FoxPro 2.0;", "FoxPro2.5;", "FoxPro 2.6;", "Excel 3.0;", "Excel 4.0;", "Excel 97;", "HTML Import;", "HTML Export;", "Text;", or "ODBC;" to provide for different formats.

The last step is to copy the CGI program into the \httpd\Cgi-Win directory (either grades1.exe, grades2.exe, grades3.exe or grades4.exe). The program can be executed from any web browser by typing in the URL your-url\cgi-win\grades1.exe (or grades2.exe or grades3.exe or grades4.exe) or a link can be provided (see my web page under "virtual classroom").

When the CGI-win program is executed, a password screen appears:



After the password is entered, it is checked against the password in the spreadsheet file, the grades are read from the spreadsheet and displayed on the following screen.



Implementation:

The CGI (common gateway interface) program was written in Visual Basic 5.0. The public domain program CGI32.BAS needs to be included as a module. CGI32.BAS provides the functions necessary to interface the Visual Basic program with the HTML server. The source code for the main module is explained below.

'** GRADE searchable database
'** Written by Jim Harris
'** The subroutine CGI_Main() is called when the CGI program is called from the browser

```
Sub CGI_Main()
      SendReQuest
      Exit Sub
End Sub


'** The subroutine Inter_Main() is not used, but its existence is necessary

Sub Inter_Main()
End Sub

'** The subroutine SendRequest is called by CGI_Main()
'** SendRequest() outputs the web page asking the user to enter their password
'** It uses the "Send" function provided by the module CGI.BAS to send the HTML out to
'** the browser that ran the CGI program.

Sub SendReQuest()

   Send ("Content-type: text/html")
   Send ("")
   Send ("<HTML><HEAD><TITLE>" & "See your grades!!" & "</TITLE></HEAD>")
   Send ("<BODY BACKGROUND = ""/images/back.gif"">")

   ' ** The SendRequest() function must send a form to the requesting browser
   '** ACTION points back to this program. This program executes after the requester
   '** submits the form

   Send ("<FORM METHOD=""POST"" ACTION=""/cgi-win\grades1.exe"">")
   Send ("<h3>Welcome to Dr. Harris's grade server. You can check your grades at any time.")
   Send ("<h3>You can even find your rank in the class (overall and for each assignment) and")
   Send ("<h3>compare each grade with the class average.</h3>")
   Send ("<HR>")
   Send ("<h3>Fill in your password and press Search</h3>")
   Send ("<br>")
   Send ("<INPUT SIZE=10 TYPE=""PASSWORD"" NAME=""password"">")
   Send ("<br>")
   Send ("<br>")
   Send ("<INPUT TYPE=""submit""")
   Send ("VALUE=""Search"">")
   Send ("")
   Send ("</FORM>")
   Send ("<HR>")
   Send ("</H3>")
   Send ("</BODY></HTML>")
End Sub
```

```
'** SendResults() is executed after the form is submitted back to the CGI program

Sub SendResults()

   Dim Db As Database
   Dim tmpDyna As Recordset      '** A tmpDyna stores the result of a query
   Dim tmpDyna2 As Recordset
   Dim tmpDyna3 As Recordset
   Dim query As String
   Dim SQLQuery As String
   Dim temp As Single
   Dim stemp, stemp1, stemp2 As String

   '**Reading in the configuration file

   Open "grades1.cfg" For Input As #123
   Input #123, gradeFile$
   Input #123, GradeFileType$

   '** Getting the password field
   '** The function GetSmallField returns the value from the form that is submitted
   '** GetSmallField is a function provided by the module CGI32.BAS.

   query = GetSmallField("password")

   '** Sending out the HTML response
   Send ("Content-type: text/html")
   Send ("")
   Send ("<HTML><HEAD><TITLE>" & "Grades Database" & "</TITLE></HEAD>")
   Send ("<BODY BACKGROUND = ""/images/back.gif"">")
   Send ("<H3>")

   '** Opening the database (which is actually an Excel File)
   Set Db = OpenDatabase(gradeFile$, False, True, GradeFileType$)

   '** Doing the query
   SQLQuery = "SELECT * FROM Grades WHERE Password= " & "'" & query & "'"
   Set tmpDyna = Db.OpenRecordset(SQLQuery)

   '** Checking the results of the query
   If tmpDyna.RecordCount = 0 Then     'Can't find password
      Send ("Your password was not found.<br><br>")
   Else
      Send ("Results of your query: <br><br>")
      Send ("<PRE>")
```

```
' ** One of the Password fields in the spreadsheet file must be called Average

SQLQuery = "Select * FROM Grades WHERE Name = 'Average'"
Set tmpDyna2 = Db.OpenRecordset(SQLQuery)

'** Reading in all records in order to calculate the rank

SQLQuery = "SELECT * FROM Grades"
Set tmpDyna3 = Db.OpenRecordset(SQLQuery)
tmpDyna3.MoveLast
s3$ = ""

'** Reading in the fields from the configuration file

Do While Not (EOF(123))      '** Send the results back
 Input #123, s$
 If s$ <> "" Then
  If Left$(s$, 1) = "*" Then    '**    * means an actual field
   s$ = Right$(s$, Len(s$) - 6)
   x = tmpDyna(s$)
   y = tmpDyna2(s$)
   s3$ = ""

  '** Checking the data type of the entry in the spreadsheet cell

  If TypeName(x) = "Null" Then
    s1$ = ""
   ElseIf TypeName(x) = "String" Then
    s1$ = x
   Else
    temp = x

    '** Getting the rank for the number

    rank = 1
    tmpDyna3.MoveFirst
    tmpDyna3.MoveNext

    Rcount = 0
    For i = 1 To tmpDyna3.RecordCount - 3
     z = tmpDyna3(s$)
     If TypeName(z) <> "Null" Then
      Rcount = Rcount + 1
      If z > x Then
       rank = rank + 1
      End If
```

```
      End If
      tmpDyna3.MoveNext
     Next i
     s3$ = Str$(rank)
     S4$ = Str$(Rcount)

     s1$ = Format(temp, "###0")
    End If

    '** Doing the same for the averages

    If TypeName(y) = "Null" Then
      s2$ = ""
     ElseIf TypeName(y) = "String" Then
       s2$ = y
       If s2$ = "Average" Then s2$ = ""
     Else
      temp = y
      s2$ = Format(temp, "###0.00")
     End If
     a$ = "              "

     '** Formatting the output

     If s$ <> "Name" Then
      sendString = s$ + Left$(a$, 15 - Len(s$)) + s1$ + Left$(a$, 15 - Len(s1$)) + s2 +
     Left$(a$, 15 - Len(s2$)) + s3$ + Left$(a$, 3) + "(Out of" + S4$ + ")"
     Else
      sendString = s$ + Left$(a$, 15 - Len(s$)) + s1$ + Left$(a$, 15 - Len(s1$)) + s2 +
      Left$(a$, 15 - Len(s2$)) + s3$ + Left$(a$, 3)
     End If
     If TypeName(y) <> "Null" Then Send (sendString)
    Else
     Send ("<HR>")
     Send (s$)
     s2$ = "----------------------------------------"
     s2$ = Left$(s2$, Len(s$))
     Send (s2$)
     Send ("            You        Class       Rank")
     Send ("            ---        -----       ----")
    End If
   End If
   Send ("<br>")
  Loop
End If
Close #123
```

```
'** Closing out the HTML
Send ("</pre>")
Send ("</H3>")
Send ("<br>")
Send ("</BODY></HTML>")
Db.Close

End Sub
```

Conclusions:

I have used this grade server for one semester and the students like it. They have access to their grades as soon as they are entered into the spreadsheet. This also removes the need for posting grades after final examinations. The source code and executables are available at the URL http://okra.fmarion.edu under okraboy software.

References

1. "Visual Basic 4 Unleashed", Conrad Scott, Brad Shannon, Frank Font, and Bill Hatfield, Sams Publishing ,1995.

2. "Microsoft Visual Basic Web Database Interactive Course", Gunnit S. Khurana and Satwant S. Gadhok, Waite Group Press, 1997.

The Power of Email over your WEB Browser

Timothy Hibbard
Hocking College
3301 Hocking Parkway
Nelsonville, Ohio 45764

Is your college looking for an E-mail system that is user-friendly, cost effective, and has the capability of adding new students and deleting existing students quarterly - a system which is easy to install, easy to administer, and highly accessible? If you answered yes to the above questions, then I have the perfect E-mail package for your college.

NELIE is a fully capable E-mail system that runs on most web browsers and has many advantages over most E-mail packages being sold on the market today. Many E-mail packages on today's market (Lotus Notes, CCmail, Netscape, Internet Explorer, and Group Wise) require you to install and configure the software at each workstation. This can be very time consuming if you have 1,000 computers or more on your local campus. With NELIE, all you have to do is to install the E-mail package on one server, then have the user access a web site. No client side configuration is needed.

NELIE's capabilities include:

- Composing E-mail
- Reading E-mail
- Forwarding E-mail
- Replying to E-mail
- Carbon copies
- Blind carbon copies
- An address book
- Sending attachments
- Receiving attachments
- Deleting messages
- Creating additional mail folders.

Additional features include the ability to find other email addresses on the local system. The users have the ability to change their password, E-mail address, and can choose where their email will be delivered.

With NELIE, administration of the system is easy. Feed your student enrollment to NELIE and it will decide what to do with each student. If the user is new to the system, then NELIE will generate a home directory, a mail directory, and an encrypted password for that user. If the user is no longer enrolled the software package will remove all files and passwords associated with that student. NELIE will create a backup file to be available if the user re-enrolls. This program can add users at a rate of 100 users per minute!

Unlike most systems, NELIE makes it easy for first time users to access their E-mail account. The user only needs to know his or her student identification number and the last four digits of his or her social security number. With this information NELIE will assign the student with a login name, a password and an E-mail address. Once the user is in the system, he or she has the option to customize his/her password. The user can also create aliases to his or her E-mail addresses. The default user E-mail address looks like: xx123456@your.college.edu, but instead the user can customize their addresses to be: john_doe@your.college.edu.

Many E-mail systems do not support sending or receiving attachments and require a client side program. NELIE has a built in MIME encryption/decryption program that can encode and decode MIME types. This feature makes sending and receiving attachments very easy.

Systems such as Internet Explorer, Netscape, CCmail, Lotus Notes, and Group Wise download the users E-mail to the hard drive. This allows the next user at the workstation to read the previous users E-mail. However, with NELIE, all the messages are kept on the server at all times, and nothing is ever downloaded to the hard drive. This feature allows the user to go to any workstation in the world and check their
E-mail.

Most importantly, installing NELIE is easy. All you need is a UNIX based system capable of acting as a mail server. My college uses Caldera Linux because of the stability of the operating system and its cost ($200 educational). Run the installation program and answer a total of 15 questions and NELIE will be installed! Click another button and you will have all your users added. NELIE is a one-time purchase, which means you do not need to buy licenses for additional users. For example: If your college has 1,000 students and you increase your enrollment to 2,000 then NELIE will automatically add the new users. The purchase of NELIE entitles the buyer to any new features released in the future.

Using Technology to Communicate, Cooperate and Collaborate

David G. Holland
Executive Director, Information Systems
Mott Community College
1401 E. Court St.
Flint, MI 48503
(810) 762-0210
dholland@email.mcc.edu

Communication - an exchange of information or ideas. When you and I exchange ideas or information we are communicating. We can tell each other our goals, objectives and how we expect to accomplish them.

Cooperate - to act jointly with another or others. I cooperate with you when we act jointly. For example, we may need to re-boot a server to fix a problem. I ask you to sign-off so we can reboot. You sign-off, I re-boot and we've cooperated. I can cooperate with you in simple ways. This allows you to reach your goal. However I may or may not assist you in reaching your goal. In this first example, you cooperated, but you didn't really help much. In another example we may agree to cooperate on an assignment. I will write Chapters 1-4 of a project and you will write Chapters 5-8. Together we will cooperate to accomplish this task.

Collaborate - to work jointly with others. To collaborate we must work jointly to accomplish something. We work together to accomplish something together. We think of this as a team effort. For example, we may decide to collaborate on writing chapters 1-8 of a book. Together we could brainstorm the ideas for each chapter. Together we could write an outline of each chapter. Individu- ally we might write the chapters and then we could edit each other's work to finalize the project.

One goal at Mott Community College is to work together with the community. Over the past several years the college has had many examples of successful projects. However the projects appeared as "smoke-stacks". Each was done well in and on it's own. However, there was not a global strategy for which all projects were done. Additionally there was little collaboration among departments at the college. Each department appeared to be doing everything by themselves. Rarely was a cross-functional team working together.

Part of the problem was clearly communication. Mott Community College is the largest college in Genesee County (Flint, MI). We have approximately 10,000 to 12,000 students enrolled in during any given year. We have a main campus and two satellite campuses (Southern Lakes and Lapeer). The 500 employees of Mott Community College relied mainly on standard communication methods: Letter, Memo, Fax, Phone and Voice Mail. These tools, while effective in small groups were not enough to allow the employees at Mott Community College communicate effectively.

In 1996 I held a discussion with the Vice President at the college. We discussed the goal of building collaborate processes at the college. Information Systems was identified as a key player in this process. How could we use Technology to help the college achieve its goals. During this discussion the process of communication and cooperation were identified as necessary. It was clear that although our goal would be collaboration, our first step would need to be communication. After building a system of effective communication we would look to build systems to help staff cooperate. Finally, we would work to build systems that would encourage collaboration among the staff or Mott Community College and our community.

COMMUNICATION

Our first goal was communication. How could we use technology to allow people to communicate? How could information be made available to many?

Person to Person Communication: email

Certainly email was the first system we implemented. However, the key to this communication is the effectiveness of it. We set the following objectives:

Email from any platform (PC, MAC, UNIX, POP3)
Email from any location (office, home, labs, hotel, outside location)
Email easy to use, training available
Student email

If you expect email to be a major means of communication it must be accessible, easy to use and available to everyone. Many systems are available which meet these needs. We chose Lotus Domino on the server. We allow access to email via Lotus Notes for PC or MAC, any POP3 client (Netscape, Internet Explorer, etc.), and UNIX client access. We allow access to email from any workstation on campus, dial-up from home, via the Internet, or via FTP. We are looking at a package to allow users to read their email via their voice mailbox. Our goal is to allow email to be used easily by anyone anywhere.

Calendaring:

Next to email we felt a huge improvement was to allow people to do calendaring on the system. This allowed staff to keep their calendar on the system. This allows staff to allow others to read their calendar, update their calendar and even manage their calendar (based on secure parameters). A common calendaring system allows people to schedule together. If you need four people for a one-hour meeting you can query the system regarding available times. This allows people to schedule time to work together more efficiently. Many systems are available on the market that will meet these needs effectively at a low cost.

Intranet/Internet:

Information is available to people in a variety of formats. A few years ago a bulletin board was key to the college communication. Today, no bulletin boards are big enough to hold all the information available. We have built an Intranet to keep internal information. This system holds a variety of information from all areas. We have built systems to allow the owners of the information to post new information directly to the web without impacting Information Systems. This process is key to keeping information new and up-to-date.

Our Internet web site is in use every day by faculty, staff, students and guests. Our web-site has become one of our most important means of communication. Anything that is prepared in print (newsletters, brochures, etc.), is also posted to our web-site.

Campus Directory:

The campus directory (available via the web) contains important about contacting anyone at the college. This information is available by person, department or service. We provide information regarding where on campus anyone or anything is located, phone number to contact people or departments and information about each service available on campus. This information is available through a variety of views (sorts). In addition, a section of FAQ's (Frequently Asked Questions) is available to help resolve issues.

Open Positions Database:

The Human Resources Department always has important information for people across campus. All open positions are now managed via a database (system) and automatically posted via Intranet/Internet (depending on how the position is categorized). This system allows staff to monitor opportunities for advancement from their desktop.

Human Resources Information System:

The Human Resources Information System allows staff the HR department to communicate information to staff. This system is a completely on-line system that can be updated by the Human Resources Department at any time. The system holds data regarding: Job Descriptions, Union Contracts, Benefits, and HR FAQ's. Statistics show this system is accessed regularly by staff via the Intranet.

Web-based Course Schedule:

Based on information previously only available in the course schedule (and often quickly outdated) we developed a web-based course schedule. This system allows staff and students to access up-to-date information about our class schedule. Two types of access methods are allowed. The Internet access allows students to query information regarding courses based on a variety of topics including: type of class, semester, discipline or course name. The system provide accurate up-to-date response which accounts for sections which have been closed, new sections opened and even sections which are full.

An Intranet version of the Web-base Course Schedule is available to internal use. This system provides up-to-date information about sections including seat counts and historical information. This system is often used by associate deans for making decisions about classes and often used by counselors and advisors when working with students.

Purchasing Procedures On-line Documentation:

One of the most important types of communication is documentation. We have rolled-out several new systems in the past two years. One of the standards for all new systems has been on-line documentation. The purchasing procedures system is an excellent example of system documentation. It provides documentation of both the system and the procedures necessary for purchasing. A user need only look to this system to gain information about all the steps necessary (on and off the computer) for making purchases.

Press Release Database:

A year ago faculty and staff was occasionally surprised about what was in the local papers about the college. A need for communication this information was quickly identified. We produced a press release database that allows staff to access all press releases. It is tied into email to allow important press releases to be immediately email to everyone on campus.

Reports System:

Last year Mott Community College delivered reports to the State of Michigan that was inaccurate. When researching the reason why it was apparent that the reports had not been seen by the right people on campus. Since most of our Federal and State reports are available to the public, they certainly should be easily accessible by faculty and staff. A system has been developed which allows faculty and staff immediate access to all required reports. In addition a list of appropriate report reviewers has been identified for each report. The system is tied to email so that all report reviewers receive the report via email at least two weeks before it is sent out to the requiring agency.

COOPERATION

Our second major task was to build systems of cooperation. We looked first to automating existing process. Our goal was to develop workflow automation systems that would allow the work to flow more efficiently between people. By eliminating "wait time" in the steps of a process we could improve the processes for everyone.

Current Events System:

It is always tough to communicate to people about what's happening on campus. We have developed a current events system that tracks and manages all the events on campus. However the system has multiple views available for accessing the data. These views allow people to see the events that they're looking for. For example there is a sporting events view which only shows sporting events. Several other views are available.

Help Desk Service Request Database:

This system has been developed to allow staff and Information Systems to cooperate to resolve Help Desk problems. The system allows any staff to input a service request into the system. The Service Request is automatically routed to an appropriate queue for assignment. From the queue the Service

Request is routed and tracked from assignment through completion. As the status of the Service Request changes it is automatically emailed to the originator. This important flow of communication allows staff and Information Systems to cooperate together and resolve problems quickly.

Facilities Service Request System:

The Facilities Service Request System is modeled from the Help Desk Service Request Database. It allows anyone on campus to originate a Facilities Service Requests. Based on the type of request it is automatically forwarded to an appropriate queue for resolution. Again, the originator is kept up-to-date as status changes occur. However, the system has an added feature which is helpful: the system provides automatic escalation. If a problem is not resolved in a reasonable amount of time (based on priority), it is escalated (via email) to supervisors, managers or directors for follow-up. This important feature has allowed the department to respond to important problems that previously might have been "forgotten" or "misplaced".

Book Request Database:

One problem on campus was getting the right books on the shelves in the bookstore on a time. This system automates the book request and ordering process from faculty to the Associate Dean to the bookstore. Faculty enters their book request into the system for each section to be taught in the upcoming semester. The system provides important reports to associate deans for follow-up. The bookstore receives the request, can combine them into groups and make sure the right numbers of books are ordered.

Room Scheduling System:

In 1996 several staff were available the first week of class to resolve classroom-scheduling problems. If two classes arrive at one classroom at the same time, they would work to find another classroom for one of the classes. A room scheduling system allows administration to schedule rooms with confidence that they are available. An interface with our administration system was necessary to allow us to insure it is done completely.

COLLABORATION

Our third step is to build collaborative systems which will allow teams of people to work together to accomplish common goals. These systems are built upon similar themes to the cooperation systems. However they allow people to work closely together to solve problems.

On-line Purchase Requisition & Approval:

The system allows people to work together to get appropriate items requested, approved, ordered and delivered in a timely manner. This system works a request from the requestor to the approver to purchase to shipping and receiving. It automates the process for checking fund availability. The system is tied to our email system and routes purchase requests automatically to the right people.

Grievance Tracking/Solution System:

Mott Community College has five different classifications of staff with five different unions. The college wants to resolve any issues in a reasonable timeframe. However, the rules and regulations regarding these issues are complex. These system tracks outstanding grievance issues through the union approved process. Many of the steps in this system are timed. If certain steps are not done within the timeframe, warning emails are automatically sent to appropriate staff. This system also help the union staff to monitor the process and steps involved in resolving a grievance. Before this system was developed the college had close to 100 unresolved grievances (many old). As of this writing there were 8 open grievances!

Faculty Information Management System:

Important information about faculty is necessary for many departments: Accounting, Payroll, Human Resources, Division Offices and Faculty themselves. We needed a system to track:

Faculty Assignment/Contract
Automate Payroll based on assignment
Automate GL allocation based on assignment
Calculate Seniority, preference points and earned benefits based on assignment.

Keeping track of all this information was difficult. It would be tracked manually by one person effectively for a period of time. Then the person would leave or transfer and the college would lose track of key information. This problem occurred many times. The college needed a system that would include all the groups necessary allow everyone to access the same information, allow proper people to enter assignment information, produce the faculty contracts (full-time, adjunct and part-time) and provide reports. By putting together a cross-functional team we were able to develop a system to meet the needs of many groups at the college.

All of the above systems were developed in cooperation with user groups. All of these systems were built in less than 2 years by a staff of 3 people (one working part-time). The Information Systems Department used an inclusive process for developing these systems with the users. At ASCUE '98 I will be discussing some of the processes used in developing these systems. In addition, in June 1998 a presentation will be place at the Mott Community College website (www.mcc.edu) regarding how Mott Community College is Using Technology to Communicate, Cooperate and Collaborate.

The Information Technology Initiative at Grove City College: Four Years Later

Frederick J. Jenny, Ed.D.
Chairman, Computer Systems Department &
Associate Professor of Computer Systems
Grove City College
100 Campus Drive
Grove City, PA 16127-2104
(724) 458-2071 (voice)
(724) 458-2329 (fax)
email: fjjenny@gcc.edu

Introduction

On August 30, 1997, five hundred fifty-eight Compaq Armada 1510 notebook computers and Canon color bubble-jet printers were distributed to incoming freshmen, members of the class of 2001. With that event Grove City College (GCC) had reached an important objective: the Information Technology Initiative at GCC fully involved all 2200 students.

Progress of the GCC Information Technology Initiative (ITI) was presented at the 1994 and 1995 ASCUE Conferences prior to and after the first year of the Initiative. This report summarizes a perspective of the Initiative after four years.

Review of the plan

The objectives of the Initiative are as follows:

1. To prepare students for excellence in their chosen profession by providing the necessary technological tools and instruction

2. To recognize, in a tangible manner, the fact that state-of-the-art computing is moving away from centralized computing and rapidly toward distributed computing facilities following the concept of Anytime/Anywhere Learning.

3. To increase the technological resources available to students on campus while reducing the total cost of technology ownership to the College

4. to centralize all support activities including the computer help desk, training and repair shop staff.[1]

Laptops were chosen for a number of reasons. Students need computers in a variety of environments: science labs, dorm rooms, study rooms, the classroom, and a faculty advisor's office. Providing desktop computing labs able to meet these kinds of needs is difficult and costly in terms of hardware and software upgrades, support and maintenance. Lack of campus-wide standards in desktop systems compounds the problems. Portable systems alleviate many of these concerns, or at least lessen the difficulty in addressing them.

The College implemented a one-computer-per-student model, creating an Anytime/Anywhere Learning[2] environment. Systems have been distributed to students and faculty. Motivation for this strategy includes the following benefits:

1. Laptop computers can easily be transported to a classroom or laboratory to be used in place of a traditional desktop computer.
2. Laptop computers are the only true mechanism that can be used to create an Anytime/Anywhere Learning environment.
3. Students have the convenient and continual access to technology. In fact, students are allowed to take the computer system home with then during breaks.
4. New technology is introduced each year through the purchase of equipment for freshmen students.
5. The number of computer labs on campus has been greatly reduced.
6. Students and faculty are required to bring defective equipment to a central repair shop, thus eliminating the need for technicians to travel to an office or dorm room.[3]

The infrastructure necessary to support this endeavor features the Computer Help Desk and the Computer Repair Shop. The Help Desk (housed in the Technological Learning Center) is open from 8:00 am until midnight on weekdays and staffed by shifts of 35 students. The Help Desk has reduced hours on weekends, as do the library and computer center. A full-time manager oversees the operation of the Help Desk, and is responsible for the scheduling and training of student workers. Workers are trained in the hardware, software (*MS Windows, Works, Mathematica*), and networking systems. The Repair Shop is staffed by a full-time secretary and a full-time repair technician who is Compaq Self-Maintainer certified.

The laptop computer initiative has been well received and has been the focus of much attention both on and off campus. The entire GCC community has positively embraced the initiative and the benefits are worth the expended energy and cost of the necessary work and planning.

Update of the Computer Initiative Systems

Since the last report of this initiative, significant changes have occurred on campus. The following table summarizes the evolution of the hardware over the past four years, and includes a tentative configuration for the system for freshmen arriving in the fall. Please note the dramatic change in computer power represented by the various systems over the five years.

1998-1999??	1997-1998	1996-1997	1995-1996	1994-1995
200 MHz processor Dual-scan color screen 32 MB RAM 2.0 GB hard drive 33.6 baud modem 10/100 PCMCIA Ethernet card 20x internal CD-ROM w/ multi-media Canon BJ-240L color bubble-jet printer *MS Windows 95* *MS Office Pro 97*	120 MHz processor 16 MB RAM 1.0 GB hard drive 33.6 baud modem 10base-T network card 10x internal CD-ROM Multi-media Canon BJ-240L color bubble-jet printer *MS Windows 95* *MS Works 4.0*	100 MHz processor 8 MB RAM 800 MB hard drive 33.6 baud modem Canon BJ-240L color bubble-jet printer *MS Windows 95* *MS Works 4.0*	50 MHz processor 8 MB RAM 350 MB hard drive PCMCIA slots 19.2 baud modem Canon BJ-240L color bubble-jet printer *MS Windows 95* *MS Works 4.0*	25 MHz processor 4 MB RAM 120 MB hard drive 1200 baud modem HP 500 Inkjet or Canon BJ-240L color bubble-jet printer *MS Windows 3.1.* *MS Works 3.0*

Some students are disappointed to keep the same computer during their four-year tenure at the College. To address some of these concerns, students have been able to add memory, upgrade hard drives, or add external peripherals. However, the College does not participate in a computer refresher program; students cannot upgrade to a newer computer model. The added cost of a computer refresher program (and the low residual value of the older machines) would have an undesired effect on GCC's tuition, now approximately $11,000 annually for tuition, room and board, and computer.

As one might guess, faculty members tend to be more vocal in expressing their dissatisfaction with the limited processing power of the first year 25MHz systems that they were distributed. While the College has been unable to replace all faculty systems on a whole scale basis, normal budgeting processes have enabled some departments to upgrade laptop systems sooner than others. Forty-six of the 50 MHz systems with memory upgrades have been distributed to faculty, most of them going to faculty in the humanities. Other faculty systems have been upgraded with larger hard drives, additional memory, and external CD-ROM drives. With respect to software, most faculty have migrated to *Windows 95* and *Microsoft Office*. This may present compatibility difficulties with student files created in *MS Works*.

Facilities

At the request of the President of the College, a Campus Technology Task Force was formed in the Fall of 1996. The charge of this committee was to contribute to the plan to continue the momentum of the Information Technology Initiative, and to "critically review the effectiveness of the program to date and develop mechanisms to rectify the shortcomings of the initial effort."[4] Many of the changes and proposed changes in the facilities that related to the ITI are a result of the recommendations of the committee report.

The campus network has expanded significantly since the last report to ASCUE. At that time, access to the network resources was available only by dialup for the majority of faculty and students. The campus backbone to the dormitories has now been completed. The infrastructure is Full-Duplex Asynchronous Transfer Mode (ATM) @ 155 megabaud, yielding 310 megabaud capacity per building connection. The network is switched on-campus and routed to the Internet. ATM uplinks feed Fast Ethernet (100 Mbps) which feed Ethernet (10 Mbps) switches.

All faculty offices have been provided with 10base-T network connections except for the Physical Learning Center. The science building, engineering building, and computer center are fully connected, as are administrative offices. Plans for the main classroom building continue, with the goal of being connected in classrooms by fall. Three relatively large lecture halls are currently capable of supporting multimedia lectures and demonstrations.

The freshman (510) were provided with network adapters as part of the ITI. Upperclassmen were given the option of purchasing network adapters and 256 have thus far. The total represents 38% of the resident population (2025). Few seniors elected to purchases parallel port network adapters. Training sessions are required before the network adapters are distributed and the systems are properly configured.

The main public access lab in the computer center has decreased in the number of supported stations and platforms over the last several years, but the lab has not disappeared. The College still has a Novell Intranetware LAN of desktops offering software not licensed broadly (*PageMaker, FrontPage, Office*, and discipline specific applications.) A significant number of upperclassmen choose to use the public access lab for Internet access for email and the Web, even though they may elect to connect to the campus network from the dorm if they have purchased an adapter. "Walk-up" stations have been installed in the computer center allowing commuting students to connect to the network at any of the stations.

Staffing

The staff to support the ITI has grown. Personnel include two network specialists, two repair specialists, secretary, ITI director, and recently added was a software specialist/trainer. The software specialist/trainer manages the Help Desk and has delivered a significant amount of training to faculty and administrative users. STARS (Student Technology AdvisoRS), based upon the model developed at Wake Forest University, is currently under development. This program provides procedures for faculty to utilize student help for technology-related projects.

Repair Shop

As mentioned previously, the Repair Shop has two full-time repair technicians who are Compaq certified. They are also certified to repair Canon printers, and will shortly be certified for Hewlett-Packard printers as well.

Except for the two-week period after the start of each semester, the shop has been able to maintain a 24-hour turn-around repair rate. During the busy start of each semester, the turn-around time averages three days.

Curriculum

The use of Internet and intranet services for class-related activities has grown on campus as the number of campus constituents on the network has increased. The integration of technology into courses various from department to department, from professor to professor. The College has been supportive of faculty wishing to increase their knowledge and comfort level of technology integration. Many challenges still lie ahead in this area. In addition to increasing the amount and availability of training, the Campus Technology Task Force has also recommended the hiring of an instructional technologist to support faculty.

Within the service courses in the Computer Systems Department(Introduction to Microcomputers and Applications, Technologies of Instruction, for example), the annual upgrade of freshmen machines has created an interesting logistical problem. These courses introduce *MS Windows* and *MS Works* as units of study; the senior class has *Windows 3.1* and *Works 3.0*; underclassmen have *Windows 95* and *Works 4.0*. Teaching hands-on lessons with students with different versions of system and application software has been challenging. This problem will surface again as the College migrates to *Office 97* for students.

The notebook computers, however, have been particularly beneficial within other Computer Systems Department courses, particularly course involving programming. Students are able to bring their notebook computers (and compilers) to class and write, test, or debug code segments illustrating new concepts. Students are able to bring their systems to faculty offices, working through a problem in the debugger on the student system. Students of other courses such as Systems Analysis or MIS are able to use their own systems for presentations with PowerPoint. The benefit of Anytime/Anywhere Learning is in full force.

Summary

The Information Technology Initiative has required a tremendous commitment of resources in terms of time, planning, personnel, and work, but the benefits for the College community have also been tremendous. The Anytime/Anywhere Learning strategy prepares GCC students for lifelong learning in the information age.

[1]Inman, John G. Available on-line: http://www.microsoft.com/education/hed/news/January/itofage.htm.
[2]Available on-line: www.microsoft.com/education/k12/aal/.
[3]Inman, John G. Available on-line: http://www.microsoft.com/education/hed/news/January/itofage.htm.
[4]Goncz, Joseph G. Report and Recommendations, Grove City College Campus Technology Task Force, March 1997.

Using Intelligent Agents to Assist Educators

Dr. Steve Knode
Information Resources Management College (IRMC)
Ft. McNair, Washington, D.C. 20319
202 685-2070
knode@ndu.edu

Professor Jon-David W. Knode
Bladen Community College
Dublin, N.C. 28332
910 862-2164 x293
jdknode@hotmail.com

Introduction

Now that information is both abundant, practically universally available, and inexpensive, what is the result of losing the "information float" time that has traditionally been available to educators? What happens when educators lose the time to read new information, digest new information, incorporate new information into lessons, and develop a new finished piece to present to students? Likewise, from the students point-of-view what happens when the amount of information that they are exposed to is escalating so rapidly they are overwhelmed?

-What happens to the teaching and learning processes when each of us has access to the databases of the world, when virtually (and perhaps literally) any piece of data on any subject is "out there" for the getting? Will we find this abundance of data availability truly a boon to learning?

-What happens when the Internet is available to everyone and "data" in almost any form is practically free for the asking and flows nonstop from any point to any point? Will educators and students be able to get the necessary and appropriate inputs for learning in a timely manner?

-What is our reaction when on-line services can get for us data on any topic at any time for minimal cost; when we can shop, explore, discover, and create new information on the fly? Will we be delighted that all of the information in the universe is now at our disposal? Or, will we soon find that we are overwhelmed with the quantity of information available, and that asking for data on a subject is likely to result in a "data dump" of unprecedented proportions?

-What if we are not able to sort, filter, collect and analyze the needed data in a timely manner? What if the amount of data (much of it unneeded and unwanted) buries us in an avalanche of minutia? What then? How will we deal with the age of "information superabundance"? What then?

Today educators, as much as anyone, are faced with the dilemma of dealing with what has become an information overload. Faced with this era of data abundance, what educators and students need are helpers or some sort of "intelligent assistance". However, it is might be possible to survive (perhaps even prosper) in the age of information hyperabundance with some help from "intelligent agents" (IA's). Already even these first generation IA's have to a degree developed the capability to find needed data in a timely manner; to sort through data and rearrange it into a more useable format; to know what is wanted and not wanted; to learn over time what is and isn't important to the user; to recognize when something relevant has arrived and inform those who need to know; to be vigilant and watch for needed information and deliver it just in time; even to coach students and to answer some of their questions.

The list of tasks that intelligent agents might someday (now?) perform is rather extensive. For example, IA's could navigate, browse, retrieve, sort, organize, filter, summarize, and store data. Further, IA's could monitor and alert when data change or arrive. They could schedule appointments and remind when the appointment arrives. They could guide, advise, teach, critique and explain in an intelligent tutor role. They could even solicit information and distribute information.

Background

Intelligent agents (a.k.a. software agents and/or knowbots) are, in simplest terms, software programs built to perform certain specific tasks for the user. According to Alan Kay, IA's had their origin in the late 1950's and have been on the rise over the last ten years.[1] Indeed, in their crudest form, they are simply macros or programs which automate steps in a software package. The earliest form of intelligent agents (although they weren't called such) were macros that the user developed for his or her own use. In programs such as Lotus 1-2-3, Excel, or Word Perfect, users could "program" steps or automate key strokes so that these steps would occur repeatedly when given the execution command (usually a certain key stroke). Certainly these early macros were not really "intelligent" in any sense of the word, but they did automate previously repetitive manual tasks, and appeared to be doing something clever when executed.

Later generations of these macro commands have appeared in the form of intelligent tutors or assistants. Again, many programs, such as the Microsoft Office products now have intelligent tutors (wizards) embedded in their systems to assist the user in performing tasks or in finding help. Just as the earlier macros automated keystrokes, these intelligent tutors or advisors in some cases automate sequences of keystrokes at a higher level. For example, in Excel, the "wizard" will actually build a graph for you. Simply define the range of data, and the wizard looks at the data, decides which of the various graph types is appropriate, and then develops the graph with very minimal input from the user. Not always is the wizard correct, but very often the correct type and form of graph is developed. This represents a distinct improvement from having to study how to build graphs, then actually perform the keystrokes to produce the graph. In reality, these improved versions of the earlier macros haven't any real intelligence either, but appear to be doing some intelligent work. Their contribution to managing data is sizeable, even though they are very crude in the programming behind them.

More recently, "intelligent agents" have begun to appear which represent a significant advance in capability. This new generation of software agents has a significantly improved ability to assist in the data collection and data management functions.. Combining elements from other fields (e.g., expert systems, neural networks, object-oriented technology, case-based reasoning, etc.) the latest generation of software agents moves closer to the ideal agent--one which can function more autonomously to assist the user. These latest incarnations of intelligent agents can retrieve information from sources, find information when directed to do so, alert when information has arrived, shop for the best prices for items, even learn what kind of information is needed. These agents can do everything from automatic routing of e-mail to gathering of stock quotations to building customized newspapers. Furthermore, this latest generation represents only the beginning. Future editions of intelligent agent software promise to become still more autonomous in their operation, perhaps even learning from the user.

Intelligent Agent Taxonomy

Intelligent agent software can be categorized in various ways, for example, by the degree of "intelligence" embedded in the software, by the mission/task performed, by the technologies used, etc. Therefore, a simple classification scheme is not possible. However, one useful means of classification is by the amount of "intelligence" contained in the software. In this type of categorization, there are four main types. The simplest type is that of end-user programmable (or "simple") agents.[2] These agents rely on end-user programming to carry out relatively simple operations. This is usually done by some sort of macro-like programming or scripting language. An example of this type of agent is the software package NewWave Agents from Hewlett-Packard.[3] Using a BASIC-like language, users can program the software to automate certain functions. Edify's Workforce is another automation tool for developing user specific intelligent agents by writing code--in this case dragging modules rather than actually writing programs.[4] Simple agent software can be very effective in automating repetitive tasks (e.g, voice-telephony, fax, database queries, etc.), but they are actually nothing more than fancy macros. Many of these packages require a rather large investment in the learning curve involved to become proficient in using them which limits their utility for most of us.

Further along the taxonomy of intelligent agents are those packages which are termed KBS (knowledge-based systems) agents.[5] KBS agents are "smart" because they obtain all or part of their functionality from knowledge-based technology. They may rely on rules or other expert systems techniques to function effectively in assisting the end-user. More expensive and difficult to build than simple agents, these KBS agents add functionality by being more flexible in accomplishing tasks. Instead of having to be programmed for each instance, they can rely on generalities to accomplish similar tasks. However, once developed their functioning is very rigid, requiring changing of the internal logic to adjust to new situations. Magic Cap is an example of a KBS agent software package. In Magic Cap rules can be specified which will govern how the computer deals with e-mail or files documents, etc.[6] Other software packages in this genre are Telescript Agents, and PersonaLink Agents (both of which are distributed or network packages). Likewise, internet intelligent agent software, such as Pointcast, allows the user to customize retrieval of information so that the information is tailored to what is needed or wanted.

The most advanced of the current types of intelligent agent software are those packages which can create self-learning agents. These packages learn to automate functions by observing the behavior of the user, through examples, observations, and feedback over a period of time.[7] The agent eventually "learns" what the end-user does and offers to automate these situations. For example, the "agent" would observe that e-mail with administrative message content is often deleted and suggest that function be automated. Examples of this type of agent software are NewT and Open Sesame! software packages.

Finally, just in the formative stages are software agents which collaborate. These applications would allow agents to interact with each other and learn or cooperate with each other. Much of the research today is in the area of collaborative agents or multiple agents.[8] Many problems exist which must be solved before these types of agents will emerge fully, however, such as the communications problem, the loss of control, the impact of malevolent agents (e.g, viruses) on workflow, the need for standard "languages", etc.

Applications

Today intelligent agents are performing several important tasks rather successfully. Although most of these applications are first generation tools and techniques, they are already having a significant positive impact on organizations. Most of them assist users in dealing with the vast information available on the internet.

(NOTE: The web addresses for these agents are listed at the end of the paper.)

Agents that Watch - Watcher agents operate autonomously, looking for specific information or events. When information relevant to the user is found, the agent can either notify the user directly (e.g. with a news flash or email) or store the information for future access (e.g., for a personalized newspaper). For an educator, having instant notification that information in a particular subject has just become available can be invaluable. These types of agents can help insure that educators are aware of the most current information and have access to it.

Pointcast and **fishwrap** are examples of this type of agent. They are basically sophisticated search engines coupled with formatted delivery items. By tailoring these agents, you can receive information meeting your interests at convenient times. Pointcast, for example, serves the information in the form of a realtime screen saver.

Remind U-Mail will send you an email when an important event is coming up. This is an agent whose obvious use would be to remind educators and/or students about key dates and suspenses.

NetMinder is another watcher agents, in this case watching for changes in key web sites and then emailing you with the news that a change has occurred. If you are using information from web sites in a course, it is helpful to know when that web site has changed.

Amazon.com, the world's largest on-line bookstore, has agents that will watch for books on a particular subject and notify you (via email) when one is found .

Webcatcher is a free e-mail service which sends subscribers lists of new URL's which are relevant to their interests chosen from a list of about 100 topics.

Agents that can Learn - This type of agent supposedly "learns" from observing your behavior. A learning agent is capable of tailoring its performance to an individual's preferences by learning from a user's past behavior. For an educator with limited time to visit the many internet sources of information and sort through that information, these agents can be most useful. Soon hypertext guides will be able to lead you through documents, only stopping on the passages which are relevant to the purpose. Intelligent agents can assess your familiarity with the information and decide which parts you need to see. They can, to a degree, learn what is of value to you and assist you in finding that information.

An example is a Windows application that provides access to Usenet newsgroups, **smart newsreader**. It lets you grade read articles and conversations and then makes sure that the more interesting ones stay near the top of the list. As newsgroup articles are read, users can express preferences which can be used to rank new articles and threads according to interest. The more articles read and ranked, the better the agent learns what is relevant to the user.

Likewise, Intel's **selection recognition agent** dynamically generates hyperlinks between information on your desktop and relevant web sites and applications.

Lifestyle finder from Andersen consulting attempts to match you up with internet sites which would be of interest to you.

Agents that can Retrieve - These agents are capable of searching for information in an intelligent fashion. The most obvious example is an Internet search agent which can conduct complex searches by interpreting the search criteria defined by a user. Being able to at least partially automate the search process (beyond just using basic search engines) is extremely useful and time saving for educators. Further, IA's can even create personalized newspapers, containing only information desired.

Autonomy agent is a good example of a type of information retrieval agent. Through simple text input and using neural network technology, autonomy agents train themselves to retrieve information of relevance to the user. The more the agents are trained, the smarter they become in finding the correct information. Autonomy agent can also be used to create a custom newspaper from over hundreds of newspapers across the country.

CNN++ is an intelligent newsfilter which can retrieve information based on keyword searches.

NEWSpot is a free e-mail service that delivers custom news summaries from Mercury Mail. Users can choose from more than 20 categories and get delivery several times per day.

In like manner, **Robobear** can scan Usenet newsgroups for multimedia (pictures, sounds, videos) based upon criteria selected by the user.

WebWhacker, is an "off line" web browser for Windows95 and NT. Users can preselect the Web sites that they want to monitor and set a schedule for Web pages, text, graphics and HTML links to be delivered to their desktop computers automatically and unattended.

AdHound searches the classified ads of hundreds of newspapers and will email you when it finds what you are looking for (Note: **Job Hound** looks for job opportunities for you in the databases of Morgan & Banks!).

Agents that can Assist in specific tasks - Assist agents perform specific tasks to assist users. For example, an agent could determine that the reason a user was unable to receive e-mail was because his mailbox was full and generate a message to that effect. Based on some expert system rules, the agent might even be able to diagnose and fix the problem automatically. However, there is potential for helper agents to assist educators also.

A useful example is **NetMechanic**, which searches your site for broken links and informs you of them. As more educators use web sites for distributing information to students, for example, more attention needs to be paid to keeping the links correct and working.

Webbie from IBM can remember where you have been on the web, what you found and recall any word on any page you might have visited. Before you visit a page, it can alert you whether the site is available and make an assessment of how slow the access time will be for visiting the page.

Agents that can "Converse" - These agents provide entertainment (currently) as they interact and "converse" with the user. The first of these was Eliza, but there are now many more. Within the near future, they might provide information such as you might look up in an encyclopedia via dialogue. Although at the current time, most of the chatterbot agents are only for amusement, more and more they will begin to resemble intelligent tutors who can provide information on demand. Already the potential exists to create agents to assist in providing information in a timely manner to students.

Millie, for example, can inform and answer questions about the Y2K problem.

MegaHAL learns from what you say to him and uses this information.

Another such agent is **Theresa**, an expert on Greek mythology.

Erin the Bartender serves drinks, keeps the bar in order and chats with customers.

Agents that can Collaborate – Collaborative agents provide assistance in several areas related to collaboration such as on line chat, finding colleagues, allowing joint preparation of lessons and lectures, automatic passing of information, etc. Although only in the early formative stages, collaborative agents offer much promise and potential use for educators.

In this general category would be such agents as **ICQ** ("I seek you"), an agent program which alerts you in real time when somebody you know goes online.

Another example of this type is agent is **firefly** which attempts to recommend specific sites you will enjoy and connect you to people who like what you like. This could be most useful in seeking like-minded colleagues for collaborative research efforts for example.

Agents that can Shop - Some agents are capable of doing comparison shopping and finding the best price for an item. It is entirely possible that an intelligent agent, armed with your credit card number, could surf the Internet (or some commercial service) and purchase for you gifts according to your specifications. This type of application is just now being developed, but will appear soon. An agent could be charged with finding a CD player for under $300 with certain features and, upon finding such, actually have the authority to purchase the item. Even educators might find use in this information for shopping for items of use in the classroom (or for personal use).

Examples of this type of agent include **bargain finder** from Andersen consulting, which attempts to find the best price on CD's.

Another interesting shopping agent is a recent one, **Excite**. Excite lets you automatically shop for garden supplies, movies, jewelry, music and sports equipment.

Price Watch looks for the best prices in computers.

MX BookFinder looks for books at the best prices, including examining several of the online bookstores.

Conclusion

It appears that the era of the Intelligent Agent has begun. For educators this will be at least a partial solution to dealing with the information overload problem. As more and more information becomes available, it will be IA's which assist educators and students in managing this hyperabundance of data. Intelligent agents are already available to perform some functions and will rapidly acquire increasing levels of "intelligence", making them more valuable in providing assistance to educators.

Some Interesting WebSites:

Sources of IA information:
www.cs.umbc.edu/agents/
www.botspot.com

URL's for Agents:

www.agentware.com	(autonomy agent)
fishwrap-docs.www.media.mit.edu/doc/	(fishwrap)
www.agents-inc.com/	(firefly)
bf.cstar.ac.com/bf/	(bargain finder)
bf.cstar.ac.com/lifestyle/	(lifestyle finder)
ahoy.cs.washington.edu:6060/	(ahoy)
intel.com/iaweb/newsrdr/index.htm	(smart newsreader)
www.excite.com	(excite)
www.cs.umbc.edu/~smart-ht/	(CNN+ newsfilter)
ciips.ee.uwa.edu.au/~hutch/hal/HEX/	(HEX)
millie.y2klinks.com	(Millie)
www.pricewatch.com/	(Price Watch)
www.netmechanic.com/link_check.htm	(Net Mechanic)
www. http://georgia.flashpoint.com.au/	(Robobear)
www.oz.com/ov/agents/main_index.html	(Theresa)
www.mx.org	(MX Book Finder)
www.morganbanks.com.au/cgi-bin/profiler.cgi	(Job Hound)
www.adone.com/html/hound/adhound.html	(Ad Hound)
www.extempo.com/webBar/index.html	(Erin the Bartender)
calendar.stwing.upenn.edu/	(Remind U-Mail)
www.netmind.com	(Net Minder)
www.amazon.com	(Amazon.com bookstore)
connectedpc.com/iaweb/selagent/bottom.htm	(selection recognition agent)
www.merc.com/index.cgi	(NEWSpot)
www.dev-com.com/~rfactory/webcatcher.html	(WebCatcher)
www.networking.ibm.com/wbi/wbisoft.htm	(Webbie)
www.ffg.com/whacker/	(WebWhacker)

Reading Sources

Wayner, Peter, Agents Unleashed: A Public Domain Look at Agent Technology, Boston: AP Professional, 1995. ISBN: 0-12-738765-X.

[1]Decision Support Systems, 3rd Edition, pp. 137-145.

[2]Intelligent Software Strategies, Vol XI, No. 1, pages 4-5.

[3]Intelligent Software Strategies, Vol XI, No. 1, pages 4-5.

[4]Intelligent Software Strategies, Vol XI, No. 1, pages 4-5.

[5]Intelligent Software Strategies, Vol XI, No. 1, pages 4-5.

[6]Intelligent Software Strategies, Vol XI, No. 1, pages 4-7.

[7]Intelligent Software Strategies, Vol XI, No. 1, pages 4-7.

[8]Intelligent Software Strategies, Vol XI, No. 1, pages 4-7.

Lessons Learned
Formal Computer Laboratory Exercises

Richard Kuntz
Monmouth University
West Long Branch, NJ 07764
(kuntz@monmouth.edu)

Abstract:

This paper discusses some of the issues involved and lessons learned in including a formal computer laboratory component in a freshman mathematics course for Business Administration Students. It will focus on both the technical and the pedagogical aspects of that component of the course.

During the Spring 1996 term, the author began the task of including "formal" laboratory exercises as a component in the required first term mathematics course for the Business Administration Students. The concept is modeled after traditional physics and chemistry laboratory exercises. Students are expected to attend a scheduled laboratory session supervised and moderated by the instructor. They are encouraged to work in teams of 2 to 3 students and are required to submit a formal laboratory report for the team. The laboratory exercises typically involve solving problems that resembled "real world" situations with data that normally is beyond reasonable paper and pencil solution techniques. The exercises require a written conclusion or a recommendation for action for the simulated problem.

The course has evolved into one based upon the concept that students are expected to do a reasonable amount of work each week. That concept is reinforced by alternating a short 30-minute quiz one week with a formal laboratory assignment the next – there are six quizzes and seven laboratory sessions during the term.

Each laboratory session is centered around a 40 to 50 minute time frame. For the past fifteen years, the course has met twice a week for 75-minute sessions. The laboratory sessions are scheduled during one of the regularly scheduled class sessions. Because of the 35 student class size and the 20 student laboratory size, the 75 minutes meeting times are divided into two laboratory sessions. However, beginning with the Fall 1998 term, the weekly meeting schedule will revert back to the traditional three 50-minute sessions, the class sized will be lowered, and the computer laboratory facility will be enlarged so that an entire class can be accommodated in one 50 minute laboratory meeting.

While no formal pre and post analysis or control group comparisons have been conducted, the author will report on surveys that were conducted. The general conclusion, so far, has been that the overwhelming majority (90+%) of students would recommend the "technology infused" course to others.

Instructional Setting:

Monmouth University is a private, comprehensive, teaching university enrolling approximately 4500 students of which 3200 are full-time undergraduate students. The University is located in the central shore area of New Jersey – about 55 miles south of New York City.

The Mathematics Department of the University has twelve full-time faculty members. The Mathematics Majors program at the University enrolls about 35 full-time equivalent students; a significant number of those students are dual majors – mathematics and education. As with many institutions of today, a substantial part of the teaching responsibility of the Mathematics Department at Monmouth University is directed toward instruction for non-majors with many in the non-science disciplines. The largest single population serviced by the Mathematics Department consists of those students majoring in programs within the Business School. Those students are required to complete three (3) credits in "traditional" finite mathematics, three (3) credits in applied calculus, and three (3) credits in statistics. The first course, Math 111, will be the focus of this paper. Eight sections of Math 111 are offered during the fall semester and five sections during the spring semester. A total of about 500 students enroll in the course on an annual basis.

The University has a computer network, called HawkNet, with a fiber optic backbone that connects all academic buildings including the library and the nine residence halls. Students residing in the residence halls may subscribe to a network connection on a no-charge basis. There are 15 "public" computer laboratories in size ranging from six to thirty computers, and an "Information Commons Area" in the library. One of the computer laboratories is located in one of the new residence halls. Each of the other residence halls contain at least two public accessible computers in the lobby. All of these facilities are accessible to students and support access to the campus network and "web" server. In addition, the University supports a modem pool for dial-in access and a T1 link to an Internet Service Provider.

As of this writing, all but a few of the 185 faculty at the University have a PC on their desk. Except for the Computer Science Faculty who have SUN Workstations and the Art Graphics Faculty who have Apple Macs, most of the faculty systems are Pentium class units. All are connected to HawkNet and most are running Windows 95 with the standard productivity tools (either Office 95 or Office 97) along with Netscape.

Faculty members in the Mathematics Department have at least a Pentium P75 system. The Department has standardized on Maple V as the mathematics symbolic algebra language. In addition to the standard productively tools, all faculty workstations have a copy of Maple V installed locally on their systems.

The Mathematics Department computer laboratory contains 16 workstations networked through a 10 Meg, 3Com Super Stack switching hub. The workstations in the laboratory are 133 MHz Pentium computers with 32 Meg of main memory. Most of the laboratory software resides on the local hard drive. The laboratory workstations are connected to the Mathematics Department server through a 100 Meg port on the switching hub.

The mathematics server is running Windows NT 4.0 with the latest service packs and fixes. It is administered locally by the Mathematics Department and acts as both a file server for individual faculty files that are publicly available in the laboratory and as a "web" server for Mathematics Department information and faculty course notes.

Each faculty member is provided with two main directories on the server. One directory is for notes and files that are accessible from the computers in the Mathematics Laboratory. Normally, those files are templates or data files to be used as source material for student exercises in Maple, Excel, or SPSS. These files are only available in the Mathematics Laboratory. The other main directory is for the distribution of course materials over the Web. These materials include course notes, study guides, and homework assignments. The Web materials are part of the University Web structure and, as such, are accessible from any station that has Internet Web access. The URL of the Mathematics Department Home Page is: "mathserv.monmouth.edu"

Course Description and Rationale:

Math 111 is the first in a three-course sequence that is designed explicitly to provide the basis for quantitative business decision-making techniques. The general theme of the three-course sequence follows the concepts being developed by the Villanova Project which, in part, is being funded by the National Science Foundation. That project is developing course materials that support decision making in business applications. The focus of the effort is to provide resource materials in the "reformed tradition;" that is, to show the connections between the mathematical topics and the student's world – present and future.

The Math 111 course provides the foundation in model building that will be used throughout the sequence. This foundation includes: clarifying a problem, making estimates, defining variables in a problem, fitting a model to the problem, finding mathematical solutions by solving model, and validating a mathematical solution against plausible answers to the original problem.

Given that our world is technology oriented, the courses in the sequence will make the most of that reality. It is estimated that a Business graduate entering the career environment will have a 90% chance or better of having computer on his/her desk. Consequently, the use of technology is an important component of the course sequence. Students are expected to be able to solve simplistic problems by hand to master the concepts, however, it doesn't stop there. Through the use of an integrated series of laboratory exercises, students experience solving realistic problems using data that would be too laborious to manipulate without the aid of technology.

The graphing and symbolic manipulation capabilities there are available through many of the technologies of today minimize many of the traditional topics. For example, how can we justify spending considerable time and effort on factoring algebraic expressions beyond the quadratics, solving complex equations, curve sketching, or laboriously calculating simple statistical measures. Even though the use of technology in mathematical instruction may reduce the need for some skills, it does increase the need for other skills. One of the important skill objectives that is included in the course is recognizing that a solution is not just a bunch of numbers on a piece of paper. The laboratory component stresses the importance of relating the numbers to the real-world problem, an interpretation, or a conclusion.

The formal catalog description for the course is:

Catalog Description: Linear equations and models, systems of linear equations and applications, matrices and techniques for solving systems of equations, linear programming and applications, quadratic functions and models, and exponential and logarithmic models. A computer laboratory component is incorporated. Designed for students majoring in Business Administration; other student majors by permission of the Mathematics Department.

Laboratory Component and Objectives:

The laboratory component of the course consists of seven "formal" laboratory sessions. They are formal in the sense that students are expected to attend and participate. As mentioned above, they are modeled after the traditional chemistry and physics laboratories.

The first exercise is intended to introduce students to some basic techniques for using the technology to assist in problem solving and some of the basic syntax of the Maple V application. The remaining six laboratory exercises present students with simulated problems that resemble real world applications.

The goal of the laboratory exercises is not to have students become proficient in using the Maple technology – it is unlikely that they will have access to such technology in the business world. It is, however, to have students recognize that such technology exists and that it can be used in a relatively easy manner to solve problems that normally could not be solved by pencil and paper. When the need for technology assistance occurs in the work place , they would be able to go searching for assistance with the knowledge that they are capable of using some technology even though they "never had a course in it." They would not be intimidated by technology – that is, they would have a comfort level toward using technology in problem solving.

In addition to the "comfort level" idea, the following list describes other objectives of the laboratory component, which are, in fact, elements in the "reformed movement:"

1. Foster and encourage collaborative problem solving through the "team" approach to comple- ting individual assignments as well as inter-team discussions during the laboratory time.

2. Relate mathematics to solving real-world problems – in particular, the problems have been selected so that students can directly relate to their applications and can translate the problem to the realm of even bigger and more complex applications.

3. Require conclusions, reports, and/or recommendations – solutions to real-world problems are not just numbers. Students are required to communicate the solution in terms of written reports.

4. Demonstrate that maybe more is not better – with the inclusion of the formal laboratory exercises, less content is being covered. However, students should be able to relate to more mathematics.

Laboratory Implementation Details:

At the current time, the official class capacity for Math 111 is 35 with an average class size of 30. Since the current laboratory has seating for 20 students with 16 workstations, classes are divided into two sections. The scheduled 75 minute class meeting is divided into two 40 minute sessions — the first half and the second half. During the first class meeting before the first laboratory, a signup sheet is sent around requesting students to sign up for the first session or the second session. For the second lab, students are told to attend the same session. After that, no mention is made of sessions; students seem to even themselves out and there has been very little contention over seats.

Each laboratory exercise consists of a sample Maple worksheet and an assignment template. Normally, the sample worksheets contain a solution to a very simplistic problem. The purpose of the sample is to demonstrate a possible sequence of steps that could be used to solve the problem as well as to illustrate the syntax of the Maple commands that can be used to solve the problem. In some cases, the sample worksheet demonstrate several different approaches to solve the problem.

The sample worksheet and the assignment template are available through a network share drive. As mentioned above, these files are only available in the Mathematics Laboratory. The Mathematics Department has recently acquired the Adobe Suite of software. We are beginning to make the sample worksheets and templates available through the PDF read-only format on the web server. At this point, not all the worksheets have be made available. The rationale is to require students to obtain copies of the material before the lab and come prepared. At this point, that might be a little too much to expect.

A brief overview of the laboratory exercise is presented during the end of the class period that precedes the laboratory. It's unclear how much time needs to be spent preparing the students for the laboratory.

Often the most difficult part that students have in solving problems is getting started. The purpose of the assignment template is to provide a starting point – they are not starting with a blank worksheet and spending time searching for a plan to get started. Even though there is value in devising a solution plan, it is the author's belief that it is more valuable to put the time in at the end – that is, translate the mathematics back to the real problem and describe the solution. We have found, that in the allocated laboratory time, it is unrealistic to require creative thinking at both beginning and the end of the assignment. The template provides suggestions on the steps to following – that is, the order of the activities.

Students are required to sign an attendance sheet for each of the laboratories. There is an implication that if they do not attend a laboratory session and they do not have an excused absence, they cannot get credit for the laboratory. Point of fact is that we haven't needed to go so far as to enforce the attendance; the unsaid implication has kept attendance at a high level. Students who miss the formal sessions have talked to the instructor and have worked out a method to insure they are doing their own work.

Approximately 75% of the teams complete the laboratory assignment during the scheduled lab session. Those that do not, have until the next scheduled laboratory to complete the assignment.

With the increased laboratory capacity and extended time allocation, the author is planning to require a preliminary report in the event that the assignment is not completed during the allocated time period. A further description of that idea is contained in the Future Section.

Testing and Evaluation:

Each of the laboratory assignments is graded on a point scale of 1 through 5 with 5 being the highest grade. Most of the scores have been between 3 and 5. One point (20% of the grade) is allocated to conclusion – the write up for the report and one point to the overall neatness of the report. The rest of the points are for general content and syntax. In general, a team has to work fairly hard at not getting a passing grade. Very seldom has a report been turned in that demonstrated a "really don't care" attitude. In general, we are fairly accommodating in providing assistance on syntax errors and procedure errors and have been rigid in requiring students to provide their own conclusions and analysis. The laboratory grades represent 20% of the course grade.

All members on a team receive the same grade. Since the teams are self organizing, the "coasters" (the students that don't contribute but expect to have their name on the report) seem to catch on after a while. It's a self-correcting problem, since other students avoid having them on their team.

Students are not tested directly on any Maple syntax. Occasionally, syntax from a previous laboratory exercise is required. In that case, students ask for assistance or use the help feature of Maple. We suggest that students use the Maple help on syntax by going directly to the example section. By entering three question marks followed by the command, Maple will go directly to the example section; for example "???plot" will display a set of examples on the use of the plot command.

Even though there is no direct testing of the laboratory exercises, an occasional quiz question will require that students to explain in their own words the processes that were followed in the laboratory exercise. These have been tough questions for the students to answer.

Summary of Individual Laboratory Exercises and Objectives:

Laboratory 1 – Introduction. In this introductory laboratory exercise, students learn to enter Maple commands and text into a worksheet, execute a worksheet, use the Maple commands *plot* and *solve*, and use the assignment command to represent long expressions. The exercise has two parts: plot three lines to discover the concept of simultaneous solutions and to approximate a line of best fit by adjusting the slope and intercept parameters until the fit looks "good."

Laboratory 2 – Supply and Demand. This exercise involves data points collected by a consultant relating to pricing and demand of tuition charges – linear demand curve; and information from a university financial office on the willingness (ability) to provide services related to pricing – linear supply function. Students are expected to translate the verbal description of the problem into mathematical models for the demand and supply curves, use the technology to estimate the point of market equilibrium, and to prepare a report representing a recommended tuition level. The report must contain at least a graph of the demand and supply curves.

Laboratory 3. – Budget Manipulation using Matrix Algebra. Matrix operations are used to manipulate data representing budget information. Rows in the matrix represent budget categories while columns represent departments. Students discover how to use scalar multiplication to accomplish "across the board" increases or decreases, and how to use matrix multiplication to adjust the budget through category changes and to eliminate the allocation of a given department.

Laboratory 4. – Quadratic equation containing data points. This exercise demonstrates solving a system of linear equations to find the quadratic equation containing three non-collinear points. The exercise is presented in the context of a business that attempts to maximize job satisfaction. Jobs within the company are divided into two broad types – Type A and Type B. Type A are the narrow, well-defined positions while Type B are the more demanding and challenging positions. An analysis collects three data points relating satisfaction to age for both of the job types. The exercise requires that students use the data points to estimate a quadratic satisfaction model for the two types of jobs. From these mathematical models, students are requested to submit a recommendation to the CEO suggesting appropriate age levels when employees should be considered for movement between job categories with the objective to maximize job satisfaction.

Laboratory 5 – Graphical solution to Linear Programming Problem. This is a Two Product Farm Scheduling Problem that attempts to maximize total income. The problem involves resources – available land, capital, and storage capacity. Students are requested to use the Graphical Method to solve the corresponding Linear Programming Problem. Students are expected to prepare at report to the farmer with a planting scheme recommendation. As it turns out, they will need to justify to the farmer why not all the available acreage is used. They need to explain how the available resources impact on the planting scheme.

Laboratory 6. – Standard Linear Programming Problem (many variables). This exercise simulates an Admissions Yield and Recruitment Constraints Problem. A University has a desire to enroll an appropriate mixture of the four categories of students, but in such a way that the admission decision process is nondiscriminatory; that is, only academic admission criteria is used in the decision process and not the fact that a specific category is over subscribed. The data for the problem consists of estimated yields of new student enrollments in four categories of students from five different geographic regions. The constraints for the problem consists of statements representing desired enrollments in each of the four categories of students and statements regarding geographic diversity.

Laboratory 7. – Fitting Exponential Growth Curve to data. This last laboratory exercise involves using an exponential function to model the recorded HIV infections for the 10 year period from 1981 through 1990. To estimate the model, students are given an exponential function with two variables (the base is given as 1.5). They are expected to adjust the values of the variables, as in Exercise 1 for a linear model, until they have a good fit. After they have constructed a model that estimates the data, students are asked to make future projections. The exercise ends with asking students to comment on their projections for 2000 and 2010 related to the total world population.

Problems:

In an open collaborative environment – such as our computer laboratory experiences – one of the biggest issues is the encouragement and support of team efforts while at the same time discouraging just coping another team's or person's report. Our position at the present time is to err on the collaborative part. That is, encourage team and inter-team collaboration but not accept duplicate reports – everyone fails the project report when duplicate reports (or sections) are submitted. Rephrasing and/or rewriting of ideas and sections are accepted.

Maple V is a syntactical and non-forgiving language. Students experience difficulty remembering the proper syntax for a given command. Initially a lot of time is spent assisting students with syntax errors. Their ability to appreciate the importance of proper syntax grows with experience. This is one of the areas where having a lab assistant in the room is very helpful. The assistant resolves most of the syntax problems, while the instructor works with the team in the thought process that goes into solving the problem.

Even after students have been able to understand and articulate the problem, there is a tendency to mimic the sample worksheet regardless of whether or not a particular step is related to their problem or not. Some have very rudimentary skills in "knowledge transfer."

Students have difficulty relating to the fact that mathematics is more than numbers on a piece of paper. They find it difficult to draw conclusions from the data, to put the mathematics in the form of a recommendation for a course of action; or to just describe what they did in the experiment. Initially, we get incomplete thoughts or poorly constructed sentences. However, by the end of the course, students recognize that something has to be said and most make attempts at writing something; albeit a single sentence.

When a problem statement is long, students have difficulty getting to the essence of the problem. We have attempted to present problems to which students can relate and have a frame of reference.

We have found it difficult to get students to come to the laboratory sessions with some semblance of preparation. Just providing written material that is either handed out in class or made available through the web is not adequate. Students don't feel obligated to prepare for the experience. We resorted to spending time in class preparing students for the experience. We'll spend 5 to 10 minutes discussing the goals and objectives of the forthcoming laboratory. In some cases, we'll discuss the steps and the command syntax..

The Maple V software is only available to students in the Mathematics Laboratory. Student versions of the product are available through the University Bookstore, but few, if any, purchase a copy for their own computer. This is understandable; but it does create a problem when students would like to use the software to solve problems not directly related to the formal laboratories. Where we are going on this issue is not known at this time.

The current laboratory has been configured as a teaching laboratory ___ with a front of the room where the instructor would lecture. The workstations have been arranged to facilitate viewing the instructor and not for encouraging teamwork or permitting circulation around the room.

Successes:

Each class has its own personality and dynamics. Even the silent classes – where it is nearly impossible to get the students to speak, ask questions, or in anyway provide feedback – experience a transformation in the laboratory component of the course. They do talk, they do interact and participate, and they do smile.

When one acknowledges the fact that almost no one in the course is there because of their choice, it is somewhat telling that most students recommend the technology infused format (computer laboratory) over the traditional course.

The student confidence level in using technology to assist in problem solving increased through the experience.

Future:

In order to accommodate the laboratory component in the revised schedule format of three 50 minute class meetings each week, the class size has been reduced from 35 to 30 students and the computer laboratory will be increased from 16 workstations to 30 workstations. These changes will permit the entire class to meet in the laboratory for the scheduled 50 minutes.

Each student registered for the Math 111 course will be assessed a computer laboratory fee of $40. Those funds will be returned to the Mathematics Department to support the laboratory expenses which include computer maintenance and upgrades, software licenses and acquisitions, and laboratory assistants. The laboratory will be open from 9 am to 10pm – Monday through Friday. During open hours, it will be staffed by a laboratory assistant, who will be able to assist students with technology and software syntax related issues. In addition, each scheduled laboratory session will have one lab assistant in the room along with the instructor.

The dramatic increase in the number of students that will be participating in the laboratory exercises next Fall has raised some concern over plagiarism of the laboratory exercises. Several ideas to minimize the issue are under discussion. The one that seems most promising at this point is the concept that students will be required to turn in the assignment at the end of the session or will present the instructor with a work-in-progress document that will be signed by the instructor. That signed work-in-progress document will permit the student to continue to work on the assignment. To receive credit for the assignment, the student must attach the authorized work-in-progress document with the final report.

An on-line Maple Tutorial directed to the needs of students in Math 111 and the subsequent courses will be developed. The current Maple tutorial prepared by the Department contains too much material that is beyond the need and reach of students in the sequence.

Student Surveys:

Appendix A contains a copy of the survey completed by the 55 students who were in attendance when the survey was administered.

Conclusion and Recommendations:

Clearly, the formal laboratory component does diminish the total amount of content covered in the course. However, our experience so far indicates that students feel a little more comfortable with mathematics, they generally interact and "come alive" in the laboratory, and they recommend the laboratory experience to others.

Faculty that engage students in the collaborative problem solving, laboratory approach must recognize that it is not possible to identify performance with any given student Some of our faculty have expressed concern over that loss of identification.

One of the significant concerns expressed by students in prior year's exercises was the wait time for assistance. In previous terms, the instructor was the only support person in the room. This problem was alleviated this past spring with the addition of a trained laboratory assistant. At this point, it appears that an appropriate support formula is one professional (instructor or lab assistant) for each 10 to 15 students.

We have found that it is unrealistic to expect students to prepare ahead of time and come to the laboratory prepared. Consequently, we spend about 5 to 10 minutes at the end of the class period just prior to the laboratory session explaining the laboratory objective.

Don't expect students to be programmers. Maple is a highly syntactical, non-forgiving language. In the laboratory environment, we quickly resolve syntax problems, but we require students to work out their own process goals – what they want to do and the order in which to do it.

We have found the need to continuously reinforce the requirement that individual team reports be unique. Unfortunately, the technology in the laboratory makes it easy to duplicate reports.

The Laboratory environment should facilitate moving around. Both students and lab assistants need to be able to easily navigate the room. One possible room configuration is to have a set of stations around the perimeter of the room with a free standing island in the middle. Our laboratory is 730 sq. ft. It will be reconfigured to accommodate 30 stations using that scheme.

It is unlikely that a laboratory experience for more that 20 or 25 students will be successful without a networked environment where the requisite files and templates can be placed. Along with that, it is imperative that the faculty member responsible for the creation and updating of the files have "easy" and "complete" access to the directories containing the files. There will be times when last minute changes and additions will have to be made.

There has to be general acceptance that the goal of the laboratory component is to get students comfortable with using technology to solve problems and not to cover more material in a more "efficient" manner.

Appendix A

Course/Instructor Evaluation

Professor:______________ Course:__________________ Section:_____ Term: _______

1. Of the on-line class notes that were made available, I obtained:
   1=none, 2=25%, 3=50%, 4=75%, 5=all
   Response: 1-0; 2-6; 3-8; 4-10; 5-31 1 2 3 4 5

2. In general, the value of the on-line class notes was:
   1=no value, 2=minimal, 3=average, 4=helpful, 5=extremely helpful
   Response: 1-0; 2-4; 3-9; 4-23; 5-19 1 2 3 4 5

3. Of the laboratory meetings that were scheduled, I attended:
   1=none, 2= 25%, 3= 50%, 4= 75%, 5=all
   Response: 1-0; 2-0; 3-1;4-11; 5-43 1 2 3 4 5

4. In general, the value of the laboratory experience was:
   1=no value, 2=minimal, 3=average, 4=helpful, 5=extremely helpful
   Response: 1-2; 2-4; 3-14; 4-22; 5-12 1 2 3 4 5

5. In general, the value of the collaborative problem solving experiences in the laboratory was:
   1=no value, 2-minimal, 3=average, 4=helpful, 5=extremely helpful
   Response: 1-0; 2-6; 3-11; 4-28; 5-10 1 2 3 4 5

6. Regarding the use of computer technology, at the beginning of the course I was:
   1=apprehensive, 2=somewhat fearful, 3=neutral, 4=comfortable, 5=very adept
   Response: 1-4; 2-9; 3-20; 4-19; 5-3 1 2 3 4 5

7. Regarding the use of computer technology, at the end of the course I now feel:
   1=apprehensive, 2-somewhat fearful, 3=neutral, 4=comfortable, 5=very adept
   Response: 1-0; 2-1; 3-6; 4-35; 5-13 1 2 3 4 5

8. For me, a frustrating aspect of the laboratory requirement was:________________________

9. From my experience, I (circle one) would/would not recommend this technology-infused course to others.
   Response: would - 49; would not - 5

10. Taking all factors into consideration, the general rating for the course is:
    1=poor 5=excellent
    Response: 1-1; 2-1; 3-5; 4-31; 5-19 1 2 3 4 5

General comments on how to improve the course are appreciated. In particular, your general thoughts on the (a) electronic mail communication, (b) on-line class notes, and (c) laboratory exercises are most welcome. Use the space on the back for these comments.

Student Support that Works: A Solid Approach

Kathy J. Lang
Director of Technology Support Services
College of Saint Benedict/Saint John's University

Collegeville, Minnesota 56321
(320) 363-2986 (office)
(320) 363-2761 (fax)
E-mail: klang@csbsju.edu

Ruth Bielejeski
Software Support Manager
College of Saint Benedict/Saint John's University
Collegeville, Minnesota 56321
(320) 363-2868 (office)
(320) 363-2761 (fax)
E-mail: rbielejeski@csbsju.edu

Introduction

Effectively supporting students' information technology (IT) needs is always a hot topic. The issues focus on identifying students' needs, implementing services that will meet these needs, providing these services effectively, and offering them within a limited budget.

The College of Saint Benedict and Saint John's University (CSB/SJU), two coordinate, liberal arts colleges, have implemented a student IT support system that works. Five years ago a disorganized, paper-based system was in place. Today, there is a full-service, cost-effective system that not only meets student needs but is pro-active.

This presentation will focus on the strategies used to design, implement and sustain a successful IT support system. It will provide practical ideas on developing a strong student support staff and expanding services to meet increasing demand.

History

The College of Saint Benedict and Saint John's University, while operating as two financially separate institutions are joint in all of their academic programs. In addition, many of the academic support areas are also joint, including the computing support. In February 1992, the two separate computing support areas, Academic Computing and Administrative Computing, merged. The new department, Computing Services, was designed to be more responsive to the needs of the institutions.

At the time of the merger, there was no formal plan for academic computing and the department had been sorely funded. There was a committee of faculty members, called the JCAC (Joint Committee on Academic Computing). During the spring of 1992, they put together a needs assessment that clearly showed the great demand for better academic technology. It was this needs assessment document that started a process that would fundamentally change the way the two colleges approach planning and support for academic technology.

Project IMPACT

The response to the needs assessment was a proposal called Project IMPACT (Implementing Academic Technology). This was a 5-year, $10 million plan to upgrade the technology for student and faculty use. Project IMPACT was approved by both institutions and implemented on July 1, 1993.

The JCAC's computer needs assessment report described the current academic computing resources as being "so limited as to threaten the competitiveness of their programs." In most areas the resources were lacking but especially those for student support. At the time, there were three student access areas, all of which only provided dual-floppy PCs. In addition to the old equipment and software, the labs themselves were unattractive and support for students was virtually not available. There was a small staff of student employees, called lab assistants, whose main task was to check software in and out. These lab assistants, however, only had limited training so were not able to provide much assistance to students with questions.

One of the first goals of Project IMPACT was to upgrade these computer access facilities. The result was five new computer access areas for student use. As these new areas were created, emphasis went not only to the new software and hardware but also to the aesthetics of the facility. These access areas now serve as a recruiting tool that showcase the technology available to all students.

Student comfort was another reason why aesthetics were considered a primary factor when upgrading the access areas. Each lab was designed to spaciously accommodate students. This included workspace around each computer, student collaboration space where appropriate and handicap accessible workstations. The emphasis was placed on a professional look and feel.

As the project progressed many mini-labs and residence hall computer clusters were added, more buildings and rooms were wired for access to the network and faculty were provided with development opportunities to utilize the technology. More importantly, the structure for student support was radically changed.

Student Support Structure

The first changes that were made in regard to student support were made because of necessity. Various staff members each had a few student employees that they supervised. Each student determined their own schedule and the staff members tried to make sure all the open hours in the access areas were covered by a student employee. This haphazard way of supervising the students created a system of no accountability and chaos. The structure was changed so that one staff person became the student employee staff supervisor. This created a more formal reporting structure for the students.

A student supervisor position was also created. This student was able to assist the staff supervisor with the various responsibilities. The student supervisor was also available during the evenings and weekends to assist the student lab assistants.

As more access areas were added and the times the areas were open for use expanded, more student employees had to be hired to fills these timeslots. Whenever an access area was open for student use there was to be at least one student employee available for assistance. With the number of student employees growing, it became harder to manage all of them with just the one staff and one student supervisor so more restructuring took place.

The student supervisor's title was changed to Student Manager. Then five Student Supervisors were hired. Now each supervisor is responsible for one access area and is accountable to the staff supervisor.

Hiring Process

Throughout the restructuring process an emphasis went on hiring the right students to fill each of the positions. The development of new job descriptions that contained detailed specifications of the job's responsibilities was extremely beneficial in the hiring process. One of the main criteria used when interviewing students was a high level of motivation to succeed. The long time career benefits of a student employee management position were also stressed with the applicants.

The last part of the restructuring process was to develop a long term hiring plan. Approximately one-quarter of the student employee staff has to be replaced each year. The goal is to fill these positions with highly motivated and responsible students who will be responsive to users' needs. Two factors affect this goal. One is the advancement opportunities available to student employees within the IT Services Department. The second is that students are able to schedule work hours around their class schedules.

Scheduling Student Employees

The next change came in regard to scheduling access area coverage. Originally, students were allowed to make their own schedules. Scheduling meetings were held each semester and the students signed up for hours starting with the seniors and finishing with the freshmen. This method resulted in many problems including not fulfill each student's work award and not accommodating their class schedules. It also created a lot of ill will among students.

The new method of scheduling was for the student employee staff supervisor, the student supervisors and the student manager to do the scheduling themselves. They start with a copy of all the employees' class schedules and first create the weekday schedule. Then they devise a rotat- ing weekend schedule so that each student has to work only one out of every three weekends.

Training

Attracting and hiring good student employees is not enough however. They must be trained from year to year. Therefore, the next phase was to revamp the training program for student lab assistants. The new training program focuses on two major areas: technical skills and customer service skills.

Mandatory training and optional training are provided. The mandatory training includes technical workshops, customer service training specific for working in the computer access areas and customer service training provided by the Financial Aid office. Approximately twenty hours of mandatory training are required for each new student employee. Returning student employees are required to take an additional ten hours of training every year.

An orientation training session was also developed. All student employees are required to attend one of these sessions each fall semester. These sessions not only provide students with an introduction to the department and their responsibilities, but also provide them with an understanding of the importance of their role and how they fit into the larger mission of the colleges. The groundwork is laid for those students interested in moving to a Student Supervisor or Manager position in the future. In addition, expectations are set, policies and procedures are reviewed and the mandatory training requirements are explained.

Mentoring Program

A mentoring program was developed as part of the orientation process. Each new student employee is partnered with a returning student employee. This gives the new student a chance to ask questions on a one-to-one basis and to become familiar with the job responsibilities in a non-threatening environment. It also benefits the returning student employees, allowing them to refresh their skills and increase their sense of challenge while fostering teamwork.

Certification

The optional training has developed into what is called the IT Services Student Employee Certification. Certification has proven to be a positive motivator for student employees. The requirements are that a student must

- have been an employee of the department for no less than one academic year
- have completed all mandatory training sessions during that year
- have been a mentor for a new employee
- have no unexcused late-shifts or missed shifts
- have completed the advanced technology workshops
- have completed additional customer service workshops
- have had an above average end-of-year performance appraisal

No student can be promoted to a Supervisor or Manager position without earning the certification.

Student Employee Expectations

The lab assistant's first priority is to assist the users in the access area. They must work well independently and are expected to project an image of service, deal with a variety of personalities and be self-motivated. Students also are expected to maintain an excellent attendance record.

The expectations of the student employees are clearly defined and communicated. All the students receive a detailed IT Services Student Employee Policies and Procedures Manual when they are employed. Also, the reprimand, warning and termination policies are used when necessary.

Solid communication between the staff supervisor and the student employees is essential due to the independent nature of the positions. Practically all communication is done electronically. The department is dependent upon e-mail for daily communication such as shift changes, up-to-the minute lab changes, etc. An electronic bulletin board accessible only to the IT Services department has proven to be an easy and effective way to organize and share policy and procedure information with student employees.

High importance has been placed on keeping the computer access facilities open and running. Student employees need immediate access to the student manager or supervisors in the case of equipment failures and problems with shift coverage. Therefore, the student manager or one of the supervisors carries a pager at all times. The student supervisor and manager also have access to the staff supervisor via a cell phone. This effective communication system allows quick resolution of problems and has resulted in virtually no down time in the access areas.

Residence Hall Support

Project IMPACT provided funding to wire all of the student rooms in the residence halls and enough ports were installed for each student in the residence halls. Each student can connect a personal PC and have access to all the software products running on the campus network. This includes access to the Internet.

As the number of students connecting personal PCs in their rooms increased, there was a definite need to provide in-room support services. Again, student employees were hired to fill this need.

These student employees, called Residence Hall Computer Assistants, (which increased from 2 to 8 over a period of three years) must have a technical background and the desire to learn both the hardware and software aspects of connecting to the intercampus network. Students selected for this position need to be willing to accept new responsibilities and challenges every time they walk into another student's room. The number of Residence Hall Computer Assistants has grown from two to eight and will continue to increase as more students connect personal PCs in the residence halls.

Special training is provided for these students and they are also encouraged to complete the certification program. These students assist in writing documentation for connecting PCs in the residence hall rooms. They also help define the training required for new Residence Hall Computer Assistants since they learn such valuable information while on the job.

Help Desk

Another goal of Project IMPACT was to create a professional "Help Desk" system. Prior to this a disorganized, paper-based system was used that was both ineffective and inefficient in responding to users' needs.

The new system consists of a professional Help Desk staff member and a call tracking software system. The concept was to provide a single point of contact for all computing related questions including those for hardware support. The entire IT Services staff has access to the call tracking software so calls are routed from the Help Desk to the appropriate staff member(s).

Since this was such a new way of addressing computing support needs, an internal marketing program directed at students, faculty and staff was implemented. A new phone number was established and it became the only phone number provided on any IT Services document. Even the staff members told users they should call the Help Desk to get faster response. Within sixty days the Help Desk line was recognized by users as the most effective means to quickly receive assistance.

As the Help Desk's success climbed the calls soon averaged over 1,000 in-coming calls per month. There was a backup to the Help Desk staff person but at times both people were overwhelmed with calls. In addition, this area became responsible for tracking all computer hardware inventory and coordinating schedules for the Residence Hall Computer Assistants. The need to relieve the pressure of this area became apparent and once again student employees were brought in to fill that need.

Student employees are now staffing the Help Desk on a part time basis, freeing up the Help Desk staff employee for other tasks associated with the Help Desk. In addition, they provide the backup support during the heaviest call periods. Utilization of student employees on the Help Desk has eliminated the need for adding another full-time staff member.

Students hired for these positions must present excellent customer services skills in the interview process. They are then enrolled in an intensive training program designed specifically for the Help Desk. Routine training throughout the year is provided and they are also encouraged to become certified.

Phone Support

In the continued effort to provide students with access to technology at the colleges, PCs were added to each of the residence halls on both campuses in what are called Residence Hall Computer Clusters. These areas are available to resident students 24 hours a day, seven days a week. This enhancement for students created a number of new support issues for IT Services. Most important was the need for phone support for these students. This support was added to the job responsibilities of the lab assistants.

In order to offer effective phone support to students, the student lab assistants were given training on phone etiquette and phone support techniques. These sessions have now been incorporated into the mandatory training program. Lab assistants now provide phone support to students calling from the Residence Hall Computer Clusters, individual rooms, and students who live off-campus. In addition they provide Help Desk support during the evenings and weekends.

Peer-to-Peer Training

Project IMPACT funded a full-time Training Coordinator to provide technology workshops to students, faculty and staff. However, the technology training was originally only offered during regular class hours during the day. Two years ago evening workshops were offered. These evening hours turned out to be the most convenient time for some students to attend workshops.

After analyzing many training methods, each having it's own set of strengths and weaknesses, it was decided to offer small group, instructor-lead workshops. This type of training allowed us to reach as many students as possible while keeping the instruction individualized. More student employees were hired to teach these workshops.

This peer-to-peer training program has worked very well to-date. The students found themselves in a more-relaxed, pressure free environment. Not only did the students attending training say they were learning more, but many times were able to share their knowledge with the rest of the group. The group of student instructors has doubled in the past two years and the number of workshops provided has expanded greatly.

The evening workshop program has recently been expanded to collaborate with other academic support departments. Resume building and term paper workshops jointly facilitated with the Career Centers and the Writing Centers are now being offered to students. This collaboration between departments allowed students to have both context and layout resources available at the same time.

Budget Implications

Some of the improvements in student support were funded through Project IMPACT. However, most were through procedural changes that utilized existing resources to better accommodate the support needs. Many procedural changes did not affect the budget at all. Hiring additional student employees was a budget factor but financially it costs less to hire and train good student employees than it does to hire full-time staff. With Project IMPACT, the administration made a commitment to provide a high level of support services for students.

Signs of Success

The success of the improved IT student support services is apparent in many ways. The computer access facilities run at a 90% usage rate, Residence Hall Computer Clusters run at a 65% usage rate, and training workshops are usually full to capacity. Student feedback has been very positive on the facilities but more importantly, on the available support from the student employees.

IT Services student employee positions are highly sought after among the student body. It is common to have more than ten applicants for every open position at any given time. There have even been students who have volunteered to work in the department because they either don't have a work award through the Financial Aid office or already work in another area but want the experience in IT Services.

Students realize the multiple benefits of working in IT Services. Flexible scheduling, paid technical and customer service training, and the opportunity to advance to a leadership position are just a few of these benefits. Not only are the skills they learn applicable while working within IT Services, but many are also greatly desired skills in the greater marketplace. More than 50% of former student employees have come back to say one of the primary reasons they were offered their current employment position is due to the experience they received as an IT Service student employee.

Issues in the Creation and Coordination of an Academic Computing Help Desk

Leslie A. Lucas
Academic Computing Department
Bethany College
Bethany WV 26032
304-829-7749
l.lucas@mail.bethanywv.edu

An academic computing help desk is probably one of the busiest places on campus these days as more people use computers the number and the variety of questions dealing with research and computer operation grow. Never the less, I like to imagine a day when everything works, and no one has questions: picture the Maytag® repairman sitting comfortably at the academic computing help desk, snoozing away the afternoon. Unfortunately, reality reminds us constantly that the need for computer assistance on college campuses keeps increasing. In the October 1997 issue of EDUTECH Report, an article titled "Troubled Times for Many IT Departments" describes a widely perceived support staff crisis. Faculty and staff comments about computer assistance on campuses in the EDUTECH article included: "They come in when I'm not here and change things on my machine without telling me why," "They need some bedside manners - they treat us like we're stupid," and "They are very arrogant and independent; they often don't return my phone calls." Having been on both sides of the help desk, I can understand the client's dissatisfaction, but primarily I worry about the demands being placed on the academic computing support team.

Given this seemingly chaotic scenario, academic computing directors must focus on what actions and policies make a well -coordinated academic computing help desk function effectively. Keeping in mind the needs of both the client and the support team, the key components of a well-managed help desk include:

- Accessing help easily
- Sorting support team jobs effectively
- Distributing support team jobs effectively
- Performing jobs efficiently
- Training the support team
- Training faculty, staff and students
- Making it easier for the support team to respond to clients
- Making it easier for clients to receive support

Accessing help easily

The initial contact with the help desk is one of the most important stages in resolving client problems. Methods for contacting the help desk should include walk-up, phone, e-mail, and WWW. During this initial stage the client must have easy access to help and a straight-forward method for submitting a job request. The support person running the help desk needs to be courteous and attentive. In fact, at this point in resolving client problems courtesy is actually more important than subject expertise. The person managing the desk needs to quickly respond to the person either by resolving the problem or by submitting the request to the appropriate support person in a friendly professional manner.

Sorting support team jobs effectively

After the client has contacted the help desk either the client or help desk personnel should enter the job request. The job request information needed is the name of the client, nature of the problem, and the necessary date for completion of the project. The request can easily be entered into a standard form (see figure 1) on the campus web site.

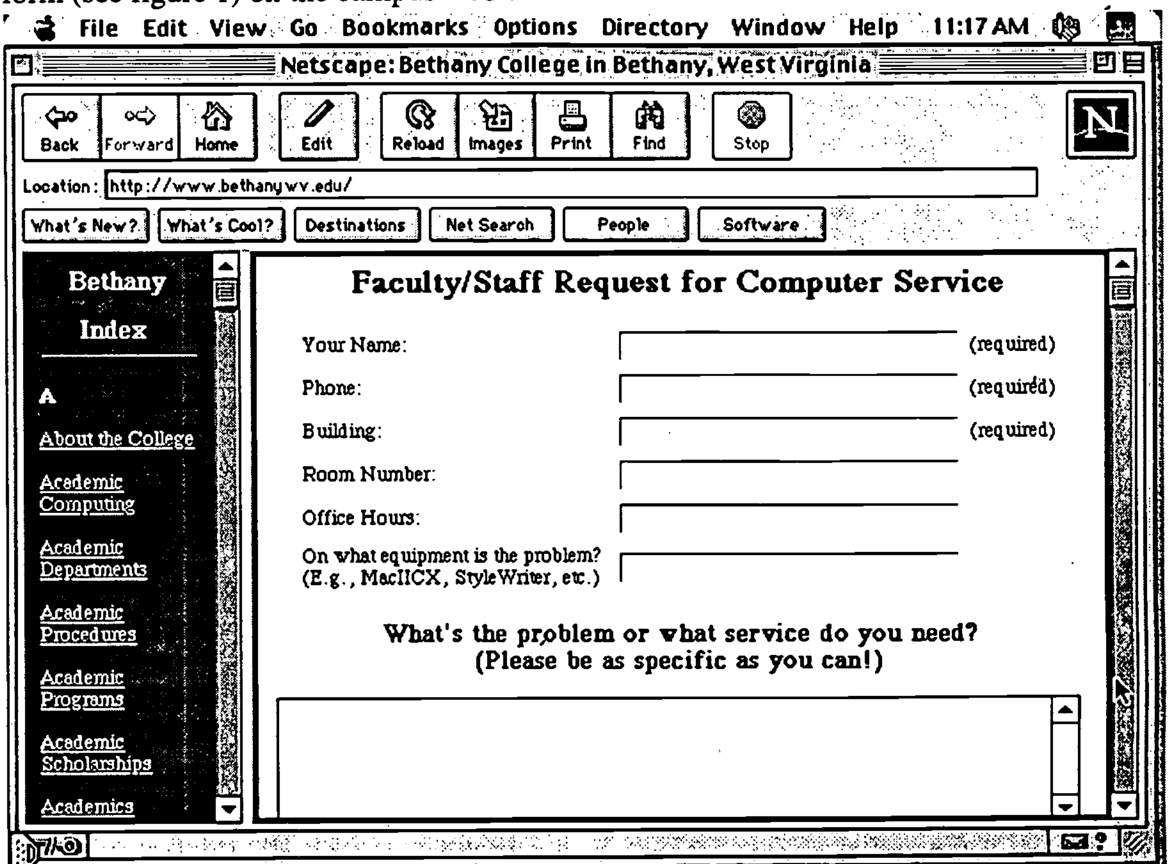
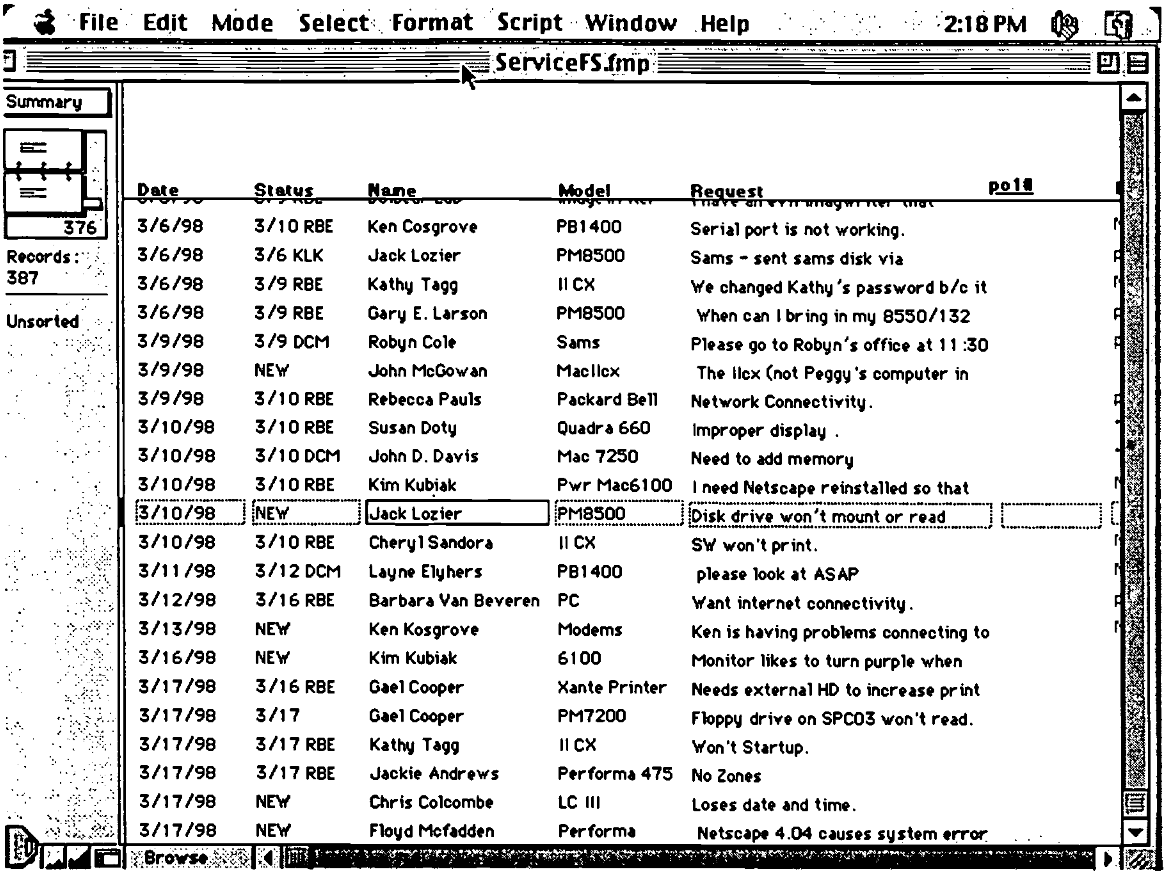


Figure 1. Picture of Help Desk Form on WWW.

Distributing support team jobs effectively

After information regarding the request has been entered into the database via the WWW, the support staff can prioritize and then distribute the jobs (see figure 2). The priority level should be based on guidelines determined by a computing advisory committee. For example, at Bethany most classroom technical problems take priority over all requests if the classroom problem inhibits learning, and most faculty academic computing requests take priority over staff requests.



File Edit Mode Select Format Script Window Help 2:18 PM

ServiceFS.fmp

Summary

376

Records: 387

Unsorted

Date	Status	Name	Model	Request
3/6/98	3/10 RBE	Ken Cosgrove	PB1400	Serial port is not working.
3/6/98	3/6 KLK	Jack Lozier	PM8500	Sams - sent sams disk via
3/6/98	3/9 RBE	Kathy Tagg	II CX	We changed Kathy's password b/c it
3/6/98	3/9 RBE	Gary E. Larson	PM8500	When can I bring in my 8550/132
3/9/98	3/9 DCM	Robyn Cole	Sams	Please go to Robyn's office at 11:30
3/9/98	NEW	John McGowan	MacIIcx	The IIcx (not Peggy's computer in
3/9/98	3/10 RBE	Rebecca Pauls	Packard Bell	Network Connectivity.
3/10/98	3/10 RBE	Susan Doty	Quadra 660	Improper display .
3/10/98	3/10 DCM	John D. Davis	Mac 7250	Need to add memory
3/10/98	3/10 RBE	Kim Kubiak	Pwr Mac6100	I need Netscape reinstalled so that
3/10/98	NEW	Jack Lozier	PM8500	Disk drive won't mount or read
3/10/98	3/10 RBE	Cheryl Sandora	II CX	SW won't print.
3/11/98	3/12 DCM	Layne Elyhers	PB1400	please look at ASAP
3/12/98	3/16 RBE	Barbara Van Beveren	PC	Want internet connectivity.
3/13/98	NEW	Ken Kosgrove	Modems	Ken is having problems connecting to
3/16/98	NEW	Kim Kubiak	6100	Monitor likes to turn purple when
3/17/98	3/16 RBE	Gael Cooper	Xante Printer	Needs external HD to increase print
3/17/98	3/17	Gael Cooper	PM7200	Floppy drive on SPC03 won't read.
3/17/98	3/17 RBE	Kathy Tagg	II CX	Won't Startup.
3/17/98	3/17 RBE	Jackie Andrews	Performa 475	No Zones
3/17/98	NEW	Chris Colcombe	LC III	Loses date and time.
3/17/98	NEW	Floyd Mcfadden	Performa	Netscape 4.04 causes system error

Browse

Figure 2. Job requests appear in the database when WWW request forms are submitted.

Performing jobs efficiently

Once the support staff prioritizes the jobs and sorts them, then they perform the jobs. The support staff contacts the client and arranges a meeting. The support team encourages the client to bring the hardware to the repair center. At this point the support staff can post information about the job to the WWW so that the client can find out the status of the request without having to contact the help desk (See figure 3). Such information includes: estimated time for job completion, start date, description of the work to be performed, and the type and status of parts ordered if it is a repair. The capabilities of the database should include easy access, password protection, and search functions.

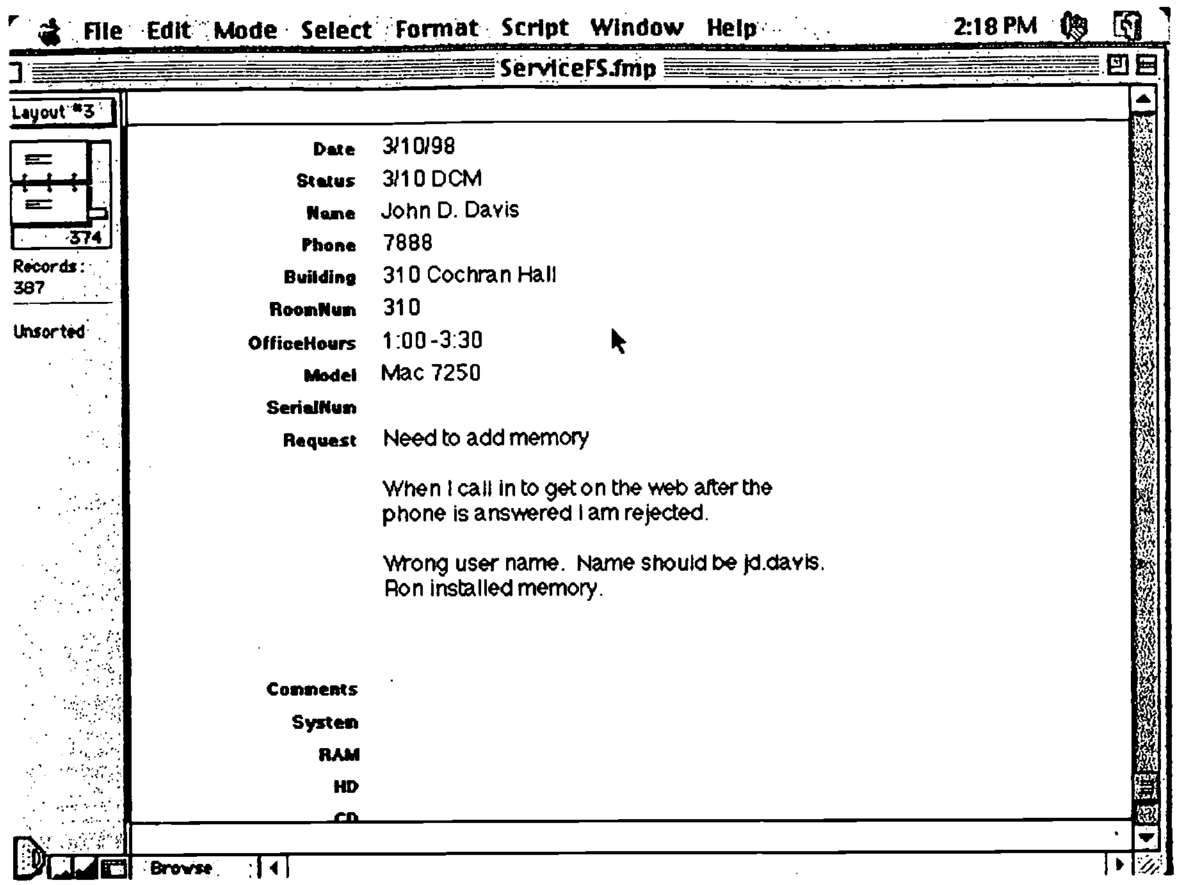


Figure 3. View of job request submitted via WWW form.

Training the support team

The support team that manages the desk needs to be well trained on both basic long term and short term computing problems. Help desk staff must be trained appropriately on how to answer phones, to talk a person through a technical problem, to solve typical problems, to find answers to information requests, and to distribute job requests to the professional support staff. Information on frequently asked questions should be provided via the WWW, so that the help desk staff and clients have quick answers to problems. Documentation about hardware and software on campus should be available at the help desk, as should diagrams of classroom configurations, along with a troubleshoot list of questions to ask the client. For example, a client might call and say that e-mail is not working. The support staff at Bethany College would ask, "Can you open the e-mail software?", "What message is returned when you retrieve your mail?", and "Can you see the network?" This initial information will help the staff identify and classify the type of problem the client is having.

Training faculty, staff and students

In addition to training the support staff, the academic computing staff needs to train the client base so that the level of sophistication of the questions increases and, possibly, the number of questions decreases. This training can take the form of "help-yourself" documentation and workshops. The "help-yourself" documentation should include easily accessible on-line manuals, brochures, videos, training materials available on the WWW, newsletters, an F.A.Q. bulletin, and information on standard hardware and software supported by the college.

In addition to "help-yourself" documentation, the academic computing staff should make workshops available for the clients. For instance, departmental workshops can be designed to meet each department's specific needs. This type of workshop succeeds best when the head of the department is involved because faculty tend to participate more when the head of the department shows an interest. The staff can also offer on-demand training when situations arise that call for immediate training. A third type of workshop, project-based group instruction, works especially well when a computer instructor wants to extend a task to different levels of skill. The group attends a series of workshops with the amount of necessary participation dependent upon the skill level with the common goal of completing a project at the end. Fourth, general workshops teach basic computer skills, such as word-processing. Of course, most clients prefer one-on-one training in which the support personnel helps clients with their personal computer. Although time-consuming, this is usually the most successful training method. With such a broad range of training responsibilities, the academic computing staff may need to request computer interest groups on campus to help take some of the pressure off the academic computer support staff. Likewise, more experienced clients can help less experienced clients at workshops and meetings.

Finding a time when most of the clients can attend workshops free of conflict is fairly difficult on a college campus. Invariably, several clients interested in attending workshops are unable to because of scheduling difficulties. Some colleges schedule open periods during the week to accommodate meetings. Consulting the Registrar for periods of lightest scheduling is useful. Scheduling training sessions during the lunch hour generally alleviates time conflicts. Varying the times of the training sessions if possible is a good idea, as is making certain they take up only one-hour time slots.

Making it easier for the support team to respond to clients

Organizing a help desk as outlined, the support team should find they can manage requests faster and more effectively. A well-managed help desk eliminates direct calls to the support staff, allows staff to stay on task, eliminates nuisance calls to technical staff, reduces the number of questions regarding non-standard equipment and, overtime, increases the sophistication of clients' requests. In addition, a well-managed help desk database can help staff identify the "caller of the week, " that is, the individual most obviously needing training. Similarly, the database can also identify the "problem of the week," thereby enabling the academic computing staff to address problems that might effect the entire campus community.

Making it easier for the clients to receive support

A well-managed help desk should also make it easier for clients to receive help. The help desk should provide "small town," personalized service like that offered in a friendly diner. Clients should feel that their request is important and will be successfully resolved within a reasonable time frame.

Making it work

When an institution plans to introduce a help desk the academic computing director and staff need to lay the groundwork adequately in order to ensure that the help desk gets off to a good start. Using the most common avenues of campus communication such as student newspapers, faculty and student electronic bulletin boards, the support staff must publicize where the help desk will be located, when it will commence service, and what its hours will be; the short and long term benefits of a help desk; and how to use the service. Ideally, the college should standardize hardware and software so that the support staff is not required to respond to questions that vary from "how to chat via the WWW" to "how to use the projector in the classroom." Help desk staff must remember, especially at the beginning, the client familiarity and knowledge may vary enormously, so shifting from simple to complex questions will probably be the norm until the college community's level of sophistication has been raised. Staff must remain courteous and never condescend; often excusing their own ignorance is very difficult for clients, especially to their peers. It is also essential that the help desk support staff understands the importance of timely response for job requests.

Regardless of how smoothly a system may run, technical crisis do occur and staff must plan for such contingencies. A help desk is not running well if staff do not respond quickly to a technical crisis. Giving pagers to key technical personnel can help ensure rapid response. Likewise, simply because a system seems to be running well is not a reason to ignore it. Periodically the staff should check how things are running. Providing surveys, such as a "Help Desk Quality Assurance Survey," will help the support staff understand how service is operating from the clients' perspective.

Conclusion

An academic computing help desk is one of the most important services provided to a college community. The clients must receive the best service possible so that their academic computing experiences are great and their experiences do not hinder them from exploring new areas in computing. In addition, the support staff must be well supported so that they can provide good support. Managing a help desk in the manner outlined should help to maintain a happy campus computing environment in which the needs of client and support staff are met.

References

"Troubled Times for Many IT Departments." *The Edutech Report*, October 1997, Vol. 13, Number 7, pg. 1, 3, 6 & 7.

Web Based Parallel Programming Workshop for Undergraduate Education

Mr. Robert L. Marcus, Computer Science
Mr. Douglass Robertson, Scientific Visualization Center
Central State University
Wilberforce, Ohio 45384

Abstract

Central State University has developed a web based workshop on high performance computing under a contract with Nichols Research Corporation (NRC) entitled "IBN SP2 Parallel Programming Workshop." The research is part of the DoD High Performance Computing Modernization Program (HPCMP). The research activities included developing techniques for converting classroom materials to web presentations, and algorithms in parallel programming techniques. Traditional classroom materials were prepared for web presentations using the Accelerated Web Page System in the Scientific Visualization Center which used an optical character reader to scan printed material and the Omnipage-Pro toolkit to produce HTML. The Internet Assistant software under MSOffice 97 was used for converting MS Word documents to web page files. The workshop material presents a series of sample programs on parallel programming using FORTRAN 90. It discusses topics on process synchronization, deadlock, data distribution, and load balancing. The workshop is designed for entrant level parallel programmers to make it suitable for undergraduate instructions and DoD applications. Basic algorithms on sorting, searching, statistical computations, numerical methods and linear systems are presented. The workshop is hosted on the web server at Central State University and the web server at Wright Patterson Air Force Base (WPAFB), Dayton, Ohio. The workshop was successfully field tested on undergraduate students at Central State University.

Introduction

Under the NRC contract Central State University was charged with developing web based education/training programs tailored for DoD high performance computing (HPC) users, other academic institutions and students. As such, Central State University developed the "IBM SP2 Parallel Programming Workshop." The programs were developed on a 128 node IBM SP2 system under a classroom grant from the *Ohio Supercomputer Center* (OSC). They were rehosted on the IBM SP2 system at the ASC/MSRC at WPAFB. The web version of the workshop is installed on the web server at CSU and WPAFB.

The workshop was designed for use in undergraduate education, or for entrant level HPC DoD programmers. Algorithms were selected from data structures, statistics, numerical methods, and linear systems. Consequently, the workshop is appropriate for (science and/or engineering) students at the sophomore level or above. Workshop pre-requisites are:

- FORTRAN 90 Programming
- An Introduction to UNIX Shell Scripts

The current versions of the programs in the workshop use task communication and synchronization constructs in the Message Passing Library (MPL). Conversions to MPI (an industry standard) is underway.

Accelerated Web Page Development System

The Center for Scientific Visualization developed the techno-scan system to reduce the amount of time needed to develop a web page. Some faculty in the department used Microsoft's Internet Assistant to develop web pages, but the quality of the HTML was not sufficient. The techno-system also reduced the amount of time to input graphics by using WYSIWYG interface. This rapid system relies heavily on OCR (Optical Character Recognition) technology and macros for its hard copy to HTML manipulation. Considerations are being given to using web tools such as FrontPage and Page Mil for developing a web site since most faculty now have suitable computing platforms for preparing text and graphical material for the web electronically.

Parallel Programming Design Issues

Three paradigms were emphasized:

apply Software Engineering principles using a modular design
use message passing only where necessary
give preference to group communication constructs

The first design paradigm is to apply traditional software engineering principles to develop programs with a modular design using functions and subroutines to separate communications from parallel computations. This facilitates optimizing the program to minimize time for communications, and maximize the time for parallel computations. Program designed to run in a distributed memory parallel environment, such as the IBM SP2, usually have three basic components that perform the following tasks:

- distribution of data using message passing
- execute local algorithms on nodes in the partition
- collect data from remote nodes using message passing

In the workshop programs, these components were named:

- distribute
- node_***application***
- collect

The term ***application*** is replaced by the name of the given application for the sample program (see the structure chart for each sample program). For example, the program ***mp_gaussm.f*** has a subroutine named node_***gauss*** which performs the Gaussian elimination algorithm on a partition (or group) of distributed nodes (or tasks). See the structure chart and data-flow diagram below.

The second design paradigm is to use one of the following data distribution methods, if the application design permits, as alternatives to message passing:

- loop parallelization, or
- parallel reads

These two methods will provide improved speedup of parallel processing over the use of message passing. Several workshop programs demonstrated that technique (the names of the programs ends with ***"lp.f")***. In other applications, collecting data may be inherent in the algorithm as demonstrated by the sorting example mp_sort.f. It uses a sort-merge algorithm. The merging process is designed so that the final merge of data occurs on task0, the master/control node, and there is no need for a distinct component to collect results.

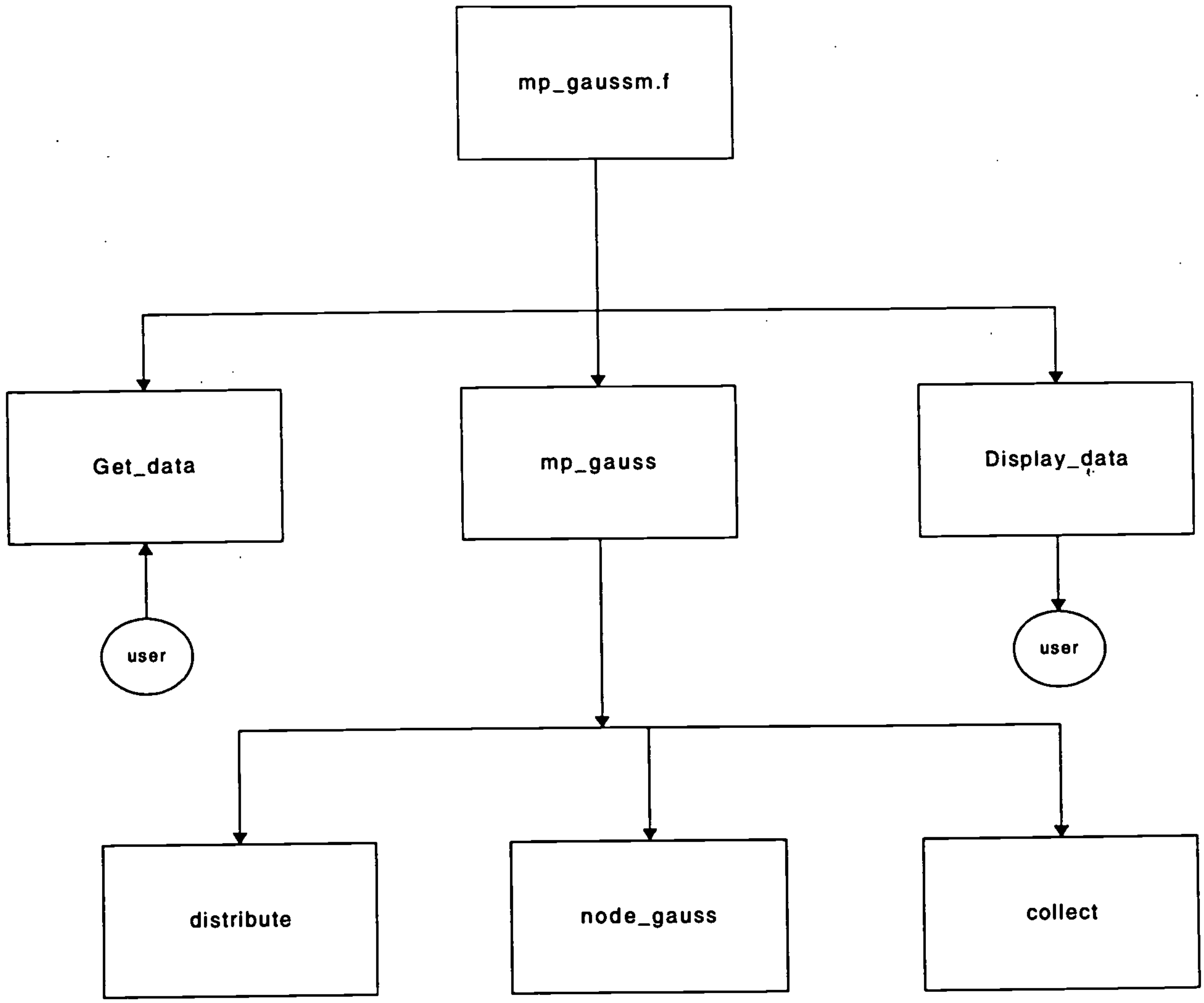
The third design paradigm is to use group communication constructs which simplifies program structure, improves its readability, enhances its maintainability. However, point-to-point communication constructs may be used to design tailored communication algorithms that are more efficient for a given application. Types of group message passing constructs are:

- point-to-group
- grout-to-point
- grout-to-group.

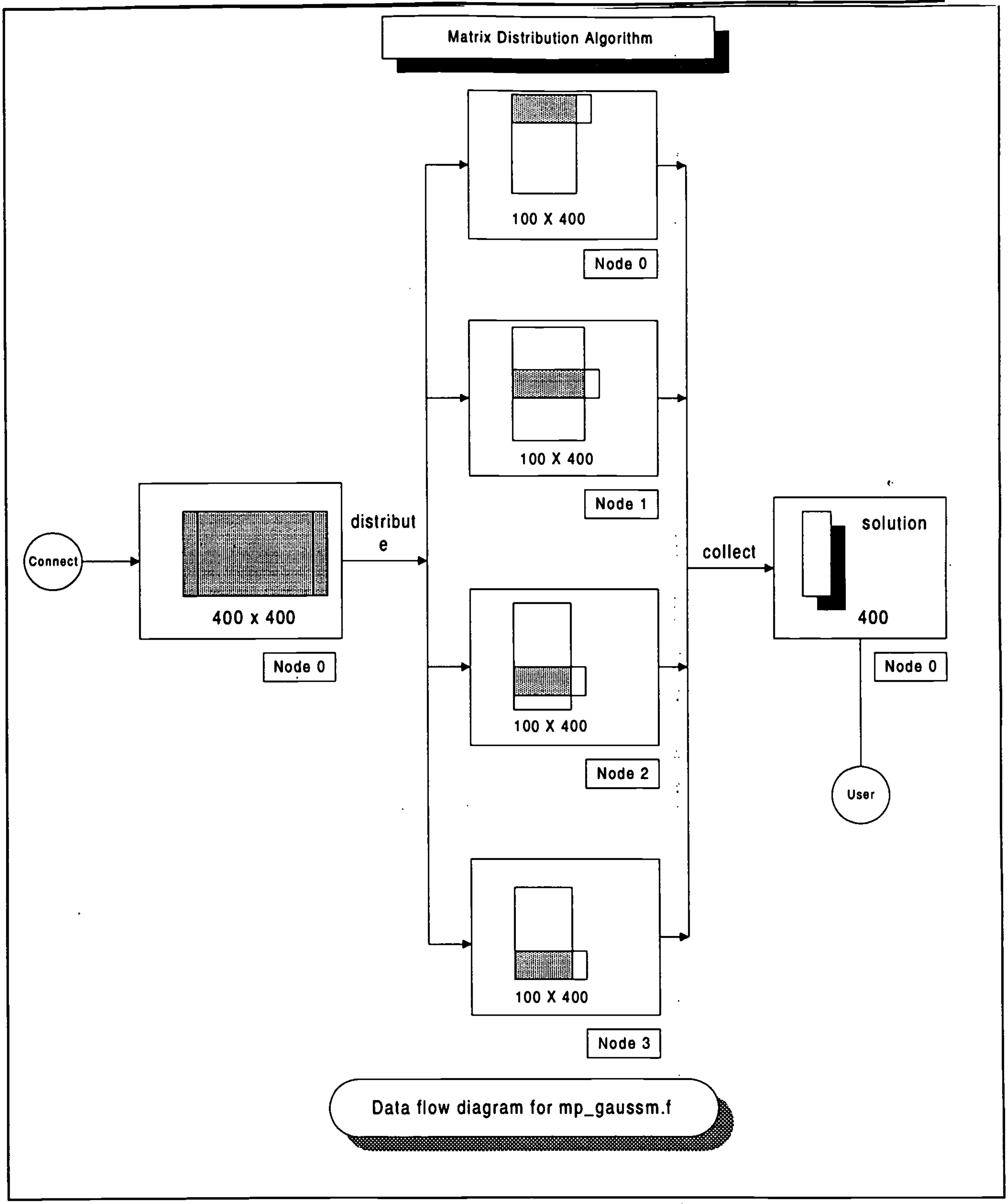
Figures and Diagrams:
Each program in the workshop included figures as the ones shown below for the program **mp_gaussm.f.**

Topic:	**Distributing a two dimensional matrix of data**
Program:	**mp_gaussm.f**
Purpose:	Solves an NxN linear system using a matrix distribution for the data and a Gauss-Jordan pivoting method.
Constructs:	mp_bcast --broadcasting mp_bsend -- blocking send mp_brecv -- blocking receive mp_bcast -- broadcast to group
Description:	This program distributes an NxN linear system of data to several tasks using a whole matrix distribution. The Gauss-Jordan pivoting method is used to pivot the system (a variation of matrix diagonalization) in parallel, leaving the solutions in column N+1 on each task. The solutions are passed to task0, put in standard order and then printed. The size of the linear system was 400x400.
Scalability:	YES, limited The maximum real time decreased for two tasks and increased when using three and four tasks.

Documentation Template: mp_gaussm.f



Structure Chart for mp_gaussm.f


Matrix Distribution Algorithm
Connect
400 x 400
Node 0
distribut
e
100 X 400
Node 0
100 X 400
Node 1
100 X 400
Node 2
100 X 400
Node 3
collect
solution
400
Node 0
User
Data flow diagram for mp_gaussm.f

Twelve Basic MPL Constructs

All workshop programs were implemented using commands from the following list:

1. Parallel environment commands:
   MP_ENVIRON, Returns the number of tasks in the partition and the caller's task id.
   MP_TASK_QUERY, Returns information about system variables and constants.
2. Point-to-point non-blocking message passing:
   MP_RECV, Posts a receive buffer for a message and returns without waiting for the message arrival.
   MP_SEND, Identifies a message to be sent and returns without waiting for the send to complete.
   MP_STATUS, Returns status of a non-blocking send or receive.
   MP_WAIT, Waits until a non-blocking send or receive has completed.
3. Group message passing:
   MP_BCAST, Distributes a message throughout a group.
   MP_GATHER - Concatenates a distinct message from each task to create a single message on task dest.
   MP_SCATTER, Distributes distinct messages to each task in the group.
4. Point-to-point blocking message passing:
   MP_BRECV, Receives a message and waits until the requested data arrives.
   MP_BSEND, Sends a message and waits until the ouput buffer can be reused.
5. Define a group within a partition
   MP_GROUP, Explicitly defines a task group.

Workshop Chapters

The following is a listing of the workshop chapters.

1.0 Introduction
2.0 Parallel Programming Design
3.0 Program Execution
4.0 Topics in Parallel Programming
  Communications
  Synchronization
  Deadlock
  Cache Efficiencies
5.0 Point-to-Point Message Passing
5.1 Sample Program mp_max1.f
6.0 Point-to-Group and Group-to-Point Message Passing
6.1 Sample Program mp_max2.f

6.2 Sample Program mp_stats.f
  Statistical Computations: Maximum, total, average, average deviation, standard deviation, and average square
7.0 Loop Parallelization

Chapter Descriptions

The first four chapters presented important background information on parallel programming. The other chapters presented sample programs on the topics indicated. Chapter 4 presented some introductory examples to demonstrate how programs are executed in a distributed parallel environment.

Chapters 5 presented the program, mp_max1.f, which computed the maximum value of an array of 400,003 elements. Only point-to-point message passing was used. All communication constructs were blocking. The program was not scaleable (see table below). Chapter 6 presented a modified version of mp_max1.f, named mp_max2.f, which used point-to-group and group-to-point message passing. Blocking and non-blocking constructs were used. Program mp_max2.f was not scaleable either. Chapter 6 presented a second program, mp_stats.f, which computed several statistical parameters shown above. The strategy was to perform more parallel computations to offset the time due to communications (group communications). The array contained 100,003 elements. The program did not show scalability.

Chapter 7 presented two programs that used the loop parallelization technique: mp_statslp.f and mp_integral.f. Program mp_statslp.f was a modification of program mp_stats.f to use loop parallelization was to define data on each node instead of message passing to distribute the data. The array size was 400,003. The program showed scalability up to 5 nodes. The second program, mp_integral.f, is an example from numerical methods. It uses the trapezoidal rule to compute an integral which has as its value pi=3.1415926. The algorithm used 4,000,000 subintervals that were summed in parallel. The program showed scalability up to 5 nodes.

Chapter 8 presented a technique to replicate the ureka command (on the Cray T3D) which sends a signal from a source node to other nodes in a partition. This technique was implemented in the program mp_find.f. The technique used a non-blocking receive from "any source" node to receive a special value in a variable. A while search loop was entered to find a key value in an array. The receive variable (or buffer) is used in the boolean expression of the while statement so that when (or if) a value is received, the while loop is terminated. The task which finds the key value sends the special value to all other tasks in the partition.

Chapter 9 presented the use of the mp_group construct to define a sub-group of tasks within a given partition. The program mp_sort.f demonstrates a parallel sorting algorithm:
Data is distributed to all nodes in the sub-group
Sorting is done in parallel on each node
Merging is done in parallel

The process for merging is to repeated pair-wise merge data in the top half of the sub-group with nodes in the bottom half until all data is merged on the initial task (task 0). The program ensures that the sub-group is a power of 2. There were 10,000 elements in the array. The program was scaleable up to 8 nodes.

Chapter 10-13 presented techniques for distributing two-dimensional data. Each program solved an NxN linear system using the Gauss-Jordan method. In Chapter 10 program mp_gaussc.f distributed columns of the matrix of data to each node. The method required several inter-communications during each step of the pivoting process. A 20x20 linear system was solved, but it was not scaleable.

Chapter 11 presented a technique (using the broadcast construct) to distribute the entire matrix to each node. Each step of the pivoting process required only one communication step: broadcast the pivot row. A 400x400 linear system was solved, showing scalability up to 4 nodes.

Chapter 12 presented a loop parallelization technique to define rows of data on each node for the pivoting process. Each step of the pivoting process required only one communication step: broadcast the pivot row. A 400x400 linear system was solved, showing scalability up to 9 nodes.

Chapter 12 presented a parallel read technique for reading rows of data into the matrix for each node. The data was stored in a direct access binary file. Each step of the pivoting process required only one communication step: broadcast the pivot row. A 400x400 linear system was solved, showing scalability up to 6 nodes.

Scalability and Speedup

The workshop material emphasize speed-up as the primary goal of parallel processing. If possible, the algorithms were improved until the program demonstrated scalability (i.e. speed-up greater than one) for a reasonable size partition of nodes. Speed-up was computed from the following formula:

$$\text{Speed-up} = \frac{\text{Wall clock time with a single task}}{\text{Wall clock time with more than one task}}$$

The wall clock time is the maximum "real time" (obtained from using the ***timex*** UNIX command) for the tasks in the partition. We consider the algorithm to be scaleable over when the speed-up factor is greater than one.

The table below shows speed-up factors for the workshop programs.

Scalability and Speed-up Listing

Program	Nodes	Maximum Real Time (sec)	Speed-up
mp_max1.f	1	0.5	-
	2	0.88	0.57
	4	5.87	0.09
mp_max2.f	1	0.44	-
	2	0.85	0.52
	4	1.28	0.34
mp_stats.f	1	0.33	-
	2	0.55	0.60
	4	0.56	0.59
mp_statslp.f	1	2.38	-
	2	1.49	1.60
	4	0.79	3.01
	5	0.66	3.61
mp_integral.f	1	3.14	-
	2	2.04	1.54
	3	1.25	2.51
	4	1.07	2.93
	5	1.45	2.17
mp_find.f	1	0.45	-
	2	0.57	0.79
	4	0.68	0.66
mp_sort.f	1	72.06	-
	2	19.59	3.68
	4	4.35	16.56
	8	2.77	26.01
mp_gaussc.f	1	0.30	-
	2	0.38	0.79
	3	0.48	0.63
	4	0.49	0.62
mp_gaussm.f	1	55.36	-
	2	32.26	1.72
	3	21.45	2.58
	4	16.60	3.33
mp_gausslp.f	1	36.56	-
	4	15.27	2.39
	6	10.58	3.46
	9	8.42	4.34
mp_gausspr.f	1	46.89	-
	2	29.24	1.60
	4	16.99	2.76
	6	11.59	4.05

The following programs:

mp_max1.f
mp_max2.f
mp_stats.f
mp_find.f
mp_gaussc.f

were not scaleable because the time saved from performing parallel computations did not offset the additional time needed for communications. The other programs:

mp_statslp.f
mp_integral.f
mp_sort.f
mp_gaussm.f
mp_gausslp.f
mp_gausspr.f

were scaleable.

Future Research Directions

Convert the programs to MPI.
Write C versions of all programs.
Write similar programs for a shared memory system (Origin 2000).
Implement a small scale application program to generate volume data for visualization purposes.

Conclusions

The workshop was used successfully to teach part of a short introductory course on parallel computing for undergraduate students. Some of the workshop programs were written by students who took the course: CPS 460 – Advance Topics, during the winter of 1997. The workshop used algorithms that were familiar to science and engineering underclassmen. Even though only 12 basic MPL constructs were used in the examples, the workshop included man pages on all MPL constructs for additional reference. Central State University continue these activities and expand the workshop to a 4 credit hour course on An Introduction to Parallel Computing.

References

[1] Analytical, Numerical, and Computational Methods for Science and Engineering, Hostetter, Santina and Montalvo, Prentice Hall, 1991

[2] An Introduction to IBM SP2 Programming, David J. Ennis, The Ohio Supercomputer Center, 1996

[3] CTC Virtual Workshop on Parallel Computing and Progarmming Languages, Cornell University, September, 1997

CNAV
A Unique Approach to a Web-Based College Information Navigator at Gettysburg College

Michael Martys, Vice Provost for Information Resources and Director of Computing
Don Redman, CNAV Programmer/Analyst
Alice Huff, CNAV/MIS Programmer/Analyst
Dave Czar, CNAV/MIS Programmer/Analyst and DBA
Pat Mullane, Director of Career Planning
Joseph Bennett, President, Perfect Order Inc.
Robert Getty, Consultant, Perfect Order Inc.
Gettysburg College
Gettysburg, PA 17325

Project Summary

In the fall of 1997, the Information Resources department at Gettysburg College deployed the CNAV (College Navigation) web tool. We created this web tool to allow our students, and the entire college community, the ability to better navigate through the college's curricular, co-curricular, and extra-curricular offerings, allowing the students to enhance their college experience as a result. The tool was designed to help the college community discover what Gettysburg had to offer and more importantly, discover new connections between those offerings.

Our original effort to solve this problem (which started somewhere around the summer of 1995) took the form of a typical group of web pages - many pages describing courses, events, extra-curricular activities, professors, etc. As this project grew in size, we began to contemplate the particular page structure needed to facilitate the discovery process. We decided to talk to students about the structure; our thinking was that if we could understand our students' thinking as they wandered through a web hierarchy, then we would be able to derive a structure that worked well for them.

The results of our discussions caused us to re-evaluate our approach to this problem. We discovered that the students' thinking was quite varied and dealt with questions we did not expect. Statements such as "I am bored" or "I am unhappy with what I am currently doing" were common. Students complained about discovering campus offerings too late to take advantage of them. When they found something that interested them (such as a lecture) they had difficulty finding other things (such as courses) that were similar to that lecture.

As a result of those discussions, we radically altered our approach to this problem. In our minds, the solution was to create a system that was interactive and focused as much on the student as it did on the college's curricular, co-curricular, and extra-curricular offerings. Rather than trying to make a static web hierarchy, we decided to make a "smart" web tool that would make it easier for students and others to find connections between themselves and other people, courses, or events on campus. Our smart web tool took the form of a Netscape secure web server, an Oracle database, Excite search engine, data exchanges between our system and other computers on campus, and approximately 700 cgi's and 490 database tables.

CNAV is unique because, rather than treating the web as a series of static pages, it treats the web as a series of "puzzle pieces" that can be assembled into different views. Courses, extra curricular offerings, events, information about groups of people, and individuals themselves are treated as puzzle pieces inside of CNAV. The unique attributes of each of these items can be thought of as the "edges" of the puzzle pieces. CNAV allows each user to describe those "edges" with as much detail as possible. For example, a student who is sponsoring an event can associate a web page with the event, a text description, indicate who might be interested in the event, list the event sponsor, who to contact for additional information, as well as the more mechanical details of where and when the event is to take place. A faculty member who teaches a course can associate a course homepage, syllabi, readings list, faculty web page with the course; add their own course description to the official description; indicate who might be interested in the course; and describe what skills, competencies and what academic themes the course covers. CNAV will use all of this information for the "description" of the course and CNAV allows totally unique information for each course section, each semester, and each year that it is taught.

From the CNAV user's perspective, how they view the "pieces" can vary depending upon what the user is interested in finding. In addition, once a CNAV user finds a puzzle piece (say, an event) that is interesting, he or she can then use the piece to find other pieces like it (such as a course). Because of this capability, truly unique ways of searching are possible. For example, in CNAV, it is possible for a "course" to search for "students", "a campus event" or find similar "courses".

CNAV remembers the particular interests (attributes) of every campus member and uses this information to draw the campus member's attention to "pieces" that the individual would find interesting. CNAV also communicates with other computers on campus (most importantly, our legacy MIS system) to acquire as much knowledge about the individual as possible. Because of this feature, it also acts as a robust student information system, providing web access to financial, academic, and other college information for each individual on campus.

CNAV takes on different personalities for different groups of individuals on campus. Each personality is optimized for that group - for example, faculty members get tools that allow them to edit course information, access their advisees' information, access budget information, etc. Advisors get tools that allow them to track appointments, access student records, monitor mid-term deficiencies. Students get tools that provide access to campus events, course catalog, electronic reserves for their courses, personal budget information, and the courses they are taking this semester.

CNAV is the first software program deployed at Gettysburg intended to be used by the entire campus. Approximately 76% of the entire campus (faculty, staff and students) have used CNAV, and approximately, 50% of the students and faculty use it frequently. Considering that CNAV was first available to the entire campus the fall of 1997, we are very encouraged with these results.

One of CNAV's unexpected benefits is that our campus community is now making the cultural change necessary to deal with on line information tools. Faculty members who would never deal with a web application are now happily using CNAV.

Another benefit is that since CNAV provides a secure, confidential web engine, and since it already contains our legacy data, it allows all sorts of follow-on projects. Some of these new initiatives are support for prospective students (which is nearing completion), a student portfolio tool (which is nearing completion), alumni-student mentoring (due August 1998), and on line course evaluations. CNAV has also forced us to face the realities of electronic privacy and intellectual property issues.

CNAV was programmed in house and required approximately 4.5 FTE years of programming talent. Many offices from the campus were involved and continue to be involved in different aspects of its design, including admissions, academic advising, public relations, career advising, and the faculty as a whole.

Application Description

CNAV currently runs on a Sun Sparcserver 3000 with two processors and 512 MB of memory. It uses an Oracle database, running version 7.3.3, and uses a Netscape Commerce Server. We currently run all connections to CNAV encrypted using a 40-bit key; thus allowing us to use encrypted connections overseas. CNAV is comprised of approximately 700 PERL scripts and 490 database tables.
A customized Excite search engine provides keyword searching on courses, people, and groups. The customizations allow the user to refer back to the course, people or help database instead of accessing random web pages, and if the user is searching for people, the customizations allow him or her to build an email alias "on the fly" using the results of the search.

CNAV makes heavy use of web pages to describe things. Courses, people, events, and groups can all be described by associating a web page with that item. Rather than have users create web pages on CNAV itself, CNAV allows users to put the web pages anywhere, and simply indicate to CNAV where they are. This approach lets people use their favorite web-page creation tools as well as to store their web pages in whatever web directory space is most convenient. CNAV assembles these pages so that a user does not notice that the pages are on other servers. CNAV uses web crawlers to load these pages into its Excite search engines.

CNAV communicates with three other computer systems at Gettysburg on a nightly basis. These are our legacy MIS (to obtain all people, course, prospective student, and alumni information), our student ID computer system (for photos of everyone on campus), and our campus events computer (to obtain information on upcoming and past campus events). Using MIS data, CNAV automatically creates accounts for new campus community members. Passwords for these new accounts are distributed to the appropriate people on campus via another CNAV tool.

CNAV has a context sensitive on line help system. With the entire campus being able to use CNAV, deploying a program of this size is logistically difficult. Thus the help system has to be capable of playing a major role in teaching people how to use CNAV. Because we wanted this help system to have a student's perspective, primarily student workers wrote it. Help screen content changes based on where you are in the system and who you are (faculty, staff or student), thus improving the effectiveness of the help screens.

A typical user interacts with CNAV by first using a secure (https) web address to access CNAV. After a user logs on, all communications with CNAV are performed in a secure (encrypted) manner, thus allowing CNAV to share confidential information with the user anywhere in the world.

Depending upon who the user is, CNAV will bring up the appropriate tools for that user. These tools are broken down into three basic groups - faculty, student and staff. The section below outlines some of the capabilities of these tools.

What an Individual Can Do in CNAV

We have found over the last couple of years that CNAV is a hard tool to describe because it handles so many different tasks. The best way to describe CNAV is through a demonstration of its abilities. The second best method is to list some of CNAV's functionality, which-given the large number of things it can do-is somewhat difficult. This is a partial list:

<u>Course information</u>: CNAV maintains a section and catalog level course catalog on line. All administrative data is available for all courses (enrollment, class roster, description, previous years and semesters the course was offered, etc). The catalog is historical, with the last 10 years of course section history on line. For example, it is possible in CNAV to bring up the Math 101 Spring 1996 section A, Math 101 Fall 1994 section D, as well as the current (Spring 1998) course catalog entry for Math 101. The historical section information is very useful for a small liberal arts college such as Gettysburg since many courses are taught on 2 or 3 year cycles, many sections of a course are taught differently, and finding a description of the last course taught might mean looking back 3 years. It also is very useful for seminar and colloquy type courses that are only taught only once or twice.

<u>Course information-editing tools</u>: CNAV provides faculty special tools so they can edit the information for the courses that they teach. The editing tools allow them to attach course home pages, web page syllabi, web page reading lists, and faculty home pages. Faculty can also link interests to the course (including course themes, skills, and competencies), as well as create their own enhanced version of the course description. The linked web pages can be anywhere in the world. CNAV uses a web crawler to "pull" these pages into the CNAV course search engines.

<u>Electronic Reserves</u>: Electronic reserves are provided as part of the course catalog information, as well as part of a student or faculty member's "this semester" view. CNAV monitors and enforces course membership restrictions in order to remain compliant with copyright restrictions. CNAV also tracks access to these reserve materials on an individual access basis to enable proper payment of copyright fees.

<u>Communication</u>: Students and faculty can see who is in their classes, including rosters for the future semesters if they have pre-registered. They can easily access all the biographic information of fellow students, as well as their photos, interest summary, major summary, etc. Faculty has quick email access to the faculty advisor(s) of each student in their class.

<u>On line Course Deficiency Tool</u>: CNAV provides faculty with the ability to submit on line deficiencies, as well as "excellent" performance reports. Each deficiency is routed via email to the student's academic and faculty advisors, as well as the student. The student CNAV screen has special "alerts" that appear when a student has a deficiency outstanding, and the faculty advisor CNAV screen will see the same special "alert" on their "list my advisees" CNAV screen.

Privacy: There is a detailed privacy subsystem that allows each user to carefully control who on campus can see each item of their "public" information such as their photo, campus address, campus phone, home address, home phone, class schedule, etc. There is a separate level of access control for students, campus employees and the outside world. Each data item (photo, home address, phone number, campus address, etc) is individually controlled. For example, it is possible for a student to allow CNAV to display their photo, campus phone number, and campus address to fellow students, but to only display their campus phone number to people outside of the college.

Interest Tool: CNAV allows the faculty and advisors to create a list of interests (perhaps they would be better called attributes) that can be used to describe campus events, courses, individuals, and groups of people. The reason for using the term "interests" was that when CNAV was first being created we were interested in describing "what was interesting" about each item or puzzle piece in CNAV. These interests are be grouped into categories, which then can have subcategories, which then have a list of "interests".

For example, presently in CNAV, the Gettysburg faculty and staff have created the categories: "Course Themes", "Options for Involvement", "Personal Interests", "Things you like to do", "Skills & Competencies", "Things I Care About", "Things to Do, Places to See", "Tools for Survival and Success". The category "Course Themes", has the subcategories: "Ethnic/Cultural Studies", "Historical Studies", "Modern Society" and " Science & Technology". "Historical Studies" has the interests: "Victorian Studies", "Renaissance", "American Studies" and several others.

The reason for having a tool that can create such a complex and rich interest "language" is that when we first started working on the CNAV project we discovered that different groups of the campus used different terms to describe things. Faculty would use one set of terminology to describe courses, and students would use another set of terminology to describe what they were interested in studying. In order to do the matching that we had hoped to do, we needed to provide some sort of common language.

Interest Matching Tool: CNAV has special tools that allows CNAV to find connections between things based on how similar those items are, or based on what particular interest (or attributes) of an item you are interested in. This is a very unique and powerful tool in CNAV, because this tool that allows you to take one puzzle piece and find another like it. For example, it is possible to attend a campus event, then to use CNAV to find courses that are similar to that event, or to find professors that have interests similar to that campus event. It is possible to have courses find students, students to find other students they have a lot in common with, for clubs to find students who share similar interests and so on.

Personal Information: All community members can update their personal information (addresses, name, billing address, etc) via CNAV. They can also inform CNAV where their homepage is, where their resume (or vita in the case of faculty and staff) web page is, as well as their web portfolio. CNAV uses a web crawler to "pull" those pages into the CNAV search engines. User's can also inform CNAV of their personal interests. CNAV uses these interests to point out campus events and URL's that might be of interest to the user. Users have the ability to "hide" their interests from other users of CNAV. Thus, a student who indicates that they have an interest in Math can hide that fact from the rest of the campus.

On line Transcripts: Students have access to their transcripts on line. Each transcript is hyperlinked so that each course refers back to the historical course section level catalog entry. It is possible for students to see who was in each of their classes, including photos. Historical web data is maintained for each course, as well as instructor data. The transcript can be displayed in department, division, or semester order to help with advising issues. At Gettysburg, all faculty members are considered advisors, and those faculty and other campus advisors who are not the primary advisor of a student can have access to a student's transcript if justification is given. CNAV enforces transcript access restrictions and provides logging tools that allow the registrar to monitor to the student's records.

Surveys: CNAV has a built-in survey tool that allows on line surveys to be created and administered to various campus constituencies. Confidentiality and authentication are provided along with a fine control of who takes the survey. For example, it is possible to survey "all female biology majors", "all residents of a particular building", or "all members of a particular course (including optional, last semester, next semester, and this semester subgroups)." Results can be obtained instantaneously. Anonymous-response surveys are allowed. CNAV has the ability to create an instant email alias out of the group of people who gave a particular answer to a question on the survey.

Besides being used for general-purpose surveys, this tool was created with the idea that there were times when students could not discover things on their own without a little help. The survey tool would allow someone to ask questions of groups of students, and then based on their answers, target specific responses to each group. For example, it would be possible in CNAV for the internship office to survey all of the sophomores about their summer plans. For those students that answered that they did not have any plans, CNAV would allow the internship office to send email to those students alerting them to the various internship and summer job opportunities on available on campus. For those students that indicated that they did have plans, CNAV would allow the internship office to ask how many of those students are planning internships.

Campus Events: CNAV has a series of campus event tools that allows each member of the campus to submit campus events. Each event can be described in detail, and a link can be made to the event's web page. CNAV maintains an RSVP list for each event, and allows the event owner to have access to an instant email alias of all users who have RSVP'd. Each event can be tagged with interests; CNAV then uses these interests to draw the attention of other CNAV users who have expressed similar interests to this event. There are special tools that allow for the editing of a previously submitted event by the event owner, requests for room reservations, and search tools that allow for searching for events during certain periods of time or based on interests. CNAV communicates with the main campus room reservation computer system so that it knows of all currently scheduled future events.

This campus event system will soon be made public on the Gettysburg Web Page (due late April 1998). Campus privacy issues needed to be first addressed; as a result, a modification needed to be created that would allow specially tagged events to be blocked from off-campus view.

Campus-wide Bookmarks: CNAV has a series of campus-wide bookmark tools that allows each member of the campus to submit URL's to the global URL database. This database can be thought of as a set of "bookmarks" that the entire campus shares. What makes these bookmarks unique is that each URL can be tagged with interests, which CNAV then uses to draw the attention of other CNAV users who have expressed similar interests. CNAV keeps track of who submitted each URL, as well as the date, so that a user's name can be used as selection criteria as well.

Financial Reports: CNAV gives users access to their financial reports. Students can see their bookstore bill, current college tuition statement, phone bill, and traffic ticket bill on line. Faculty and staff can access any departmental budget reports that they normally would receive in paper form. Reports are updated daily.

Email aliases on the Fly: CNAV allows users to create instant email aliases as a result of almost any search or query. For example, it is possible in CNAV to send email to all faculty who like banjos, all members of a floor in a campus building, all biology minors, all members of Sociology 101 section "A", all members of the 1997 football team, all people who like classical music, etc.

Fast Email Access to Student Advisors: CNAV provides all faculty and staff with fast email access to all advisors of any student. This allows an instructor to quickly contact the advisor of any student who is having problems in his or her class. All faculty have direct access to their advisees, including photos, student records, deficiency alerts, and a student planner.

Keyword and multiple condition search tools: CNAV has two types of search tools - keyword, which is based on the Excite search engine, and multiple condition, which is a database search. Keyword search tools exist for searching through faculty, the campus community, and all 10 years of course data. Each tool allows the creation of an instant email alias based on the search results. The search engines make use all information about each item (including web pages) when performing a search. Multiple condition searches allow very specific searches that are based on certain criteria. For example, in CNAV it is possible to find all students living on a particular floor in a particular building; to find all courses taught in the last 10 years by a particular professor; to find all English courses with available seats on Tuesdays at 10 am, and so on.

Group tools: CNAV has a series of tools that allows groups on campus to identify themselves. Each group can create its own email alias, assign members, describe itself, point to its own web page, and assign interests to itself. Each group can subsequently be edited and maintained by the group's creators. CNAV uses the group's interests to find courses, events, individuals and other groups that have similar interests. Each group can allow people to join automatically or join after approval of the group creator. CNAV users can search the list of groups to find groups that would interest them. This tool was created from the idea that there were many groups of students on campus doing interesting things, but hardly anyone knew about them. Many of these groups are unofficial, meaning they do not have formal recognition by the college. The idea was that given a chance to identify themselves, they would do so. In practice, this is turning out to be true.

Customizable: CNAV uniquely draws every CNAV web screen based on that particular user's preferences. This includes the ability to choose between three different menu systems, the ability to customize what screen appears first for them, and the ability to ask CNAV to filter things based on how well those things match the users chosen interests. For example, in CNAV it is possible to have CNAV only display campus events that have a very high degree of correlation with your personal interests and hide those from view that do not.

Appointment Tracking Tools: This series of tools is used by administrative advising offices, such as career planning, to help track student interactions with the office and share that information with the student. The office creates an appointment record upon completion of the meeting and enters any pertinent notes and comments as well as assigning category and outcome codes to the appointment. The student can pull up their own records and review their appointments and notes.

Access logs: CNAV has special tools to monitor and log accesses to student transcripts by various Gettysburg student advisors. The registrar uses this tool to monitor accesses to transcript to ensure campus policy is followed with regards to transcripts. CNAV also monitors the use of electronic reserves, tracking each individual access for each document in each course. This ensures copyright compliance and allows us to pay the appropriate copyright usage fees.

30,000-foot views: CNAV has several special tools that allow high level overview of various data elements in CNAV. For example, CNAV can display a high level overview of the current CNAV course catalog. It can indicate which courses in CNAV (and which sections) have home pages, faculty web pages, syllabi, reading lists, interests, and enhanced descriptions. This information is available for all departments, and for each semester and year. Thus it is possible at a quick glance to see how much material is available for each department, and how it changes over time. This is very useful for targeting training and assistance initiatives.

Looking to the Future

CNAV has been well received by our campus community. As a result, they want new features and capabilities. Three new modules are currently under development. Each module is being developed in coordination with the appropriate departments. A brief description follows:

Admissions: A new CNAV module is being built (currently in beta; formal release March 23, 1998) that will allow all accepted students to have: on line access to their admissions materials; to submit any college forms electronically (and be able to edit them in subsequent visits); to easily contact everyone on campus with whom they have had previous contact (including photos). For each one of their interests (for example, lacrosse and biology), the designated campus contact (again with photo) will be provided.

Alumni: A new CNAV module is being built (beta release due August 1998) that will allow all alumni to have the same CNAV capabilities as all campus members. Special alumni tools will allow them to contact other alumni (by various criteria, such as geographical region, sport, year of graduation, campus activities, etc). The primary purpose of this module is to foster student-alumni interaction and mentoring. As such, special tools will be provided which will allow alumni to search for and contact students who share academic, career, or personal interests, and likewise for students to search for alumni. Alumni will have access to their personal information (addresses, memberships, event registration, giving record), and recent alumni will also have scholastic information such as an online transcript. Alumni photos will be scanned from previous yearbooks and alumni will be able to update their photo's online. An extension of CNAV's privacy tools will allow Alumni to control who (students, faculty, employees, or other alumni) has access to their information.

Advising: A new CNAV module is being built (currently in beta, formal release August 1998) that will allow students to manage a portfolio. The portfolio will allow students to plan and list academic, co-curricular, and extra-curricular activities, allow and facilitate reflection on these activities, and provide display tools to allow different views of the portfolio. The goal of this module is a more holistic approach to the college experience; with the help of CNAV, students will be able to see the connections between all aspects of their years at Gettysburg.

Major Contributors:

Michael Martys, Vice Provost for Information Resources and Director of Computing
Mike is the project leader for CNAV, guides the future direction, provides technical advice and direction, and acts as the primary liaison to other campus groups who contribute ideas and concepts to CNAV.

Don Redman, CNAV Programmer/Analyst
Don is the primary "front end" web programmer for writing the majority of the code that is used to create the custom web screens. Don also works with departments, conceptualizes new functions, and provided the primary "bug" support for the end users of CNAV.

Alice Huff, CNAV/MIS Programmer/Analyst
Alice acts as the primary end-user support person CNAV, monitors CNAV security, performs specialized CNAV training and troubleshooting, and acts as the CNAV evangelist.

Dave Czar, CNAV/MIS Programmer/Analyst and DBA
Dave is the college expert for our Oracle databases, the college expert on the legacy systems and data content, and is responsible for the communications of CNAV with other information systems on campus.

Pat Mullane, Director of Career Planning
Pat's acted as a campus representative to the CNAV programming team, bringing specialized expertise in the areas of advising, user interface, and end-user. Pat also acted as an evangelist and a liaison with many of the campus departments.

Joseph Bennett, President, Perfect Order Inc.
Joe is a consultant from a consulting firm called Perfect Order Inc. Joe primarily acted as the CNAV's database expert. Joe also provided technical advice direction for the CNAV team, conceptualized many CNAV functions, and did some of the front-end programming. Joe left the project at the end of 1997.

Robert Getty, Consultant, Perfect Order Inc.
Bob took over from Joe as a consultant with Perfect Order Inc. Bob presently is doing database work and front-end programming.

Using Disk Imaging Software and Network Distribution to Automate Installation and Maintenance in Computer Labs

Aaron Minnick
Franklin College
Franklin, IN 46131
minnica@franklincoll.edu

Abstract

Computer labs in small undergraduate institutions are obliged to meet a wide variety of needs. One lab may serve variously as an undergraduate classroom, a general computer lab, a multimedia production lab, a foreign language lab, community training center, a desktop publishing production facility, and a professional and administrative training facility. To meet these needs, labs must be flexible and reliable, must perform consistently, and must be kept current with bug fixes, service packs, and new software releases. Given the limited resources of small institutions' IS departments, traditional methods of manual installation, troubleshooting, and maintenance on individual workstations quickly become inadequate. This presentation will demonstrate the use of disk imaging software and network distribution to automate setup, updates, and troubleshooting in computer labs.

(Aaron's paper was not available at the time the Proceedings went to press. He will supply copies of his paper at his talk.)

Web Management of Student Information

David Moldoff
Applied Business Technologies, Inc
4631 West Chester Pike
Newton Square, PA 19073
(610) 359-0700
moldoff@abtcampus.com

Abstract

Using Microsoft's new Visual InterDev development tool, ABT has developed two new web products that extend student and faculty access to administrative information. IQ.Student allows students to take a more proactive role in their own affairs, allowing them to view their account balances, grades, courses and unofficial transcripts, and to register for classes. This product also allows prospects to view course catalogues and access other on-line applications. IQ.Faculty allows faculty to enter grades, access student histories and profile sheets and update student records. Both these products run on either PC's or MAC's, requiring only internet browser software such as Netscape Navigator or Microsoft Internet Explorer. The presentation will include a product demo.

(David's paper was not available at the time the Proceedings went to press. He will supply copies of his paper at his talk.)

TQM in a Computer Lab

Dewey A. Swanson
Assistant Professor Computer Technology
(812) 348-7238
dswanson@iquest.net
Julie A. Phillips
Assistant Professor Organizational Leadership and Supervision
(812) 348-7207
jphillip@iupui.edu
Purdue University School of Technology
4601 Central Avenue
Columbus, IN 47203

Introduction:

TQM is a philosophy that can be applied to any aspect of work or life. With this philosophy and a set of quality tools, one can transform the way work is done to meet customer needs, besides delighting the customer. At Purdue University School of Technology the Columbus Campus, we used the TQM philosophy in our computer lab to try and better met our customer (students) needs. We conducted a customer satisfaction survey, compiled the data, and implemented PDCA. Since we are currently involved in this endeavor, we still need to measure the results and continuously improve.

Background of School:

Purdue University School of Technology at Columbus Indiana (PST) is a unique partnership between education and business, industry, and government. It was established, along with eight other regional campuses, to meet Indiana's need for trained technologists and technicians. The curriculum at the Columbus campus is the same as Purdue University at West Lafayette, Indiana.

PST is located at the Indiana University Purdue University Columbus (IUPUC) campus. IUPUC is part of the Indiana University regional system. PST partners with IUPUC to offer eight-degree programs. There are 1900 students at this campus, of which over 300 are PST students.

Background of Computer Personnel:

PST maintains two computer labs that are available to PST students, as well as, IUPUC students. PST has a Lab Manager, along with students hired as Lab Assistants to staff the labs. The Lab Manager duties include: hiring, training, scheduling, installing new hardware and software, troubleshooting hardware and software problems, fixing problems, managing PST lab file server, writing and posting lab related documentation (both user and employee-based), and maintaining the lab environment.

The Lab Assistants duties include: providing assistance to lab users, answering questions, trouble shooting hardware and software problems, assisting in hardware and software installation. Lab Assistants must have taken an introductory computer classes with basics of Windows and Microsoft Office toolset, or has equivalent knowledge. They must also posses good interpersonal skills and be willing to work set hours. Each semester there are approximately 6-9 Lab Assistants depending on their availability.

Background of Computer Lab:

The two PST computer labs include 42 PCs and one printer. Of the 42 PCs, 37 are 486 machines and 5 are Pentiums. The printer is a laser printer that is 4 years old. IUPUC provides one lab with a staff that maintains the labs, which contains 15 PCs and one printer. The labs are in three physically adjacent rooms, with windows in between the labs. This allows one Lab Assistant to cover all three labs.

TQM Approach:

The Sample

A sample of the students at the IUPUC campus was surveyed, concerning customer satisfaction with the PST labs. The surveys were distributed to students in different classes from different programs. A representative sample was from Computer Technology, Organizational Leadership and Supervision, Nursing, Mathematics, and Accounting. The survey was given during the summer and fall 1997 sessions. There were approximately 100 respondents, which represents about six percent of the campus population.

The Survey

The survey followed the Likert-Type format. It contained 22 questions that the customers were to rank from 1 to 5 whether they strongly disagreed, disagreed, undecided, agreed, or strongly agreed, respectively, to the statements. The questions were broken into categories: lab facilities, lab assistants, and hardware/software. Other sections of the survey included: demographics, general comments, and suggestions and improvements to the process. (See attachment)

The Results

Following is a summary of the demographic results of the survey:

Thirty-five percent of the students surveyed were Computer Technology majors. Organizational Leadership and Supervision represented the next largest group at twenty-five percent. The rest distributed among twenty majors. Eighty-eight percent of the students are a junior or less and thirty-four percent are sophomores. Most of the students considered themselves novices with PCs. Two-thirds have a home computer. Slightly over half of the respondents use the lab one or more times a week with sixty-five percent saying that they use the lab generally on Monday to Thursday between 8:00am. -4:30pm.
(See Figures 1-5).

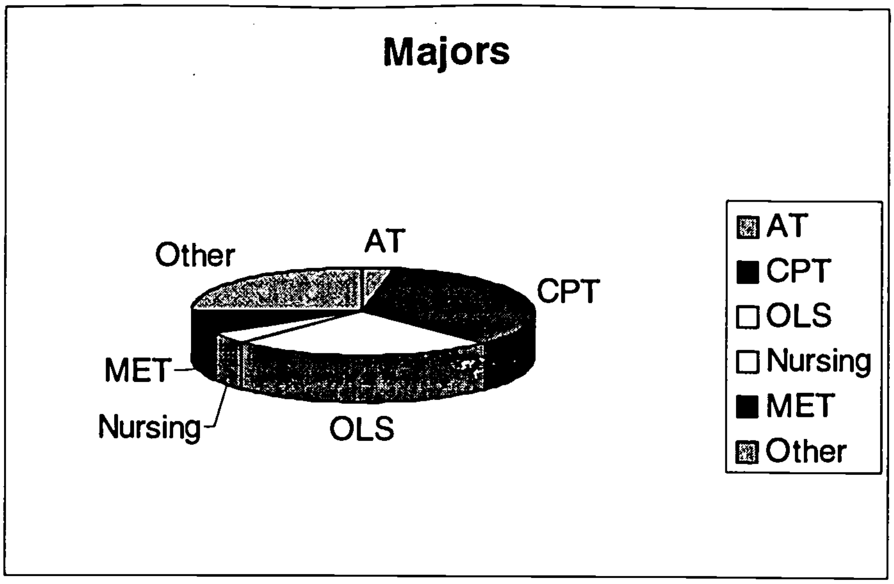


Figure 1

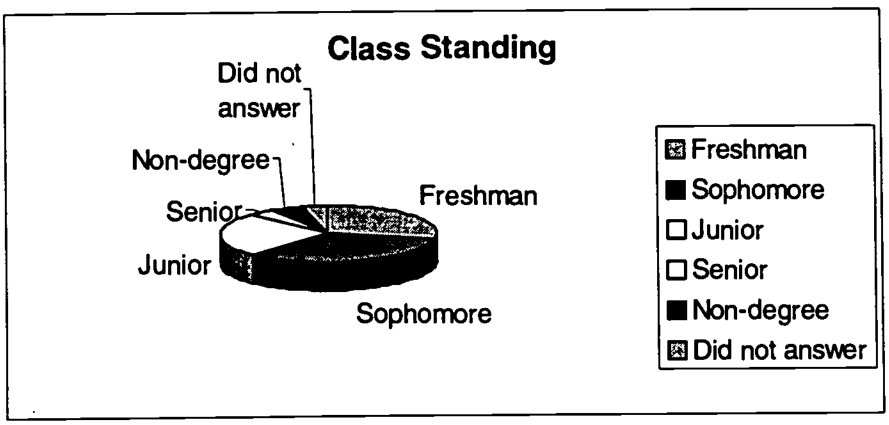


Figure 2

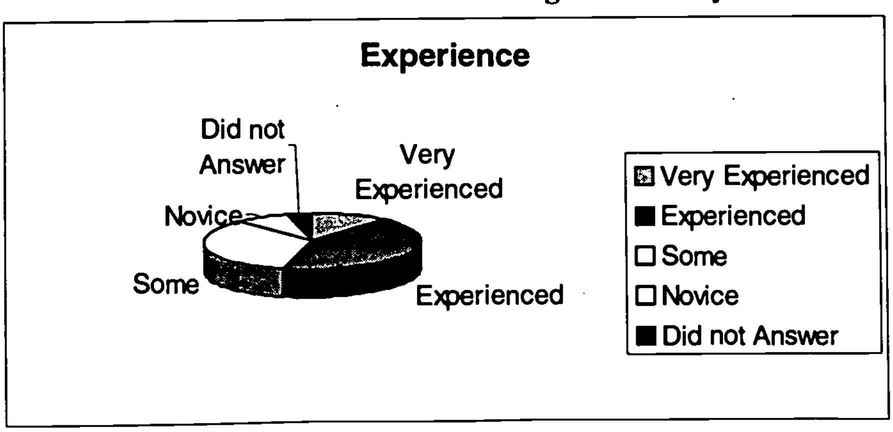


Figure 3

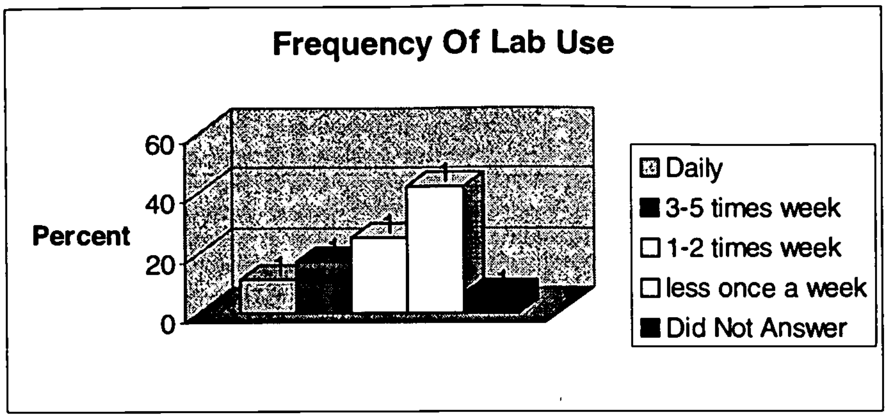


Figure 4

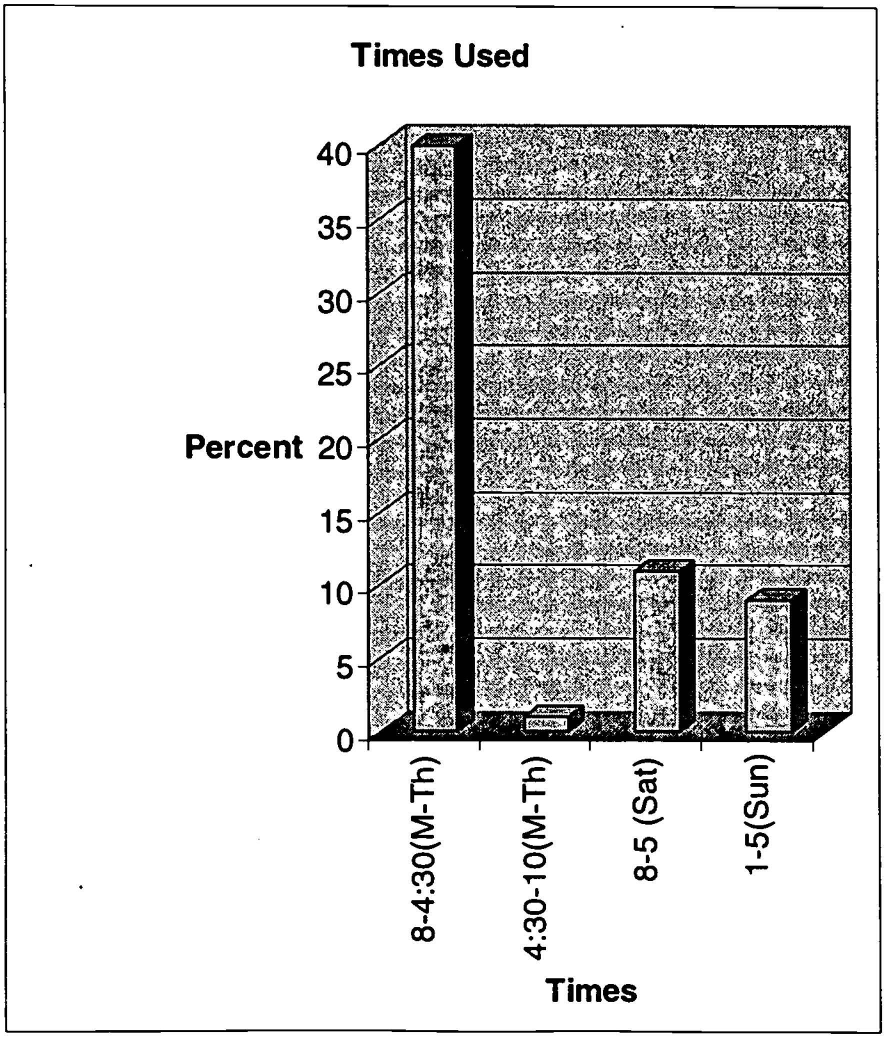


Figure 5

Following is a summary of the question results of the survey:

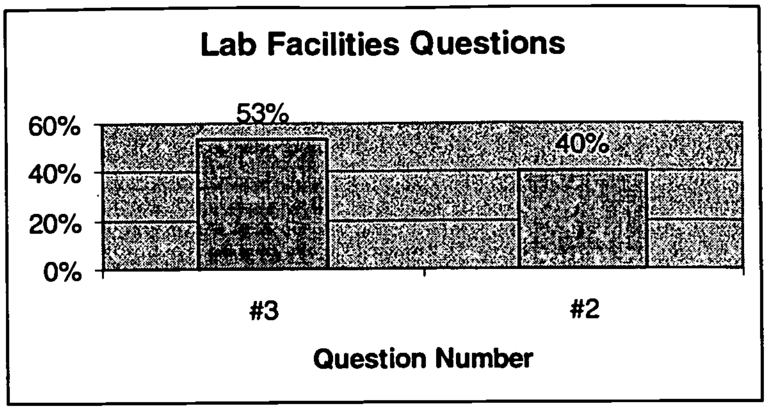


Figure 6

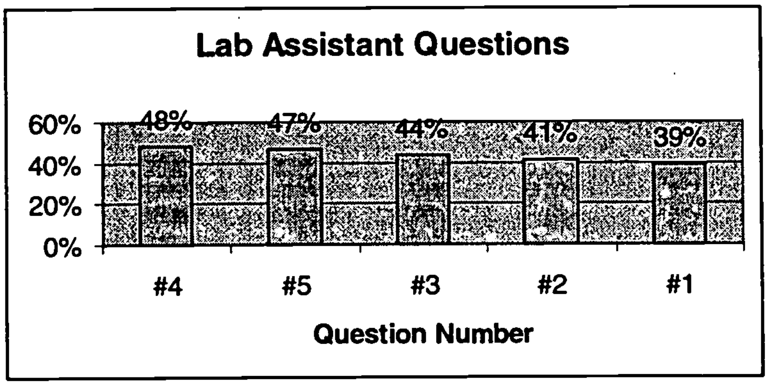


Figure 7

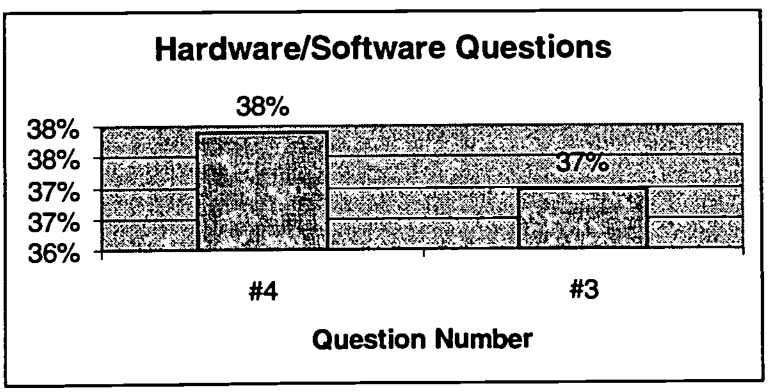


Figure 8

PDCA

We decided to use the quality tool, Plan-Do-Check-Act (PDCA) to help implement change and improve our processes in the computer labs. A team of individuals consisting of the head lab assistant, another lab assistant, and the authors of this paper was formed to brainstorm possible solutions to the lab assistant quality dimension problems and to implement these solutions.

Based on the data concerning the Lab Assistants, the team developed the following list of solutions:

12. Make name tags for the lab assistants and the head lab assistant identifying their name and function.
13. Make name plates, i.e. "Lab Assistant" and "Head Lab Assistant" displayed on the respective desks in the computer lab office.
14. Have the head lab assistant generate a list of skills necessary for a lab assistant to possess. This would aid in the selection process and the training process of current and future lab assistants.
15. Have the head lab assistant schedule a beginning-of-the-semester meeting with all lab assistants to review training issues, changes in the lab facilities, scheduling, expectations, team building, and general information.
16. Work with Continuing education department to allow the head lab assistant to participate (free of charge) in computer classes offered by that department in order to keep her skills from becoming obsolete. This would enable the head lab assistant to, in turn, train the other lab assistants.

Thus far, items 1, 2, and 5 have been implemented. Items 3 and 4 are scheduled for implementation for Fall 1998. The team will address the other quality dimensions of the survey when money becomes available to the university to purchase such things as computers, printers, chairs, workstations, etc.

In order to measure our results and see if what was implemented satisfies our customers, the survey will be redistributed at the end of the Fall 1998 semester.

Conclusion

Utilizing TQM techniques in the university computer lab has paid dividends by providing a technique to improve our lab facilities. With limited financial resources it was difficult to address all issues, especially hardware and software issues. With many universities facing similar tight budgets on capital equipment, it is important to note that it was still possible to improve the quality and services provided in the lab. The process is now in place to continuously improve the lab.

References

Hayes, Bob E., Measuring Customer Satisfaction, Milwaukee, Wisconsin, Quality Press, 1998.

Deming, W. Edwards, The New Economics, Cambridge, MA, Massachusetts Institute of Technology, 1994.

COMPUTER LABS
CUSTOMER SATISFACTION SURVEY
1997

Demographics:

1. What is your major? ________
2. What is your level? ___Freshman, ___Sophomore, ___Junior, ___Senior, ___Non-degree
3. How do you rate your computer experience?___Very experienced, ___Experienced,___Some, ___Novice
4. Do you use a computer a work? ___Yes, ___No
5. Do you have a computer at home? ___Yes, ___No
6. How often do you use the computer labs ___Daily, ___3-5 times a week, ___1-2 times week, ___less than once a week
7. What times do you generally use the computer lab? ___8:00am-4:30pm (M-Th), ___4:30pm-10:00pm(M-Th), ___8:00am-5:00pm(Sat), ___1:00pm-5:00pm(Sun)
8. What are the types of software you generally use? ___Word processing, ___Spreadsheets, ___Databases, ___Presentation (Power Point), ___Programming, ___E-mail, ___ Internet Browsing, ___other
9. What is your primary reason for using the computer labs? ___class-related, ___personal, ___both

Please rate the following questions using the scale below.
1 - Strongly Disagree(SD), 2 - Disagree(D), 3- Undecided(U), 4- Agree(A), 5 - Strongly Agree(SA)
Please circle one.

Lab Facilities:	SD	D	U	A	SA
1. The lab facilities are generally clean	1	2	3	4	5
2. The lab facilities are comfortable(temperature, lighting, chairs, desks)	1	2	3	4	5
3. There are ample facilities for use(computers and printers)	1	2	3	4	5
4. Equipment in the lab is in good shape	1	2	3	4	5
5. Lab hours are adequate	1	2	3	4	5
6. Other students are not distracting	1	2	3	4	5
7. The workstation layout is adequate	1	2	3	4	5

Lab Assistants:					
1. Lab assistants are available when you need help	1	2	3	4	5
2. Lab assistants are easily identifiable	1	2	3	4	5
3. Lab assistants are knowledgeable about software problems	1	2	3	4	5
4. Lab assistants are knowledgeable about hardware problems	1	2	3	4	5
5. Lab assistants explain the problem so that you can correct problems yourself in the future	1	2	3	4	5
6. Lab assistants are polite when giving assistance	1	2	3	4	5

Hardware/Software:	SD	D	U	A	SA
1. The hardware is adequate to do the work required	1	2	3	4	5
2. The hardware is generally reliable	1	2	3	4	5
3. Computer performance is good	1	2	3	4	5
4. Computers and printers are up to date	1	2	3	4	5
5. Software is adequate to do the work required(MSOffice, E-mail)	1	2	3	4	5
6. Software is generally reliable	1	2	3	4	5
7. Available software is up to date	1	2	3	4	5
8. The variety of available software is good	1	2	3	4	5
9. Internet access is readily available	1	2	3	4	5

General Comments:
(If you circled 1 or 2, please explain below)

Suggestions and Improvements to the Process

PREPARING A COURSE FOR DISTANCE EDUCATION DELIVERY

Dr. Thomas A. Pollack
A.J. Palumbo School of Business Administration
Duquesne University
Pittsburgh, PA 15282
(412) 396-1639
pollack@duq.edu

ABSTRACT

Many institutions of higher education are plotting strategies to enter the competitive arena commonly referred to as Distance Education. Distance education is the process of delivering learning or instructional resources to a remote location through the combined use of information technology and traditional delivery methods. At my institution, we have committed to the electronic delivery of an MBA program to Northern Jiaotong University in Beijing, China. The process of preparing a course for electronic delivery involves great attention to detail as well as a significant commitment of time. The process, along with related course preparation issues will be discussed in this paper.

INTRODUCTION

The School of Business at Duquesne University has agreed to deliver an MBA program to Northern Jiaotong University in Beijing, China. There was a great deal of uncertainty as to how program delivery might best be accomplished. We are in the third year of exchanging professors for a four to six week summer session. The cost of sending professors to do on-site delivery is prohibitive. The demand for a U.S. MBA in China is tremendous, thus creating a huge market potential. The students are very bright but economically very poor and cannot afford U.S. tuition rates. Their educational programs must be subsidized, often by American corporations with facilities in China. Computer availability and communications capabilities in general are not at all what we have grown to expect. The nature of the teaching process is more lecture-based than interactive. Thus there are numerous problems to overcome -- with finances, teaching methodology and technology.

There are countless technological options to consider for distance education delivery (Dede, 1996). After much thought and deliberation of alternatives, we decided that the best approach for our program was online coursework, and the best delivery vehicle was the Internet. The following is excerpted from a white paper prepared to explain the distance education delivery system provided by University Online (UOL), the company we have partnered with (UOL Publishing, Information and Insight...).

> Although distance learning is not a new concept, the advent of the Internet and World Wide Web has created the capability of providing distance education online. Using existing computers with Internet access, worldwide educational offerings are delivered to the desktops of learners anytime and anywhere. Unlike other forms of distance education -- satellite, computer-based training, video conferencing, etc. -- electronically delivered online instruction has a broader scope. The interactivity of Web-based delivery offers simulations, problem-solving exercises, links to information resources, and the ability to work in collaborative environments, and participate in real-time discussions with other learners and instructors.

For the reasons stated above, as well as forecasts that indicate tremendous demand for online adult education, we decided that, for our purposes, an online Internet-based instructional program would be best. We then set out to choose a partner to deliver our courseware.

EDUCATIONAL DELIVERY PARTNER

For our distance education initiatives, we considered various options and delivery methods, including interactive video (using a University-owned V-Tel system), and decided to pursue an online delivery methodology that is capable of delivering interactive courses through the World Wide Web (Web) or corporate intranets. We felt very strongly that the only economical means of providing education to Beijing, China is through an online educational program. This program will possess the following characteristics:

It will be available 24 hours per day, 365 days per year.
It will not require travel.
It can be readily updated.
It offers flexibility in course design.
Correspondence between faculty and students can be established through electronic mail or "chat".

We ultimately signed an agreement with a "for-profit" organization called (University Online) UOL Publishing, Inc. of Mclean, Virginia (www.uol.com). UOL can be reached by email at Info@uol.com. UOL is traded on the Nasdaq National Market as UOLP. UOL Publishing, Inc. has been a leading edge publisher of interactive web-based courseware and technology-enabled education for more than ten years. The Company introduced its first Web-based demonstration course in November 1995 and its first revenue-generating Web-based course in the spring of 1996 (UOL Prospectus). UOL prides itself in providing cost-effective, high-quality, user-specific courseware and one of the world's largest online courseware libraries (UOL Publishing, Inc. Information and Insight...). The Company offers its courseware primarily to part-time students and working adults in partnerships with academic institutions and business partners (UOL Prospectus).

UOL Publishing promotes its courseware offerings as a "virtual campus" (UOL Website, Take a tour...). The "virtual campus" is intended to offer all the familiar features of a physical campus with the affordability, availability, and accessibility offered by online training. Courses and training programs become part of this "virtual campus" as a result of strategic acquisitions and partnerships being forged with businesses and universities. Courseware and training programs published on the "virtual campus" are available to subscribers at the fee agreed upon by the owner. Truly, the subscriber can "shop" the virtual campus for what he or she considers to be the "best in class" offerings in any given area. The "virtual campus" concept includes a large courseware library, and delivery, registration and tracking systems and it enables any business or academic entity to quickly create an online, interactive system of education using the Internet or existing intranet. The "virtual campus" is made up of the following:

The **Registrar Building** is where students can sign up for available courses. Administrators can add new courses and admit students into existing courses. Upon registering, the student receives email to verify acceptance into a course.

The **Classroom Building** allows the student to enter their registered courses or check their grades for any course. Administrators and instructors can enter courses in progress, check students' grades, download progress reports and administer courses.

The instructor (in the **Faculty / Administration Building**) can build customized courses and tests, and the student can leave messages for instructors or attend faculty office hours.

The **Information Building** is where administrators and instructors can post news, announcements and frequently asked questions.

The **Commons Area** is the area where students can "chat" with other students or professors (UOL Website).

UOL believes that its online courseware combines convenience, affordability, self-pacing, standardized curricula, individualized tailoring of courses, immediate performance measurement, and a high degree of student-teacher interaction. These characteristics are designed to address the educational needs of part-time students and working adults, which constitute a rapidly growing segment of the education market (UOL Prospectus).

COURSE DESIGN TOOLS

UOL provides an extensive Courseware Construction Set (CCS) that is a web-based editor that guides one on the construction of a course for online delivery. It is an assembly tool, not an authoring tool in that it does not assist in the development or design of content. UOL provides access to a "blank" course and a means of uploading modules, lessons, text, graphics, audio and interactives. Each course is assigned a unique URL to access the "blank" course and CCS tools. To use the CCS, one must be connected to the Internet at all times. The CCS features an extensive Course Author's Reference Guide (UOL Course Author's Reference Guide). In the next segment of this paper, I will briefly summarize the purpose of each of the main component modules of the Reference Guide.

The **Course Structure Editor** allows you to create, modify, and delete elements of your course structure.

The **Learning Objectives Editor** allows you to associate particular pages and test questions with the objective.

The **Glossary Editor** allows you to create, modify, and delete the glossary terms and definitions within your course.

The **Syllabus Editor** allows you to create the syllabus for your course.

The **References Editor** allows you to create a page of URLs, books, articles, and other resources that your students can access as references for the course.

The **Help Editor** allows you to generate a page where you can provide guidance to your students in the form of general instructions, answers to frequently asked questions, and any other useful course-related information.

The **Test/Quiz Editor** allows you to create, modify, and delete the test and quiz elements within your course.

The **Gradebook Editor** is a reporting tool that allows the instructor and students to check the status of test scores, attendance records and progress on the Learning Objectives.

As the reader may ascertain, the modules are designed to address each of the major instructor concerns during course design. There are other modules and countless other features that cannot be addressed in the space of this paper.

PLANNING

Careful course planning is critical for success in online instruction. Although estimates vary, there seems to be some consensus that successful online courses require roughly three times as much time and attention as a course delivered in a conventional manner (Guernsey,1998). The following four guidelines are provided in the UOL Course Author's Reference Guide.

Determine the appropriate instructional technologies to use in the course. This requires decisions on how the course material is to be presented – text, graphics, audio, exercises, tests, etc.

Determine the appropriate structure for your course. This step requires that the course be divided into modules, lessons and pages. It also suggest some detailed planning about such things as pre-tests, post-tests, and interactive exercises.

Plan the screens. This step requires that you complete a storyboard to outline elements and layout and also define the order of the screens. This is a critical step in organizing your course.

Author the content of your course. Here, any graphics, multimedia, or audio files must be determined and appropriately placed.

CONCERNS

Distance education is a different experience for most educators venturing into it for the first time, because the distance students have different needs than our traditional undergraduate and graduate students. In order to create effective learning experiences for distance students, course delivery strategies must be developed that cater to any special needs of this group of learners. Specifically, if they are non-traditional adult learners, they may be somewhat unfamiliar with academic practices. They may also lack the typical support systems (such as tutors) and other resources typically found on campus. However, they are generally considered to be a disciplined group of learners (Guernsey, 1998). Our primary objective, however, should be to add educational value -- not to achieve economies of scale. Technology actually subtracts educational value if we simply try to imitate our regular classroom behavior without changes to curriculum or classroom organization (Kinnaman, 1995). This represents my greatest concern – how to effectively organize and package course materials so as to optimize the learning process.

In "Structural Issues in Distance Education (1996), Kearsley and Lynch focus on the balance between structure and flexibility in curriculum design and course delivery. Structure refers to the organization and delivery of learning events and activities, and is dictated by the syllabus. Generally, the more dialogue in a course, the less structure, and vice versa. The authors encourage a balance between structure and flexibility and summarize key points as follows:

Structure includes:

The development of a detailed course syllabus and study guide;
Well-defined instructional activities;
Fixed time schedule for course completion dates and assignments; and
A breakdown of large classes into small sections, each with a teaching assistant to help with grading and provide student feedback.

At the same time, flexibility is maintained by:

Making almost all coursework project-oriented so students can make adaptations to their interests and needs;
Encouraging students to develop their own style of leadership;
Permitting students to use email and conferencing capabilities to pursue their own interests; and
Allowing students to have access to course materials anytime.

A delicate balance between structure and flexibility must be achieved to be effective in the instructional process.

Another of my concerns is evaluation of the distance learner and the distance education program. Evaluation is defined as the process of determining the merit or worth of a product, process, or program (Hawkes, 1996). Evaluation is frequently difficult in face-to-face learning environments and is compounded in distance education. We are certain, however, that evaluative information can provide timely feedback and constructive criticism to the designers and developers of a distance education program (Ibid.). Certainly, in this regard, we will want to evaluate technical criteria. But more importantly, we must evaluate instructional criteria, including learner achievement, interactivity, course structure and flexibility, and the quality and availability of learning resources. Finally, in a distance learning environment, we must also evaluate organizational criteria --everyday support and use of technology (Ibid.).

CONCLUSION

The Internet is a tremendous resource for the effective delivery of online educational courses and training programs. It has the potential to re-shape distance education programs as we have come to know them. The Internet provides an ease of accessibility that simply does not exist with other forms of distance education delivery.

Mapping an overall strategy for the delivery and preparation of a distance education program is a very time-intensive process. Many considerations must be taken into account. A vehicle for delivery must be secured. The course design process must be carefully managed and must maintain a balance between structure and flexibility. A good deal of attention must also be given to the evaluation process as it relates to learning, courseware quality, technolgy, and delivery.

BIBLIOGRAPHY

Dede, Chris. (1996, March). Emerging technologies in distance education for business. Journal of Education for Business.

Guernsey, Lisa. (1998, March 27). Distance education for the not-so-distant. The Chronicle of Higher Education.

Hawkes, Mark. (1996, September). Criteria for evaluating school-based distance education programs. Bulletin.

Kearsley, Greg and William Lynch. (1996, March). Structured issues in distance education. Journal of Education for Business.

Kinnaman, Daniel E. (1995, January). The future of distance education. Technology and Learning.

UOL Courseware Construction Set. (1998, March 31). Course Author's Reference Guide.

UOL Publishing, Inc. (1996, November 26). Common Stock Prospectus.

UOL Publishing, Inc. (www.uol.com). "Information and insight about online education & training".

UOL Publishing, Inc. (www.uol.com). "Take a tour of a UOL Vcampus."

Internet-Based Software System for Student Evaluation of Faculty

Gerard Rambally
Georgia Southwestern State University
gkr@canes.gsw.peachnet.edu

Abstract

With approximately 90% of colleges and universities involved in distance education, a web-based approach to faculty evaluation is essential. I will discuss the design and implementation of an internet-based software system for student evaluation of faculty. First, I will discuss the design and implementation of the online questionnaire which was written in HTML and JAVA. Then we'll review the compiling of student input into a single file using a CGI script and the generation of a historical database using Personal Oracle. Finally we will look at statistical and report generation, security issues, and a review of the benefits to this approach to faculty evaluation.

(Gerard's paper was not available at the time the Proceedings went to press. He will supply copies of his paper at his talk.)

Infusing Technology into Undergraduate Core Courses

Rhonda Richards
Winthrop University
richardsr@winthrop.edu

Abstract:

This paper describes how a new program in the College of Education has designed coursework that introduces and meaningfully infuses technology into its core curriculum. A scope and sequence chart with instructional activities and intended outcomes will be included.

(Rhonda's paper was not available at the time the Proceedings went to press. She will supply copies of her paper at her talk.)

Creating a Full Interactive Online Course with *Web Course in a Box*

Steven Saltzberg
Director, Information & Technology Services
Randolph-Macon College
Ashland VA 23005-5505
ssaltz@rmc.edu

Susan Polyson
Project Manger, Instructional Development Center
Virginia Commonwealth University
Richmond VA 23084-0326
spolyson@vcu.edu

There are many tools available today that allow instructors to easily create HTML documents. These tools range from HTML WYSIWYG editors like Netscape Gold to Web Site development products such as Microsoft's Front Page. However, a new type of software now creates a full teaching and learning environment. *Web Course in a Box* is one of these new breeds of software, and was created because most faculty want to spend their time developing and delivering course content, not on figuring out new and complex technology.

With *Web Course in a Box* faculty, in a single day, can:

- create and update Web pages for classes, including an online syllabus, course schedule, course announcements, and student roster;
- setup sophisticated access control by assigning usernames and passwords to students;
- create on-line threaded discussion forums;
- create interactive online tests and drill and practice exercises;
- upload and organize course content; with WCB instructors can integrate audio, video, graphics and other media into their course content materials; and
- create a faculty home page that is linked into all of the WCB course pages.

WCB Overview

There are many tools available today that allow instructors to easily create html documents. But few faculty are really interested in learning the nuts and bolts of HTML coding. While easy to use tools like Netscape Navigator Gold, Microsoft's Internet Assistant for Word, Hot Dog or Microsoft's Front Page, can create HTML pages, the faculty member soon discovers that online courses are more than simple web pages. There are other issues such as how to get the html pages onto the server, how to setup access control functions (limiting access to students in your class), how to integrate interactive features such as discussion forums and assessment. In addition faculty need easy-to-use tools for managing, as well as creating, Web-based courses. Most faculty want to spend their time developing and delivering the content of their course, not on figuring out new and complex technology.

To address this need, WCB is designed to be a Web-based course management system that is:

Easy to Learn (for both instructors and students). Instructors and students need only know how to use a web browser to be able to use WCB. Most faculty are able to learn to use WCB with a few hours of training.

Customizable. Although WCB is a template-based system, instructors can customize their course pages to achieve individualized "looks".

Instructor controlled. In WCB, the instructor has control of their web-based environment. We wanted faculty to have the ability to add, delete and edit all components of their web-based course without having to rely on the assistance of system administrators or technical staff.

Easy to Use. Faculty do not need to know HTML to use WCB. However, instructors can use their knowledge of HTML or an HTML WYSIWYG editor to enhance their WCB course pages.

WCB Features

To use WCB, an instructor uses their Web browser to connect to a WCB enabled web server. After logging in to WCB with a unique username and password, the instructor will use The WCB Authoring Tools which make extensive use of HTML forms. These are Web pages containing fill-in text fields, pull-down menus and buttons.

With the WCB Authoring Tools, an instructor can:

create and update Web pages for classes, including an online syllabus, course schedule, course announcements, and student roster;

setup sophisticated access control by assigning usernames and passwords to students; access control allows instructors to restrict access to their course materials to only those who have authorized usernames and passwords;

manage course and student information;

create on-line threaded discussion forums; the WCB forums include features such as forum archiving and attachments (allowing you to post messages and include binary file attachments) and the ability to specify access to subgroups of students within a class;

create interactive online tests and drill and practice exercises;

upload and organize course content; with WCB instructors can integrate audio, video, graphics and other media into their course content materials;

Create a faculty home page that is linked into all of the WCB course pages.

WCB supports an unlimited number of instructors, students and courses. Instructors can edit existing courses at any point and modifications are immediately available on the web server. When a student is added to the WCB system, a unique username and password is automatically assigned. Students can be enrolled in multiple WCB courses, and their single username and password provide access to all courses in which they are enrolled.

To use WCB, a student uses their Web browser to connect to a WCB enabled web server. Students have the ability to view course pages for all courses to which they have been given access. In addition, students are able to participate in online class discussions by posting messages to the forums, complete online quizzes and tests, submit homework assignments, and create their own WCB homepage.

How to Get Web Course in a Box

WCB is a product of a development effort by Virginia Commonwealth University which has licensed its distribution and support to MADDUCK TECHNOLOGIES. MADDUCK supports WCB in two ways.

First: MADDUCK provides a download site (http://www.madduck.com) so that institutions can download and install the WCB server software on a Web server at their institution. Typically, the institution's web master would perform this installation. Once WCB Server is installed, individual faculty and students will use the web browser installed on their personal computers to access the WCB web site at their institution. Instructors and students should use Netscape Navigator 3.0 or higher for the best results with WCB.

Second: MADDUCK TECHNOLOGIES manages a Web Course Hosting Service so that interested faculty members can purchase access to a WCB site. For more information visit http://www.wcbcourses.com.

The WCB Server runs on Unix, Windows NT, and Macintosh systems. WCB is an integrated set of CGI scripts that are installed as an add-on to an existing Web server running on one of these operating systems. For information about which Web server software is compatible with WCB, you may visit the WCB Home Page at: http://www.wcbinfo.com . WCB is written in perl and also requires that perl 5.001 or higher be installed on the web server system.

Undergraduate Data Mining on the World Wide Web

Anthony Scime'
Business and Economics Department
Wilson College
1015 Philadelphia Avenue
Chambersburg, PA 17201
(717) 264-4141
ascime@wilson.edu

Abstract

Currently available World Wide Web search engines determine a site's qualification as a response to a search request by matching keywords in the request to keywords representing the site. The returned sites are given a score and ranked according to the match on keywords. Many of these retrieved sites can be irrelevant to the user's true information needs.

Undergraduate students with information retrieval and computer literacy skills should be able to search the Web to find and extract information relevant to a domain. A course in the fundamentals of information gathering from distributed heterogeneous sites can improve these skills. This course examines methods, theories, and techniques and provides practice in information retrieval, categorization, and knowledge discovery from text and other unstructured data sources such as the Internet and the World Wide Web. This paper is a report on the results of such a course and the experiences gained.

Introduction

In this information age, people are inundated with vast amounts of information. Technology has made this inundation possible and placed information sifting and evaluation on the individual. This is most evident on the World Wide Web. Students need to learn the fundamentals of information gathering from the distributed heterogeneous sites that make up the Web. As they go out into the world, they will use the Internet and the World Wide Web to find, assess, and disseminate information [FULT98]. They need to understand theories and techniques for information retrieval, storage and categorization from textual, unstructured data sources. This paper discusses a course that investigates theories and practices skills in finding information and assembling knowledge from the Web.

Course Requirements

A special topics course in basic distributed information retrieval was developed to allow students to learn how to appraise the quality and reduce the quantity of information retrieved from the Web. The course studies the internet, distributed databases, text based database technology, information retrieval, data warehousing, data mining, web page construction using HTML, data validation, and source citation. The objective is to enable the student to use the World Wide Web as a research tool and gain an understanding of the Web's structure. Using Thuraisingham [THUR97] for the distributed data management theory and Caswell [CASW97] for the practical aspects of the Web, the student:

Demonstrates an understanding of the structure of distributed data sources,
Performs basic information retrieval from the Web,
Develops a data evaluation method,
Creates a data warehouse of retrieved Web site addresses and summaries,
Practices knowledge creation through data mining of the data warehouse, and
Places the knowledge back into text form.

As the major assignment, each student selects a topic to research on the World Wide Web. They design and construct a data warehouse type database to store the results of the information search. Based on the sites found and the use of data mining techniques, an 8-10 page research report is written as an HTML document. A second report of 5-7 pages is written describing the information retrieval, evaluation, storage, and data mining techniques used for creation of the first paper. In the process of meeting the course's research paper requirement, the student must develop the concept of the paper, conduct Web searches, evaluate, store, and review the results, and assess the results for new knowledge. The architecture for this process is provided in Figure 1.

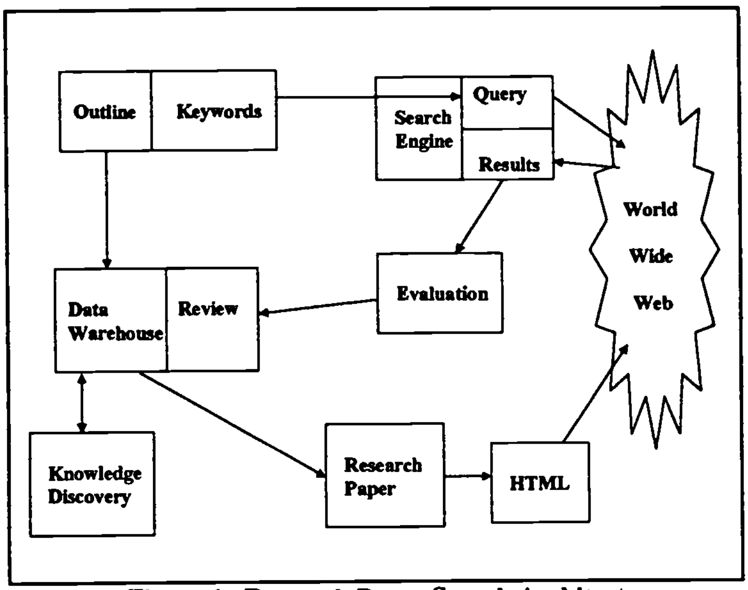


Figure 1. Research Paper Search Architecture

The Data Warehouse is implemented in MS Works 3.0 database or MS Access (for more advanced students). Because a database course is not a prerequisite, the database design consists of a data dictionary, sample table(s) and notional data. During the course, data models are discussed. Students can choose to implement the data warehouse as multiple tables as discussed in this paper. The final populated database contains 40 – 100 entries derived from at least two search engines. Data warehouse design, search methods, evaluation techniques, and data mining methods are determined by the student.

Research Paper Outline and Data Mining the Web

One of the problems with conducting a search is the need to know something about the subject before beginning to research information. The student's initial ideas are expressed as the paper's outline. On the Web, research comes in the form of search engine queries using keywords. The queries should be formulated so that all possible information about the subject will be found. The paper's outline helps define the queries.

Outlining involves the decomposition of a topic into its component parts, which are further decomposed until non-decomposable leaf-node sub-topics are reached. The final sub-topics are defined in an unambiguous, precise manner that avoids confusion as to the intent of the sub-topic. This preciseness will allow for accurate measurement of a document's applicability to the sub-topic. See for example, the outline for an Air Pollution paper in Figure 2.

Topic – Air Pollution	Keywords
1. Environmental Pollution	"Environmental Pollution" or "Air Pollution"
2. Sources	
2.1 Transportation	
2.1.1 Aircraft	"Air Pollution" and Aircraft
2.1.2 Automobiles	
2.1.2.1 Emissions	(Automobiles or Cars) and Emissions
2.1.3 Railroads	"Air Pollution" and Railroads
2.2 Industrial	
2.2.1 Oil Refineries	"Air Pollution" and Industrial and "Oil Refineries"
2.2.2 Power Generation	"Air Pollution" and "Power Generation"
2.3 Fuels	
2.3.1 Coal	Coal
2.3.2 Gasoline	Gasoline
2.3.3 Nuclear Power	"Nuclear Power"
3. Effects	
3.1 Health	"Air Pollution" and Effects and Health
3.2 Agriculture	"Air Pollution" and Effects and Agriculture
4. Control	
4.1 Industrial	"Air Pollution" and Control and Industrial
4.2 Transportation	"Air Pollution" and Control and Transportation

Figure 2. Air Pollution Paper Outline

Data mining is one of the more significant parts of the course. Data mining occurs in two phases. First, the Web is mined to find sites of interest concerning the subject domain. This mining operation is accomplished using the paper's outline and search engines. A query is created for each of the leaf-node sub-topics in the paper's outline using keywords, which represent the sub-topic concept. See Figure 2 for the leaf-node sub-topic keywords for the Air Pollution topic. The queries may be a single keyword, a collection of keywords, a string, or a combination of keywords and strings.

Although a leaf-node sub-topic may have a specific meaning in the context of the paper, the use of a keyword or string could lead to the retrieval of many irrelevant sites [BRAY96]. In this case, keywords and strings may be "and-ed" as far back along the branch path as necessary to increase the specificity of the retrievals, as in sub-topics aircraft, oil refineries, and health (Figure

2). "And" between keywords requires that both keywords occur in the result. For example, the Health sub-topic search using the WebCrawler search engine and the keywords: "Air Pollution" and Effects and Health returns more specific results (310 documents) than simply Health alone (62,757 documents). Because no search engine completely indexes the Web, it is best to use more than one search engine [SEL95]. After evaluation, data on selected sites found are added to the data warehouse.

Evaluation

As sites are found by the search engines, some method is needed to quickly evaluate the site's quality. This effort could be completed programmatically on all the sites returned by the search engine. However, these students have no programming experience. Moreover, manual evaluation forces the student to "think" like the computer. She must evaluate the sites independently of knowledge in the domain. She must use the techniques she developed for evaluation as if she was the computer. Manual evaluation does limit the number of sites that can be considered. Time is a factor in completing the paper. Thus, the order provided by the search engine influences the evaluation results.

Each student develops her own evaluation method. Among possible methods are those based on the syntax of the site and the semantics of the research problem. The first, syntactic, uses heuristics to determine the quality of the site's information. The second, semantic, considers knowledge of the domain from which the search keyword is derived. These techniques are intended for use prior to reading the document. One or a combination of techniques can be used to select document data to populate the data warehouse. Below are provided two sample evaluation methods.

Syntactic Evaluation of a Site

Considering just the document and making a quality evaluation based on its internal structure constitutes a syntactic evaluation. A simple syntactic technique is to assign a value based the URL domain type. The domain type is the last two or three characters of the URL. The most common domain types are ".com" for a business, ".edu" for a school, ".gov" a government agency, ".mil" the military, ".net" a network provider, and ".org" an organization. Two letter domain types refer to the country of the Web site, such as ".us" and ".ca" (United States and Canada).

Checking the three-character domain type may provide a hint about the quality of the information contained at a site. The types ".gov" and ".mil" may be trusted to present government policy and positions on issues.

Information in domain types business and organization may provide information bias to the owner's goals. Business sites are often marketing tools and present the business' products in the best possible manner. Organizations generally have an agenda that is supported on their Web sites.

Schools can present unbiased information on a subject in their search for truth and learning. Quality at ".edu" sites can still be a problem requiring additional syntactic methods. There may be a difference between a site containing a university researcher's published work and a progressive elementary school's Web posting of student papers. Perhaps this can be accomplished syntactically by looking for clues such as "Dr", or "PhD" in the author portion of the document.

A scoring system may be developed where a 3 is assigned to ".edu" PhD sites, 2 to ".gov", ".org" and other ".edu" sites, 1 to ".com", and 0 to any others. With the 10 selections scoring highest having their documents read by the student and their data added to the data warehouse.

Applying the syntactic rules and scoring above on the first 2 of 310 health sub-topic results, the site EPA's Indoor Air Quality Home Page with URL

http://www.epa.gov/iaq/

scores 2 because it is a ".gov" site. Air Pollution and Health with URL

http://www.tec.org/greenbeat/may96/health.html

is an ".org" site and also scores 2.

Semantic Evaluation of Sites

Semantic evaluation considers the value of the site found within the conceptual intent of the query. Search engines use semantics in their query processing by the match of keywords.

A simple semantic method is to count the occurrences of the search keywords in the document title. The more keywords present in the title the higher the score. The 10 sites with the highest scores are entered into the data warehouse.

The top scoring sites are added to the data warehouse. A semantic evaluation on the top two sites from the Health sub-topic gives the first site, EPA's Indoor Air Quality Home Page, a 0 score. None of the keywords "Air Pollution", Effects, or Health, appears in the title. The Air Pollution and Health site scores 2, two of the keywords are present.

Repertory Grid

An improvement in evaluation occurs if the syntactic and semantic scores are combined to provide a site's overall relevance to a user's search request. The product of the syntactic and semantic scores can be computed using a repertory grid [SHA87]. The repertory grid score for the EPA's Indoor Air Quality Home Page, 0, is the product of the syntactic score, 2, and the semantic score, 0. The second site found by the search engine, Air Pollution and Health has a repertory grid score of 4 (2 x 2). This result places the search engine's second place site before its first place site.

Performing this analysis on all sites returned by each search creates rankings that consider the returned documents quality and content. The documents scoring highest for each sub-topic search are added to the data warehouse.

Data Warehouse

A data warehouse is a storage place for data awaiting use. It is a subject specific relational database, which has been populated from diverse and possibly heterogeneous data sources such as exist on the Web. In the operation of the data warehouse, data is selected for inclusion but never sent back to the original source. Transfer is in one direction only. The selection of data items for inclusion is dependent on the purpose for the data warehouse [ROB97]. The previous evaluation phase accomplishes this selection.

As source sites are found and evaluated, a record is written containing the sub-topic of the paper's outline and a quote or summary from the source or the student's ideas on the sub-topic as discussed in the source. The summary record includes additional keywords provided by the student. These additional keywords are generated by the student as she reviews the site. The summary keywords may be different than the keywords used in the site's selection by the search engine. These summary keywords provide additional specificity on the information in the site, as it applies to the overall topic of the research paper. These are details that may not have been evident without the focused reading of the student.

This summary record also contains a bibliographic code to link this record to a second record of bibliographic data. The bibliographic record contains bibliographic information about the source site. This bibliographic record permits the retrieval of the source document should it be needed again. As well, the collection of bibliographic records provides data for the bibliography at the end of the paper. The bibliographic record contains the source's author, title, publisher, copyright date, URL, retrieval date, search engine used for retrieval, the keywords or phases on which the retrieval was based, and the bibliographic code (used to connect to the summary card).

In the example, the Air Pollution and Health document is eligible for entry into the data warehouse. The Air Pollution and Health document found for the sub-topic Health is summarized and results in summary keywords: lungs, ozone, smog, Texas, cars, emissions, and "reduction of air pollution." The keyword and summary (below) becomes part of the summary record in the data warehouse.

> Air Pollution causes a lack of oxygen being taken into the lungs. The air containing large amounts of ozone, sulfur dioxide, nitrogen dioxide, carbon monoxide, lead, and particulate matter reduces the amount of room for oxygen. There is a high correlation between lung disease and air pollution. The less capacity the lungs have, the greater the effect of pollution, such as in children and the elderly. Urban areas in Texas have been found to have especially high levels of ozone pollution. Smog caused by ozone and chlorofluorocarbons is pollution that can be seen by the naked eye. High levels of smog have caused death in some cases. Automobiles are the greatest cause of smog and air pollution. By controlling automobile emissions and increasing fuel efficiency, air pollution can be controlled. Other controls are to reduce the production of electricity by coal.

Knowledge Discovery – Data Mining the Data Warehouse

The second phase of data mining is knowledge discovery from the data warehouse. Data mining is the process of extracting information from large quantities of data. Queries are designed to explore relationships between data records that are not necessarily obvious. Data mining, hopefully, results in nuggets of knowledge heretofore unknown.

This is analogous to gold mining. The prospector through experience and expertise selects a location likely to produce gold nuggets. She proceeds to dig, sift, and pan until gold is found or she realizes this is not the place for a gold mine.

The research paper data warehouse is mined to discover connections between the documents, and their application to the paper outline. This may result in a possible reorganization of the paper. This data mining attempts to classify, associate, or cluster the sites [YEVI97] (Figure 3).

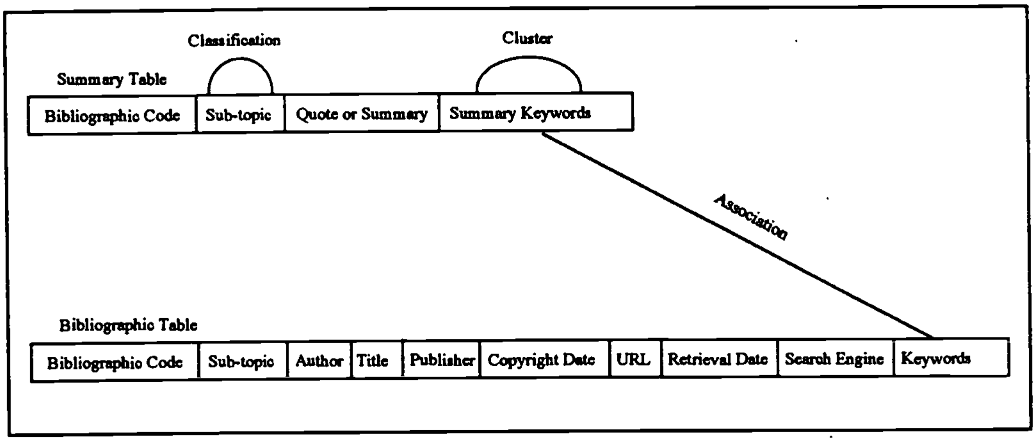


Figure 3. Data Mining the Research Paper Data Warehouse

Classification is the assignment of a database record to a class defined by the database user. For the research paper data warehouse, the paper's structure, its outline, provides the classes of interest. The retrieval keywords used, coming from the paper outline, created groups of documents based on the sub-topics of the outline.

Associations are found by discovering a common value in two data items in different record types. When the keyword in a bibliographic record matches a summary keyword in a summary record, a second document becomes a possible source of information for the original document's sub-topic without regard to the bibliographic code relationship. An association table can be added to the data warehouse.

In clustering, a summary keyword of the same value occurs in records in different classes, different sub-topic values. Documents in the same cluster (using the same summary keyword) are identified by the addition of a cluster table to the data warehouse. Of course, a document may belong to multiple clusters.

For example, data mining the air pollution data warehouse discovers the document "Air Pollution and Health" also fits under the Emissions sub-topic. There is an association between the "Air Pollution and Health" document and the Emissions sub-topic. The "Air Pollution and Health" document contains the summary keywords cars and emissions. These summary keywords are the keywords used in the Emissions sub-topic search. Data mining of the data warehouse discovered this otherwise unknown relationship.

Writing the Research Paper

After complete analysis, old and new structures for the paper appear. The old structure is the outline and the classifications originally developed by the student. The new structure includes associations, and clusters as a result of data mining the data warehouse. The paper is now ready to be written.

The actual writing process is a review of the summaries of the documents in accordance with the paper's outline while being aware of the associations and clusters. The summaries were written by the student after the evaluation process, creating the summary keywords. The summary keywords are the representations of the student's understanding of the concepts in the document. These summary keywords are the basis for the data mining and paper reorganization. The ideas of all the documents are now organized to permit easy writing of the research paper. In addition, the student has the bibliography completed in the data warehouse. Finally, by writing the paper in HTML and placing it on the Web, all the source documents can be reached through hypertext links in the paper.

References

[BRAY96] Bray, T., and Weinberger, D.; "The Hits Just Keep on Comin'"; http://www.opentext.com/corp/otm_hits.html; 1996.

[CASW97] Caswell, S.; New Perspectives on The Internet Using Netscape Navigator Software: Introductory; Course Technology; Cambridge; 1997.

[FULT98] Fulton, K.; "Learning in a Digital Age: Insights intro the Issues;" T.H.E. Journal; 25, 7; February 1998; pp. 60-63.

[ROB97] Rob, P. and Coronel, C.; Database Systems: Design, Implementation, and Management; Course Technology; Boston; 1997 pp. 677-746.

[SEL95] Selberg, E., and Etzioni, O.; "Multi-Service Search and Comparison Using the MetaCrawler"; Proceedings of the 1995 World Wide Web Conference; 1995.

[SHA87] Shaw, M. L. G., and Gaines, B. R.; "KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices;" Int. J. Man-Machine Studies; September 1987.

[THUR97] Thuraisingham, B.; Data Management Systems; CRC Press; New York; 1997.

[YEVI97] Yevich, R.; "Data Mining;" in Data Warehouse: Practical Advice from the Experts, ed. by Bischoff, J. and Alexander, T.; Prentice Hall; Upper Saddle River, NJ; 1997; pp. 309-321.

Effectiveness of Multi-media Training Programs

Charles W. Smith
Executive Vice President
Applied Business Technologies, Inc
4631 West Chester Pike
Newtown Square, PA 19073
(610) 359-0700 Ext. 105
csmith@abtcampus.com

Introduction

ABT is in the process of developing multi-media training CD's to support the implementation of its PowerCAMPUS student information system product. Multi-media training is a very powerful tool for achieving training objectives. Trainees take the course at a time convenient for them and progress at a pace suited to their ability. Post course testing reveals this type of training is significantly more effective than traditional classroom training. Especially effective is the eye-to-eye contact with the instructor during the video portion of the training. A comparison of the life-cycle costs to deliver system training using a multi-media enhanced program vs. a traditional one shows the significant economic advantage of the technique. The advantage escalates dramatically with scale.

Traditional Training Techniques in the Software Industry

The dominant approach used in the software industry today to deliver training is on-site instructor based training. This method requires an expert to deliver a standard curriculum of training on each module in the system. The average number of modules a senior trainer can master is four to six. Many enterprise systems contain twenty to thirty modules.

Training in each module is normally broken up into three sessions. Each session normally lasts three days. In the first session, the objective is to teach the students how to set-up the system to meet their needs. For this session, the students consist of managers and/or power users as well as administrative computing system administrators. The second session is designed to train the end-users, after the data conversion has taken place. This second session also serves to review problems encountered during set-up, provide remedial training to decision makers and advise on set-up options. The third session occurs during the go-live period, and consists primarily of on-site operational assistance.

This type of training is used in three situations, initial system training conducted during system implementation, training new employees when there is turnover in the organization and to re-train personnel who were not able to grasp the fundamentals during the initial training. Re-training generally occurs three to six months after the initial training and generally consists of one two or three-day session.

Traditional instructor based training has the following characteristics:

- No session can be conducted without the instructor present,
- For maximum effectiveness, the instructor must review the course script and course objectives prior to each session; this is especially true as the number of modules an instructor is expected to master climbs,
- As demand on the provider organization for training in a module climbs, there is a diseconomy of scale; that is, as the vendor becomes more successful at selling its enterprise system, it must hire more instructors to conduct the training. The timing of the hiring generally lags the demand, which means there is stress placed on incumbent instructors until the new hires are up to speed. New instructors do not have the experience base of the existing instructors.
- The client organization incurs 100% variable cost for the training; all re-training and training of new employees is equal to the unit cost of the original training.

The effectiveness of traditional instructor based training is a function of:

- the quality and health of the instructor,
- the quality of the training program, and
- the attendance, skill level and health of the students.

One of the most challenging obstacles facing an instructor is the varying skill level of the students. The pace of the instruction is a function, in general, of the "average" skill level of the students. However, no one is "average". Furthermore, all training must be completed in the time allowed. Invariably, some students "don't get it". The slow pace frustrates others.

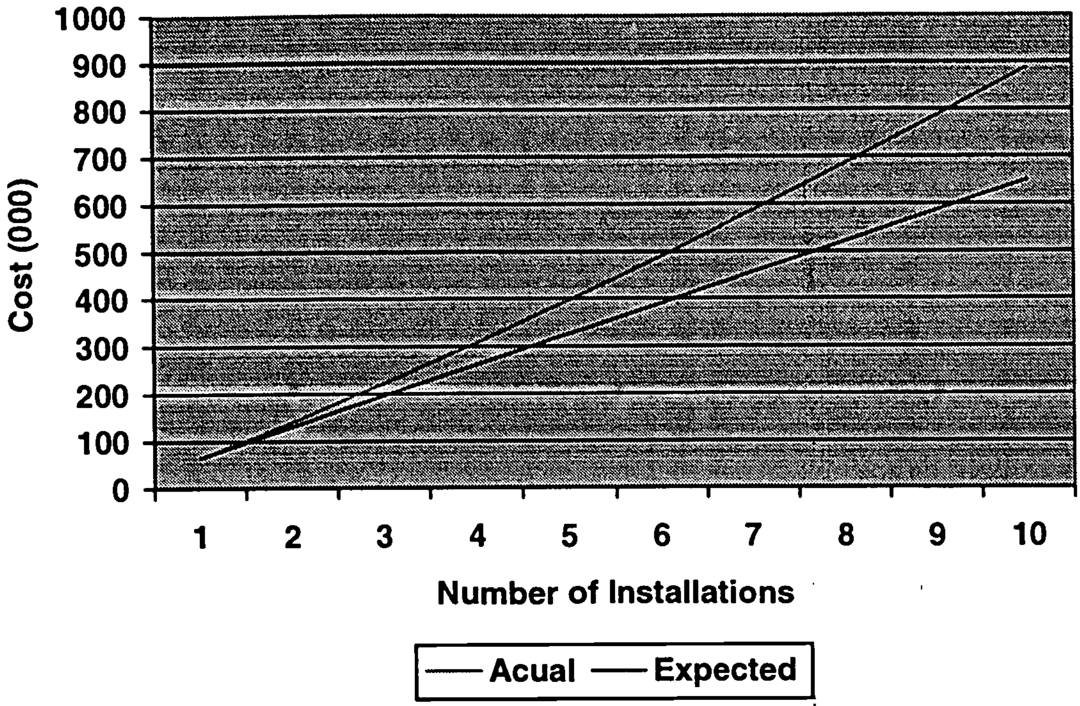
Of course, absenteeism for a scheduled training session is devastating. All costs to deliver the training are incurred. Yet, the absentee student will require remedial training and will be unable to continue with the rest of the curriculum effectively until the remedial training is delivered.

The unit costs of delivering traditional instructor based training average:

- $1000 per day for the instructor's time; this includes the overhead for supporting the instructor, return to investors in the vendor enterprise, as well as the direct costs of the instructor's salary and benefits,
- $330 per day travel expense,
- $100 per day materials expense.

An average administrative system requires 65 days of implementation training and 10 days per year of maintenance training. The average life cycle of an administrative system is seven years. An average small college or university has five users per department in six different departments – admissions, records, billing, financial aid, alumni/development and the business office. Total training costs during this period, with an average class size of five pupils, are $180,000. Exhibit 1 presents the costs of delivering traditional instructor based training as a function of the number of installations.

Exhibit 1
Cost of Traditional Training



Multi-media Training

Multi-media training is a new paradigm in delivering training programs. It incorporates all the benefits of CPT (computerized program training) with the full action video and audio capabilities of modern computers. CPT is a form of training that allows the user to learn when they want to, and at what pace they want to. The program itself, furthermore, is not dependent on the quality of the instructor or the health of the instructor. The programs can be designed and delivered by the top instructors in the organization.

Multi-media programs are laid out very similarly to traditional programs. The training objectives for each segment of the program are the same as traditional programs. However, unlike traditional training, the student controls the pace of covering the material.

With multi-media, the student is one-on-one with the instructor. There is direct eye-to-eye contact with the instructor. This characteristic has been proven to be more effective than traditional classroom training. The attention of the student to the training material is stronger, for a longer period.

If the student does not understand a section of the material, the student may return to review the lesson. Hence, the student is in control at all times. Returning for review when the student desires is more effective than "later", since the feedback and reinforcement are immediate.

After each lesson, the student can be tested on the material, as with traditional training. With multimedia, however, the student can be put back into the lesson at the point where the material is covered. This is tantamount to tutoring. Multi-media courses can track grades and report student performance to the course administrator. To gain certification, a minimum grade must be achieved.

Limitations

Multi-media training does not completely replace traditional training. Programs cannot anticipate all questions or needs. Multi-media is most effective at addressing "basic" skills, such as navigation through a software product, basic functionality and set-up. Instructors are still required to help decision makers analyze the best set-up for their environment and to guide students who are unable to orient themselves to the training they need.

Multi-media training is enormously scalable. Once developed, a multi-media course can be duplicated for a few dollars. The amortized cost of a multi-media workstation is a few dollars a day. Support is minimal.

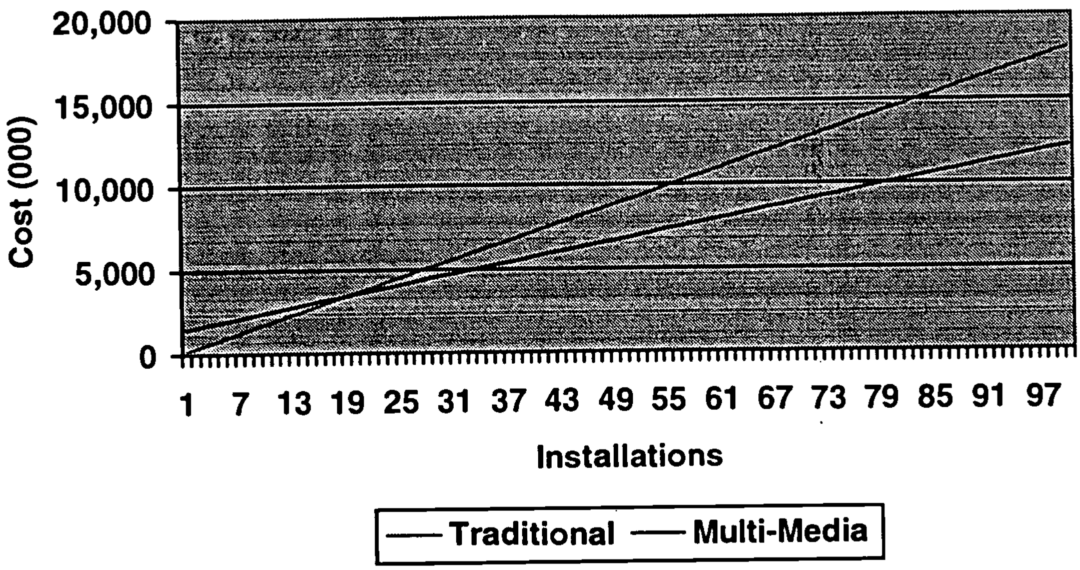
Most students like multi-media courses better, and most score higher on tests after shorter exposure to the material. The same quality course can be delivered consistently. Modifications to course material are easy with modern development tools. Video productions, however, are expensive and skill is required to produce a multi-media program to know what should be and what should not be shot.

Much of the cost of delivering multi-media training is fixed. The cost to produce an hour of multi-media training is about $20,000. This includes all programming, course design and audio/video production. Experience indicates that the ratio of contact time between traditional and multi-media training methods is about three to one. That is, three hours of traditional training can be condensed to one hour of multi-media training.

Approximately forty percent of the material in a traditional software training program can be adapted to a multi-media program. That is 26 days in an implementation project, or 69 hours of multi-media programming. The cost for a software company to produce a full set of multi-media programs would be $1,380,000. Exhibit 2 shows the break-even point for a company to develop a multi-media program.

Exhibit 2

Comparison of Traditional to Multi-media Training Costs



As can be seen, the argument for multi-media is quite strong.

Demo

The presentation of this paper includes a live five minute demo of ABT's multi-media training CD for its PowerCAMPUS application.

Summary

ABT has adopted a multi-media approach to its delivery of training services. We believe multi-media will allow us to scale up much faster with little or no loss of training effectiveness. We believe our customers will benefit from the reduced cost of training both initially and on-going as required for existing and new staff. ABT offers its training CD's at no additional cost to the basic cost of its software.

Computing Trends in Small Liberal Arts Colleges: Ten Years Later

Peter Smith
Mathematics Department
Saint Mary's College
Notre Dame, IN 46556
(219) 284-4493
psmith@saintmarys.edu

Abstract

Ten years ago, I visited 41 small liberal arts colleges in the east and midwest and presented a paper at ASCUE summarizing the answers to questions such as: What is happening to Computer Science programs at these colleges? What trends are noticeable in support staffing levels and organization? How many public access microcomputers are enough? Should students be required to buy computers? Should colleges provide faculty, administrators and staff with micro-computers? [Smith]

This past fall and winter, I visited another group of 41 colleges with significant overlap with the group from ten years ago. Some of the questions I asked ten years ago were still relevant and others were not. One phenomenon which has revolutionized academic computing in the last decade is the ubiquitous nature of networks and access to the web. This paper contains tables in the appendices and graphs throughout to help colleges position themselves on the spectrum of answers to many questions, both old and new, regarding information technology at small colleges.

1. Introduction

As part of my sabbatical project during the 1997-98 school year, I visited 41 small liberal arts colleges ranging in size from 300 to 3,800 students (c.f. Appendix 1). Over half of the schools had enrollments between 1000 and 2000. I selected these particular schools because they constituted the list of schools to which Saint Mary's compares itself. At each school, I made appointments with the Academic Dean, the Computer Center Director and a faculty member in Computer Science and asked 15 factual and 10 open ended questions from a survey form which I sent in advance to the chief academic officer at each school (c.f. Appendix 4).

My purpose in visiting these schools was to talk to key people in the information technology areas and to see the computing facilities in action. I was interested in innovative means the schools were using to solve problems common to all of us – how to optimize the use of scarce equipment and personnel resources to support teaching and learning. At each campus, I walked through all the academic buildings and poked into classrooms, departmental labs and campus-wide computing facilities.

2. General Impressions

This is an exciting time for information technology in higher education. Access to a networked computer is becoming almost as commonplace as access to a phone or to electricity. Whereas only

14 of the schools I visited ten years ago planned to put a computer in each faculty office, every one of this year's schools had already provided or was in the process of installing networked workstations in every faculty, administrative, and staff office, as well as network drops in most classrooms. In fact, by next fall, all but six of the colleges will have network connections in their residence halls on a "port per pillow" arrangement. Four of these six have plans to network their residence halls within three years.

An interesting phenomena which has emerged in the last few years is the merging of library and computer services under a single director. Only 8 of the colleges I visited had gone this route and one of these had reverted back to separate directors after a year or two. In all cases, the merger occurred because of the particular situation the college found itself, usually upon the resignation of one of the two directors. The most successful mergers occurred when a technology-oriented library director was put in charge of both library and computer services. Computer support staff were more open to the service-oriented approach of the library than the library staff was to the technology-oriented atmosphere of a computer center. In two cases, the director of the merged entity was elevated to CIO (Chief Information Officer) status.

One area in which there has been no change from ten years ago is the policy requiring students to buy computers. While there is discussion at several schools about leasing workstations to each student or weaving the cost of a computer into a student's financial aid package, none have established such a policy. While not many schools have survey data to back up their estimates of the percentage of students bringing networkable computers to campus, the median estimate is 40%. The percentage of networked computers is significantly lower. The usage trends indicate that the first year network connections are available in residence halls or classrooms there is not much use made of them. The following year the usage increases significantly and by the third year there is widespread use.

Providing network connections in the residence halls has not reduced the demand for public access computers. The most common reason for this counter-intuitive result is that much academic work in residential colleges is beginning to stress collaboration among groups of students. These groups prefer to work in a space where there are several computers and a printer available. E-mail is very popular, with more than 90% of the students checking their email each day at many schools. The public labs are used to read and answer email by students who prefer not to return to their residence halls during the day. In fact, the median number of students per public access workstation has fallen over the past ten years from 40 to 11. The last decile in 1998 is the same as the first decile in 1987 (c.f. Appendices 2 and 3, line 4). The number of hours per week a public workstation is available for each student has risen from a median of 2 in 1987 to 10 in 1998. Even with this increased availability and access, all schools are still reporting steady demand for their public lab facilities.

I was continually surprised by the different but effective ways that schools are designing classroom and lab facilities to make them easy for non-technically-oriented faculty to use, and by the commitment to service on the part of over-worked information technology staff. I discovered that several schools give students excellent opportunities for leadership in helping faculty, staff and other students make optimal use of the college's information technology resources.

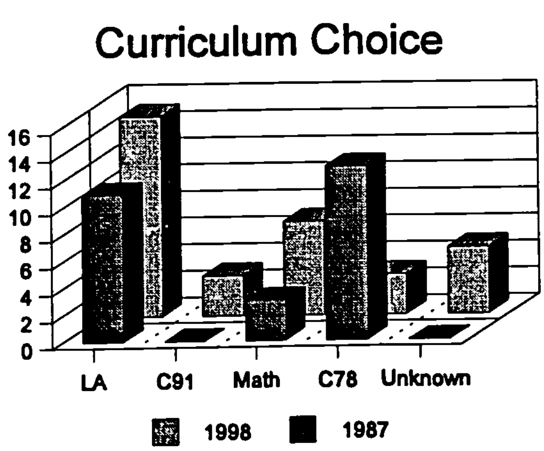
Ten years ago students were used almost exclusively as lab monitors. Now, they are acting as webmasters, conducting workshops for other students, faculty and staff, handling network problems in the residence halls, repairing equipment, working at help desks, etc. Nine schools use the Residence Computer Consultant (RCC) model pioneered at Stanford and adapted for small colleges by Wellesley. [RESNET] This model entrusts students with varying degrees of responsibility for residence hall computing in return for special privileges, equipment and/or salaries. Of course, the experience gained by students in all these support areas is invaluable as they enter the job market.

I will group the rest of my findings into categories and cover each category in a separate section: Computer Science Programs, Support Staff, Equipment, and Financial Results. Interspersed in the text are graphs summarizing the data on the topic being discussed.

3. Computer Science Programs

Of the 41 colleges visited, all but 8 have a computer science (CS) major and all but 8 have a minor program. Only 2 colleges have neither a major or a minor (some schools with CS majors do not allow minors). This marks an increase from ten years ago when only 26 of the schools had major programs. The enrollment in the major is somewhat limited as none of the schools had more than one percent of their student body graduating with a computer science major this year. I asked if the CS major was expected to attract students who would not normally choose the school, and no one, from the deans on down, thought that this was the case. Some faculty believe that Computer Science attracts a higher quality of student, however.

Ten years ago, the schools offering majors were evenly split between those who followed Curriculum 78 [ACM] and those who followed the Model Curriculum for a Liberal Arts Degree in Computer Science [Gibbs & Tucker], with a few opting for a math-oriented model (i.e., one requiring more than five math courses). This year, the most popular model was the Gibbs & Tucker Liberal Arts (LA) model (15 schools), followed by the math-oriented one (7 schools). Only 3 schools were still using Curriculum 78 (C78), 3 had full blown Curriculum 91 [ACM/IEEE] (C91) programs, and 5 were impossible to classify.



Although I expected to find that the location of the Computer Science major had a significant influence over the curriculum, this was not the case. Of the 7 colleges with math oriented curricula, 4 had Computer Science under the Math department and the other 3 had separate Computer Science departments.

The number of schools with separate Computer Science departments has increased over the ten years from 6 in 1987 to 15 in 1998. The reasons cited for separate departments ten years ago (i.e., differences in faculty evaluation, equipment needs and course loads) are still valid. In addition, many of the CS faculty who were originally trained as mathematicians are ready for retirement and are being replaced whenever possible by faculty with PhDs in Computer Science. It is easier to attract these new PhDs to an independent Computer Science Department.

The total number of Computer Science full-time-equivalent (FTE) faculty in all 41 schools this year is 93, compared with 88.5 in 1987. Of these 93, 41 have a PhD in Computer Science, compared with only 15 in 1987. The faculty with Computer Science PhDs are fairly spread out among 25 of the 41 schools, whereas only 9 schools had Computer Science PhDs in 1978. A number of schools are actively searching for faculty with CS PhDs, with a confidence that was lacking ten years ago.

It is interesting to note that the number of FTE Computer Science faculty in the schools I visited has not increased much over the last ten years in spite of the insistence of Curriculum 91 and the ACM/IEEE accrediting body that every department that offers a CS major should have a minimum of 5 faculty FTE. All but six of the departments have 3 or fewer teachers and seem to be coping quite well, although the graduating-senior/CS-faculty median ratio is 3 to 1. If we expand this ratio to include the other 3 classes, realizing that the lower division classes have more potential majors, we obtain a much higher student/faculty ratio than the one for the college as a whole.

Although the number of graduating seniors in Computer Science is not very high, many schools report that their lower level courses are bursting at the seams and that this is a new phenomenon in the last year or two. Only 3 schools are experiencing a decline in the numbers of majors, 12 are holding steady, 11 are experiencing a slow rise in the numbers of majors, and 9 are seeing a steep increase in majors. This trend mirrors the perceived shortage of computer scientists in business and industry. The question remains whether or not the increased numbers of students in the lower level courses include a significant number who have the ability and stamina to succeed in the major.

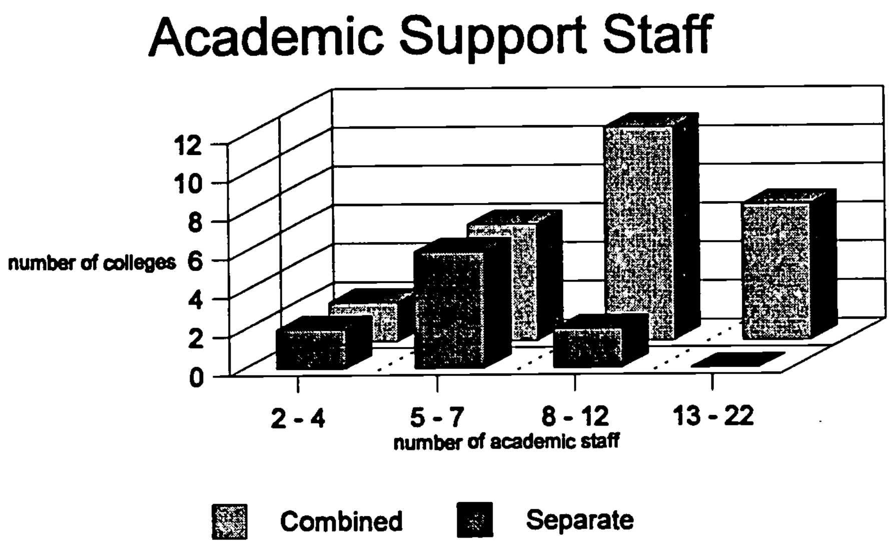
Before closing this section, I want to point out that there were only 3 schools with Management Information Systems (MIS) major programs among the colleges I visited. A reason for this may be that faculty are not taught traditional MIS topics, such as systems analysis and design, in graduate school. MIS faculty tend to come from business and industry and lack the academic credentials to teach at a liberal arts college. Also, MIS programs often operate out of Business departments which are not accepted in traditional liberal arts colleges. It is interesting to note that many of the Computer Science graduates of these colleges take MIS-type jobs after they graduate.

4. Support Staff

One of the most surprising results from this year's college visits was the fact that information technology support personnel for academics now greatly outnumber those dedicated to administrative computing. Ten years ago, just the opposite was true. In 1987, the median number of academic computing staff was 2 compared to a median of 5 for the administrative staff. This year the median for academic staff is 7 and that for administrative staff is 4.5. There are several reasons for this reversal in trends.

All but four of the colleges are using packages for administrative computing, so they need few, if any, programmers. Ten years ago more than half of the colleges had developed their own administrative software. Most of the data entry is being done by users, alleviating the need for data entry people in administrative computing. The advent of both local and wide area networks and the software to ensure easy access to and full use of these resources (the intranet) has created a need for network managers and assistants. These positions were not needed ten years ago. Since the network supports the academic mission and the number of networked computers on the academic side now far exceeds the number on the administrative side, I counted network staff under the academic umbrella.

I also noted that 31 of the schools had a single combined staff structure, whereas the other 10 had separated the staff into academic and administrative departments often reporting to different senior officers. Ten years ago, only 20 of the colleges had combined staffs. Many of the schools with a combined staff have blurred the academic/administrative staff responsibilities, choosing a functional breakdown as more efficient, i.e., one or more of the staff would do desktop support throughout the college, others would do training, others network support, or help desk, or ordering, etc.



Some schools had divided up the departments so a staff member would be primarily responsible for several departments which could be a mixture of academic and administrative. Being primarily responsible for a department meant that the department personnel would call that staff member when a problem occurred and the staff member would make sure that the problem was solved. In many cases this meant calling in one of the other staff members who were more expert in handling the given type of problem. Several colleges had sophisticated help desk software which kept track of the progress in solving problems, so nothing was allowed to fall through the cracks. The graph above illustrates that the schools with combined staffs tended to have a larger proportion of academic staff.

The advent of internet and intranet access throughout the campus and especially in the residence halls has brought with it its own set of support problems. Some of the colleges spend significant staff time hooking up students to the campus network, as much as six weeks of the semester. Other colleges do it all over a two-day period where students bring their computers to a central place for installation of network software and checkout. Colleges using the RCC model do not have to commit staff time to the actual hookup, but they do need to train the resident computer consultants. Many schools are reluctant to have staff open up a student-owned computer to install network cards unless the college is a certified repair shop for that computer. A few colleges charge students for network connection and use the money to pay students or staff working overtime, but this extra charge may discourage students from hooking up. The key to efficient installation seems to be widespread availability of well prepared documentation and software so that, in many cases, users can hook themselves up without assistance.

5. Equipment

The most striking impression I received from my college visits this year is that computer equipment is becoming viewed more and more as having a 3 - 5 year useful life with replacement of 1/5 - 1/3 of campus workstations being budgeted every year. Ten years ago, a computer was expected to last at least ten years and colleges which had invested heavily in campus-wide computing were sitting back and resting on their laurels. A few schools this year are still not committing to a regular replacement cycle. Instead, they depend on windfall replacement, i.e., they wait for a large donation or grant to come along before replacing obsolete computers. I found that 11 colleges replace their desktop units every five years; 21 replace them every four years; and 4 do so every three years.

One result of the 3 - 5 year replacement cycle is that most colleges have powerful up-to-date equipment in public access computer labs. Seventeen of the schools have a policy that brings the newest equipment into the public access areas for a year and then this equipment flows (cascades) to other users on campus. This policy shows that students have top priority and also ensures that the latest software will run in the public labs. A few colleges discourage cascading of computers from one user to another since a staff member must reconfigure the same system several times during its lifetime, thus draining scarce staff resources.

One school has delivery of systems down to a science. They spread out the ordering of new computers over the year in small batches. When an order arrives, the technicians install software in a converted classroom. Then the new users are brought in for group training, each on the machine that s/he will soon have in their office. Once the training is done, the computers are quickly delivered and systems to be cascaded are brought back for reconfiguring. The person to get the reconfigured system is brought in for training in the next wave. This avoids a large inventory of systems waiting for weeks to be configured and delivered.

Replacing equipment on a regular basis raises the question of what to do with obsolete systems. Colleges which repair their own systems (26 out of 41) often cannibalize the old systems for parts and junk the rest. Another approach is to refurbish the obsolete systems so that they are functional and give them to non-profit entities like schools or other agencies. Some schools sell the old computers at an auction or give them to their own faculty or students, which sometimes means that the obsolete systems reappear at the college and the new owners expect support.

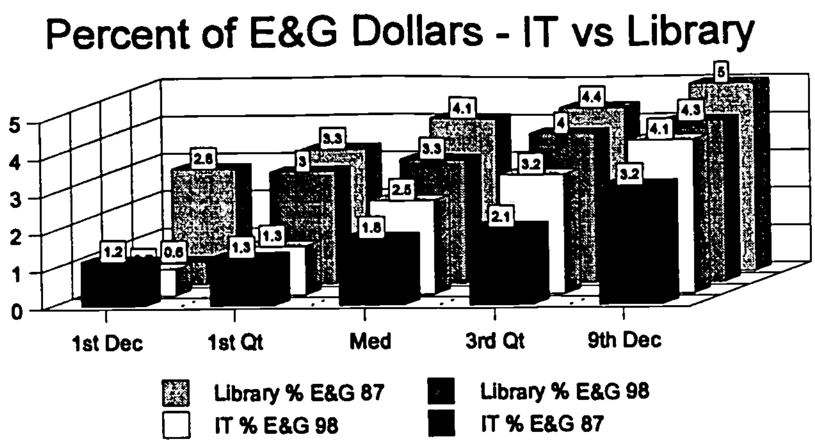
Colleges which do not repair their own desktops either outsource the job to a local repair shop (7 schools) or have some combination of inhouse repair and outsourcing (6 schools). The number of schools with technicians to repair equipment has almost doubled over the last ten years. Many schools purchase extended warranties with their new equipment, ensuring their repair by the manufacturer's local representative for two or three years, after which they can be replaced if they break down. Many schools (23) do not repair student-owned computers although they do provide some level of advice about what is wrong and where they can get it repaired. Colleges which sell computers to students (8) do provide repair service and 6 repair student-owned computers on a time and materials basis. The other 4 schools arrange to have a repair service come on campus to pick up and deliver student machines.

Some schools specify one or two network cards which students are required to use. A few schools provide free cards and cables. Others sell them in the bookstore. A few have computer stores where they offer network-ready systems at a discount. With competition from mail order resellers, it has been more and more difficult for college stores to afford to sell systems at an attractive price. Several schools are closing their computer stores as a result. If a student computer does not meet specifications to be network ready, a few colleges have the policy that staff will spend one hour trying to hook it up and then the student is on their own. It is very important to send specifications for network-ready computers to students and their families as early as possible before the start of the school year – before high school graduation for incoming first year students.

On a final note before closing this section, Apple Computers is alive and well in higher education regardless of how small a fraction of the microcomputer market they hold in the outside world. In the colleges I visited, Macintosh systems make up a median of 40% of the desktop systems. Several of the schools are now or recently had been Apple resellers. One advantage of Macintosh computers is the ease with which they can be connected to the network. Most schools, however, are making their new purchases from the Intel market, needing to hedge their bets in case Apple goes under.

6. Financial Results

Only one-third (14) of the schools I visited this year provided me with financial data concerning the comparison of the Educational and General (E&G) budgets for Computer Services and the Library vis a vis the total E&G budget of the school, and I am not certain that the data these schools provided can be fairly compared since different schools have different accounting procedures. Lines 20 - 27 of Appendix 2 and lines 17 - 25 of Appendix 3 give summaries of the financial results for the 1998 and 1987 surveys respectively. What is noteworthy is that, although the overall dollar figures have almost doubled in all categories for the library and nearly tripled for the computer center, the respective percentages of the total E&G budget have actually fallen over the ten years except for the mid to high end of computer center spending. When one talks to some administrators and faculty, one gets the impression that spending on information technology has skyrocketed out of control. At best, it has barely reached parity with library spending. In my 1987 paper, I quoted from Roger Haigh's 1983 article that an institution should spend at least as much on computer resources as on the library [Haigh]. With the expanded role of Information Technology in providing information resources to the college, this goal of parity between information technology and library spending is even more important today than it was 15 years ago. See the next page for a graphical illustration of the Information Technology and Library budgets from 1987 and 1998.



7. Key Issue

One question that still remains to be asked is to what extent investment in technology can improve teaching and learning. It is clear that today's citizens must become knowledge workers and must be good at accessing and processing data electronically. If students are to be encouraged to use technology in their education, faculty must first learn how to incorporate technology into their teaching. I asked each dean I talked to whether there were any incentives in place for faculty to use technology effectively in the classroom. No school had a policy which specifically mentioned technology use as a factor during the tenure and promotion process. Most deans told me that successful and appropriate integration of technology into their teaching and/or creative or scholarly work would be looked upon very favorably when faculty come up for review.

Nine of the colleges had external or internal grant money set aside for faculty to use to gain skill in using information technology. Some other schools had general grant money which could be sought for this purpose. It was very rare that a faculty member would get released time to prepare a technology-rich course or for any other reason. Not many schools have realized how much a risk faculty take when they use technology in their courses. If it is not used well or the equipment does not work correctly, students are ready to complain. Also, since students are very familiar with high-quality video game interfaces, they can be very critical of a faculty member's first attempt to bring technology into the classroom.

8. Conclusion

In this paper, I have tried to identify some questions and provide a range of answers which will further the planning effort at small colleges. Any school can ask the same questions (c.f. Appendix 4) and see where it stands by plotting its position in the table given in Appendix 2. In this way a college can see the areas that need improvement and take steps to change. Where I spotted trends, I pointed them out, but there is so much data that it was hard to identify all of them.

In my opinion, the stage is being set for an exciting experiment in learning with technology. Most colleges have wired the campus, established an intranet, and are committed to providing up-to-date equipment. Support staff levels are beginning to climb and better organization is making them more efficient. Faculty are observing how the use of technology has the potential to extend learning beyond the classroom, and students are accepting the challenge. I am looking forward to the changes the next ten years will bring to education and to the role that technology will play in bringing about those changes.

References

[ACM] Austing, R. H., et al., "Curriculum 78," ACM Curricular Recommendations for Computer Science, Volume I, Association for Computing Machinery, 1983, pp 59 - 78.

[ACM/IEEE] Allen, A. B., Barnes, B. H., et al., Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force, ACM Press, 1991.

[Gibbs and Tucker] Gibbs, N. E. and Tucker, A. B., "A Model Curriculum for a Liberal Arts Degree in Computer Science," Communications of the ACM, Volume 29, Number 3, March 1986, pages 202-210.

[Haigh] Haigh, R.W., "Evaluating Computer Resources," North Central Association Quarterly, Volume 60, Number 1, Summer 1985, pp 7-13.

[Leach & Smallen] Leach, K., Smallen, D. L., "What Do IT Support Services Really Cost?," http://www.hamilton.edu/personal/staff/dsmallen/cause97paper

[RESNET] http://www.resnet.emory.edu Annual conference to help colleges handle the problems associated with residence hall computing.

[Smith] Smith, P. D., "Computing Trends in Small Liberal Arts Colleges," SIGCSE Bulletin, Volume 20, Number 1, February, 1988, pp. 299-303.

Appendix 1
List of Colleges Visited

1987	1997-98
Allegheny College, PA	* Alma College, MI
* Alma College, MI	* Augustana College, IL
* Augustana College, IL	* Beloit College, WI
Bates College, ME	Butler University, IN
Beloit College, WI	Coe College, IA
Butler University, WI	Connecticut College, CT
* Carleton College, MN	Cornell College, IA
Coe College, IA	* Denison University, OH
* Cornell College, IA	* Depauw University, IN
* Denison University, OH	Dickinson University, PA
Depauw University, IN	Earlham College, IN
* Earlham College, IN	* Gettysburg College, PA
Franklin and Marshall College	Goucher College, MD
* Gettysburg College, PA	* Hanover College, IN
Hanover College, IN	Hollins College, VA
Hillsdale College, MI	Holy Cross College, MA
* Holy Cross, College of the, MA	Hope College, MI
* Hope College, MI	Kalamazoo College, MI
* Kalamazoo College, MI	Kenyon College, OH
Kenyon College, OH	* Knox College, IL
* Knox College, IL	* Lake Forest College, IL
Lake Forest College, IL	Lawrence University, WI
* Lawrence University, WI	Macalester College, MN
Macalester College, MN	Millikin University, IL
* Millikin University, IL	Muhlenberg College, PA
* Monmouth College, IL	Randolph Macon Womens College, VA
* Muhlenberg College, PA	* Ripon College, WI
New Rochelle, College of, NY	St. Catherine, College of, MN
* Ripon College, WI	St. John's / St. Benedict, MN
* Russell Sage Women's College, NY	* Saint Mary's College, IN
* St. Catherine, College of, MN	St. Norbert's College, WI
* Saint Mary's College, IN	* St. Olaf College, MN
* St. Norbert's College, WI	Salem College, NC
* St. Olaf College, MN	Skidmore College, NY
Skidmore College, NY	Stonehill College, MA
* Stonehill College, MA	Washington & Jefferson College, PA
Washington & Jefferson College, PA	Washington & Lee University, VA
Wheaton College, MA	Wells College, NY
* Williams College, MA	* Wheaton College, MA
* Wittenberg University, PA	* Wittenberg University, PA
Wooster, College of, OH	Wooster, College of, OH

* marks the colleges which provided financial data

Appendix 2 (1997-98 Visits)

		First Decile	First Quart	Median	Third Quart	Last Decile	Mean
1.	Enrollment	720	1160	1600	2158	2905	1694
2.	Number of Public Access Micros	72	90	126	172	248	142
3.	Mac % of Public Access Micros	9	16	42	60	76	38
4.	Number of Students per Micro	7	9	11	15	21	12.4
5.	PA Micro Hours/Student/Week	5	6	10	12	16.5	10
6.	CS Faculty FTE	1	1.5	2	3	4	3
7.	Admin Comp Center Staff FTE	1	2	4.5	6	11	4.9
8.	Acad Comp Center Staff FTE	4	5.5	7	12	14	8.6
9.	Acad Staff % of Total CC staff	48	50	70	75	80	63.5
10.	Number of Students per CC Staff	75	100	120	167	219	125
11.	Number of Faculty per Acad CC Staff	11	13	17	21	29	16.8
12.	Graduating CS Majors (0 if no CS Major)	0	3	6	10	13	6
13.	% CS Graduates of Student Body (" " ")	0	0.2	0.4	0.5	0.9	0.4
14.	CS Graduating Majors per CS Faculty(" ")	0	1.5	3	4.3	5.5	3
15.	Percent of Students Owning Micros	13	25	40	64	80	46
16.	Year Residence Halls Networked	94	96	97	98	99	97
17.	Desktop Replacement Cycle (Years)	3	4	4	5	7	4
18.	Student Comp Center Workers	14	22	30	40	60	31
19.	Student Assistant % of Student Body	1	1.5	1.9	2.6	3.3	2

Financial Data for 14 Schools (I was unable to obtain this data at the other schools)

		First Decile	First Quart	Median	Third Quart	Last Decile	Mean
20.	Enrollment	880	1160	1415	2195	2604	1658
21.	Total E&G Expenditures	13750	21300	27600	40000	47450	30000
22.	Library E&G Expenditures	333	696	1000	1295	1925	948
23.	Computer Center E&G Expenditures	246	311	649	1001	1295	685
24.	Library Percent of E&G Dollars	0.65	2.95	3.35	4.05	4.3	3.2
25.	Computer Center % of E&G Dollars	0.75	1.35	2.45	3.2	4.1	2.4
26.	Library E&G Dollars / Student	97	497	585	816	923	592
27.	Computer Center E&G Dollars / Student	95	278	496	672	644	437

Values in lines 21, 22, and 23 are in thousands of dollars.

Appendix 3 (1987 Visits)

		First Decile	First Quart	Median	Third Quart	Last Decile
1.	Enrollment	950	1080	1670	2000	2400
2.	Number of Public Access Micros	11	18	37	55	80
3.	Public Access Dumb Terminals	7	19	26	39	69
4.	Number of Students per Micro	19.6	28.2	50.1	76.9	154
5.	Public Access Micro Hours/Student/Week	0.6	1.1	2.0	3.3	5.0
6.	Students/Dumb Terminal	19.2	32.7	50.1	76.9	150
7.	Dumb Terminal Hours/Student/Week	0.3	0.8	1.7	2.7	4.5
8.	CS Faculty FTE	1.5	2	2.5	4	5
9.	Admin Comp Center Staff FTE	2	3	5	7.5	12.0
10.	Acad Comp Center Staff FTE	0.5	1	2	3	5
11.	Acad Staff % of Total CC staff	13	20	28	38	57
12.	Number of Students per CC Staff	118	167	209	280	611
13.	Graduating CS Majors (26 Schools)	4.5	5	9	11	17
14.	Faculty FTE in CS Major Programs	2	2	3.5	4.3	5.5
15.	% CS Graduates of Student Body	0.3	0.4	0.6	0.9	1.0
16.	CS Graduating Majors per CS Faculty	1.4	1.9	2.6	3.7	5

	Financial Data for 25 Schools	Values are averages for 3 years 6/83-6/86				
17.	Enrollment	884	1039	1705	2095	2372
18.	Total E&G Expenditures	8436	10659	13629	18411	26740
19.	Library E&G Expenditures	287	393	529	672	1002
20.	Computer Center E&G Expenditures	116	177	231	325	722
21.	Library Percent of E&G Dollars	2.8	3.3	4.1	4.4	5
22.	Computer Center % of E&G Dollars	1.2	1.3	1.8	2.1	3.2
23.	Library E&G Dollars / Student	216	271	311	450	610
24.	Computer Center E&G Dollars / Student	92	117	143	243	370
25.	Hardware Dollars/Student	14	41	81	207	404

Values in lines 18, 19, and 20 are in thousands of dollars.

Appendix 4
Questionnaire

Comparative College Visitation Questions: College Name ___________

I. Data

1. FTE enrollment______ men________ women_______
2. CS major? ______ # CS seniors ________ #CS faculty FTE_____
3. Info-Science/MIS major ____ # seniors ____ Minor programs? _____
4. #faculty w/CS PhD _____ # CS faculty w/o CS PhD ______
5. Central or distributed public computer labs __________

For each public access lab
   a. # PCs _______ # Macs ________ # terminals _____
   b. # hours/week available (not counting scheduled class hours) ______
   c. Networked? ______ If yes - to Internet? ____ LAN only? _____
   d. # hours/week supervision: student worker ________ staff _______

6. Is there a fine arts computer facility? ________
7. What type of computer is used for administrative work? ______________
8. What administrative package is used? ________ Central Database? _______
9. # Info Tech staff ______ # acad comp staff_______ # admin comp staff _______
10. Are staffs separate? ____
11. # students bringing computers to campus ______
12. Are residence halls networked to the internet? If so, when did/will it happen ____
13. Does your campus network provide public file space? ____ Software metering?
    Megs of faculty space _____ Megs of student space ______
14. Are students required to own or buy computers? _____
15. Do faculty/staff have access from off campus to college network? ____ How?

II. Open-ended questions

1. If you have a CS/MIS major, which curriculum do you follow? Are you accredited?
   Did other majors decline as your CS/MIS major gained students?
   Did instituting a CS/MIS major lead to increase in college enrollment?
2. Do you have CS/MIS minors? What are the requirements?
3. What are the enrollment trends in your CS courses? In the CS major?
4. How is innovative use of IT weighted in tenure/promotion decisions? Basis?
5. What is the repair policy for student-owned computers? For college-owned computers
6. Is Computer Services a separate entity or is it subsumed into another admin. Unit?
7. What is the policy for maintaining/upgrading computers? Obsolete computers?
8. Do you have a cascade or "hand-me-down" policy for college computers? What is it?
9. What is your policy on laser printing?
10. How are you dealing with support issues across the board?

E&G Budget _________ IT E&G Budget ________ Library E&G Budget ______

Nothing But 'Net? A Cautionary Tale of Teaching in the Electronic Village

Robert L. Sedlmeyer
Indiana University Purdue University Fort Wayne
Fort Wayne, IN
sedlmeyer@ipfw.edu

Abstract

Does the Internet herald the end of teaching as we know it? Will the college professor be transformed from a sage on the stage (or a guide on the side) to a celeb on the web? This presentation will review the my experiences in utilizing the Internet to teach a variety of computer science courses. The discussion will focus on what impact the Internet has had on lecture content and presentation, delivery and management of student projects, student-teacher interaction, preparation time, and student performance.

(Robert's paper was not available at the time the Proceedings went to press. He will supply copies of his paper at his talk.)

Using 4DOS batch files to create an infrastructure that makes it easy for students to create and maintain HTML Web Pages

Robin M. Snyder
Byrd School of Business
Shenandoah University
Winchester, VA 22601
rsnyder@su.edu

ABSTRACT

Learning HTML is not that difficult. Neither is using an HTML editor. Many of the practical problems in having students create and maintain HTML web pages are a result of a lack of a suitable infrastructure. This paper (and presentation) will describe (and demonstrate) the use of simple 4DOS batch files to automate the creation and maintenance of such a system. Although 4DOS supports quite sophisticated batch processing, many common tasks involving multiple users, directories, and files can be done by taking the supplied batch file templates and making simple modifications to them. This paper (and presentation) would be of interest to anyone, including network administrators, who want to make it easy for students to create and maintain HTML web pages, or who wish to automate some of the many tasks that must be done for each and every user, directory, or file on the network.

WEB PAGES

Web pages are text files that are written in a text formatting language called HTML. It is assumed that the reader is somewhat familiar with HTML files. HTML files are accessed on the Internet via their URL (Uniform Resource Locator). The URL for Shenandoah University is

`http://www.su.edu`

which refers to the University's web server running Microsoft Windows NT. All clients run Windows 95 while the file servers run Novell NetWare. The author's web directory on the Internet is in the WEB directory on the KITHARA: volume of the NetWare file server called STUDENT. When logged onto the network, this directory is mapped as the DOS directory S:\WEB, as the DOS directory (mapped as root) W:\, as the NetWare directory STUDENT/KITHARA:WEB and as the Internet directory `http://www.su.edu/kithara`. Thus, when logged into the STUDENT NetWare file server as member of group KITHARA, the following all refer to the same directory.

```
S:\WEB
W:\
STUDENT/KITHARA:WEB
http://www.su.edu/kithara
```

It is a simple matter for the network/web administrator to map the Internet URL

`http://www.su.edu/kithara`

to the NetWare directory STUDENT/KITHARA:WEB. In this case, the Windows NT web server is attached to the NetWare server as user/process WEBMASTER and has "**Read**" and "**File Scan**" rights in that directory. Thus, instead of using FTP (file transfer program) to put files on the web server, a simple file copy operation can be used by both teacher and students.

Each student is assigned a subdirectory in this web subdirectory with appropriate rights. Drive mappings provide each student convenient access to his or her own directory, while a web page in the root provides convenient access to any student web page for Internet browsers.

There are many ways to automate the required NetWare mappings. The following NetWare login script is used for Kithara network users.

```
IF FILE_SERVER = "STUDENT" THEN BEGIN
   IF MEMBER OF "KITHARA" THEN BEGIN
      WRITE "BEGIN KITHARA login script..."
      MAP S:=STUDENT/KITHARA:
      MAP ROOT U:=STUDENT/KITHARA:USER/%LOGIN_NAME
      MAP ROOT W:=STUDENT/KITHARA:WEB
      WRITE "END KITHARA login script."
      END
   END
```

Alternatively, if login scripts are not used, the student (or teacher) can create the drive mapping once and specify that the mapping is to be reconnected at login. (This is a feature of Windows 95).

However, creating and supporting the necessary infrastructure to make it easy for students (and teachers) to use this capability without having to know too much about the process is the topic of the rest of this paper.

ISN'T DOS DEAD?

To most users, DOS means the command line interface found in the program `command.com`. Let's face it. DOS was usable, but not easy or friendly. Since 1986, sophisticated users have used 4DOS in the form of `4dos.com` to replace `command.com` as their command-line processor. The advantage of 4DOS is that the command set and user interface is a large superset of DOS, providing most of the conveniences that have been provided in UNIX command line interfaces for several decades. 4DOS is very compatible with DOS, the only differences being in areas where the makers of 4DOS, JP Software, Inc., had to decide whether to duplicate certain bugs (or features) in DOS. For example, should 4DOS duplicate what the official documentation says `command.com` does or what `command.com` actually does.

4DOS is shareware that is available from JP Software, Inc. Additional information is available at their web site at http://www.jpsoft.com. And, since JP Software has similar versions available for OS/2, Windows NT, and Windows 95, the same batch files can be used with those systems.

But, why are we talking about DOS? Isn't DOS dead? Well, yes and no. Few users use DOS, and those that do use it to run legacy programs that still serve some useful purpose. But, some form of batch processing is still required in any system, unless one is willing to do every task, no matter how many times it is to be done, via point and click and keyboard input.

Programming languages can be used to automate such tasks. Visual Basic is the primary systems language in Windows. REXX is the primary systems language in OS/2 (and other IBM systems). Java is fast becoming a viable alternative for certain types of batch processing. Products such as

Novell GroupWise and IBM Lotus Notes/Domino have made their API (Applications Programmer Interface) available through Java applications (but not via applets, since there are additional security restrictions on applets that are not present for applications). But, for many batch tasks, 4DOS batch files fit the bill quite nicely, the other alternatives being just to complicated and difficult to develop and maintain for certain tasks.

BATCH FILES

A DOS batch file is a text file that contains DOS commands. A 4DOS batch file is a text file that contains 4DOS commands. The default file extension of a DOS batch file is `.bat`. The default file extension of a 4DOS batch file is `.btm`, which stands for batch file executed in memory. 4DOS loads the entire batch file into memory and executes it there rather than the DOS approach of loading each line from the file as it is needed, so that changes in the file during execution modify the behavior or the batch file. For example, the following batch file outputs the text "Hello, World.".

```
@ECHO OFF

ECHO Hello, World
```

The `@ECHO OFF` turns the ECHO off after this statement, and the `@` causes the ECHO not to be echoed. A 4DOS setting fixes this feature (or bug) so that every batch file does not have to begin with `@ECHO OFF`.

An ASCII text editor such as Notepad can be used to edit text files, whether they be data files or batch program files. The task can be simplified by creating a shortcut on the desktop with a command line that invokes Notepad and opens the desired file. For example, the following command invokes Notepad and opens the batch file `F:\USERS\hello.btm`.

```
C:\WINDOWS\notepad.exe F:\USERS\hello.btm
```

This command assumes that the Windows directory is `C:\WINDOWS`.

Running a batch file is simple. If the batch file were `F:\USERS\hello.btm`, the command to run the batch file would be `F:\USERS\hello.btm`. A shortcut to run the batch file would have the following command line.

```
G:\4DOS600\4dos.com /C F:\USERS\hello.btm
```

This command assumes that `4dos.com` is in the directory `G:\4DOS600`.

The redirection character "`>!`" can be used to redirect the output of a batch file to another file. For example, the command

```
F:\USERS\hello.btm >! F:\USERS\hello.out
```

causes the output of `hello.btm` to be sent to the file `hello.out`, overwriting any previous `hello.out` file.

We will now look at 4DOS batch files from the point of view of making it easy for students to create and maintain HTML web pages. The extensions to user and group management in a network environment are intimately related, as we shall see. Only those features of 4DOS that are necessary for this goal will be covered, leaving many other useful and powerful 4DOS features unmentioned.

ASSUMPTIONS

We have to start somewhere, so let's make the following assumptions.

A teacher (or network administrator) is going to create such an infrastructure to support students (or users) on a network which is connected to the Internet.

To avoid overly complicating the system, the removal of students from the system will not be addressed, but is left as an exercise for the reader. In practice, since it can be difficult to determine manually which directories should be deleted, the system will just grow larger until provisions are made to reduce the size of the directories.

The list of student names and userids for the students are in a text file, one per line, separated by the colon character ":". This text file could be created manually or created as the output of a database query or spreadsheet program. The text file `F:\USERS\users.dat` will be used for example purposes and appears as follows.

```
Snyder    : Tamara     : TAMARA
Snyder    : Gregory    : GREGORY
Snyder    : Cory       : CORY
Snyder    : Luke       : LUKE
Greenly   : Emily      : EMILY
```

For convenience, the user last name, first name, and userid are on the same line. This means that we will need to separate them in order to process the userid separately from the user name. The last name appears first so that the file can be easily sorted by last name.

The first batch file uses the environment variable G and the FOR command to ECHO every line of the text file `users.dat`.

```
FOR %G IN (@F:\USERS\users.dat) ECHO %G
```

The environment variable `F` is set to each of the lines of the text file and then echoed to the screen using the ECHO command.

But, in most cases, more than one statement is needed to accomplish the desired purpose. The previous program can be written as follows.

```
FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

QUIT

:PROCESS
```

```
ECHO %G
RETURN
```

We now need to separate the user name from the userid. The environment variable U will be used for the userid and the environment variable N will be used for the name. The following program does such a separation and outputs just the userid.

```
FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

QUIT

:PROCESS
   SET I=%@INDEX[%G,:]
   SET L=%@TRIM[%@SUBSTR[%G,0,%I]]
   SET F=%@TRIM[%@SUBSTR[%G,%@EVAL[%I+1],255]]
   SET I=%@INDEX[%F,:]
   SET U=%@TRIM[%@SUBSTR[%F,%@EVAL[%I+1],255]]
   SET F=%@TRIM[%@SUBSTR[%F,0,%I]]
   GOSUB :PROCESS1
   RETURN

:PROCESS1
   ECHO Last="%L"  First="%F"  UserID="%U"
   RETURN
```

The statements in PROCESS look imposing, but they are just used to split the line of text and set the environment variables N and U. Note that a feature of 4DOS is that since the comma "," is used to delimit arguments to SUBSTR, a comma cannot appear in the text file, as might be desired if the names were to be of the following form.

<last> , <first>

The output is as follows.

```
Last="Snyder"  First="Tamara"     UserID="TAMARA"
Last="Snyder"  First="Gregory"    UserID="GREGORY"
Last="Snyder"  First="Cory"       UserID="CORY"
Last="Snyder"  First="Luke"       UserID="LUKE"
Last="Greenly" First="Emily"      UserID="EMILY"
```

Note that we are not concerned with lining up the output. Note also the introduction of the procedure PROCESS1. To save space, unless otherwise specified, all program code will refer only to the body part of the PROCESS1 procedure of the program. The PROCESS1 statements of the above program are as follows.

```
ECHO Last="%L" First="%F"  UserID="%U"
```

The following PROCESS1 statements create a user and web directory for each student, if that directory does not already exist.

```
IFF NOT ISDIR D:\WEB\%U THEN
      ECHO Making directory D:\WEB\%U
      *MD D:\WEB\%U
      ENDIFF

   IFF NOT ISDIR D:\USER\%U THEN
      ECHO Making directory D:\USER\%U
      *MD D:\USER\%U
      ENDIFF
```

But, just making the directory is not enough. Appropriate rights must also be given to the user in that directory. The following program grants the "**Read**", "**Write**", "**Create**", "**Erase**", "**Modify**", and "**File Scan**" rights to the user, all but the "**Access Control**" right.

The following `PROCESS1` statements grant the appropriate rights for each student.

```
ECHO.
   E:\PUBLIC\grant.exe R W C E M F FOR D:\WEB\%U TO USER %U
   E:\PUBLIC\grant.exe R W C E M F FOR D:\USER\%U TO USER %U
```

Note that many networks store the `grant.exe` command on the `Z:` drive.

This batch file is the same as issuing the following commands.

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\TAMARA TO USER TAMARA
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\TAMARA TO USER TAMARA

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\GREGORY TO USER GREGORY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\GREGORY TO USER GREGORY

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\CORY TO USER CORY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\CORY TO USER CORY

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\LUKE TO USER LUKE
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\LUKE TO USER LUKE

E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\WEB\EMILY TO USER EMILY
E:\PUBLIC\GRANT.EXE R W C E M F FOR D:\USER\EMILY TO USER EMILY

Note that the `ECHO.` command is used to output a blank line.

Obviously, the batch method scales up to hundreds of users with very little additional effort. That is exactly the purpose of repetition and looping actions in programming.

Note that, in Novell NetWare, if a user were given the "**Access Control**" right, the user could inadvertently use that right to disallow themselves access to information in subdirectories of the directory to which they were granted rights. (Is this a feature or a bug?). In addition, in order to maintain individual responsibility for web directories, it is not advisable to allow users to permit other users to place information in directories that appear on the Internet.

MASTER INDEX

The next step is to create a master index as the file `D:\index.htm`. This can be accomplished with the following program.

```
ECHO `<HTML>`
ECHO `<HEAD>`
ECHO `<TITLE>Kithara students</TITLE>`
ECHO `</HEAD>`
ECHO `<BODY>`
ECHO `<H1>Kithara students</H1>`
ECHO `<UL>`

FOR %G IN (@F:\USERS\users.dat) GOSUB :PROCESS

ECHO `</UL>`
ECHO `</BODY>`
ECHO `</HTML>`

QUIT

:PROCESS
   SET I=%@INDEX[%G,:]
   SET L=%@TRIM[%@SUBSTR[%G,0,%I]]
   SET F=%@TRIM[%@SUBSTR[%G,%@EVAL[%I+1],255]]
   SET I=%@INDEX[%F,:]
   SET U=%@TRIM[%@SUBSTR[%F,%@EVAL[%I+1],255]]
   SET F=%@TRIM[%@SUBSTR[%F,0,%I]]
   GOSUB :PROCESS1
   RETURN

:PROCESS1
   ECHO `<LI>`%L, %F - `<A` HREF="./%U/index.htm"`>`%U`</A>`
   RETURN
```

Note that since "<" and ">" are special characters (for redirection) in DOS, those characters need to be enclosed in backquotes (i.e., all but environment variables can be enclosed in backquotes).

The output is as follows.

```
<HTML>
<HEAD>
<TITLE>Kithara students</TITLE>
</HEAD>
<BODY>
<H1>Kithara students</H1>
<UL>
<LI>Snyder, Tamara - <A HREF="./TAMARA/index.htm">TAMARA</A>
<LI>Snyder, Gregory -<A HREF="./GREGORY/index.htm">GREGORY</A>
<LI>Snyder, Cory - <A HREF="./CORY/index.htm">CORY</A>
```

```
<LI>Snyder, Luke - <A HREF="./LUKE/index.htm">LUKE</A>
<LI>Greenly, Emily - <A HREF="./EMILY/index.htm">EMILY</A>
</UL>
</BODY>
</HTML>
```

Suppose that this program is called `F:\USERS\makelist.btm`. The following 4DOS command will run the batch file with the output redirected to the file `D:\WEB\index.htm`.

```
F:\USERS\makelist.btm >! D:\WEB\index.htm
```

Note that the command line from an icon shortcut would be as follows, where `4dos.com` is located in the directory `G:\4DOS600`.

```
G:\4DOS600\4dos.com /C F:\USERS\makelist.btm >! D:\WEB\index.htm
```

Now, anyone who can access the file `D:\WEB\index.htm` has convenient access to any of the student web pages.

Suppose that each student is supposed to create a web page as the file `index.htm` in their web directory (i.e., the directory created by the above program). Which students have web pages? The following `PROCESS1` statements display the userids of those students who do not have an `index.htm` file.

```
IFF NOT EXIST D:\WEB\%U\index.htm THEN
     ECHO User %U does not have a web page.
     ENDIFF
```

Note that we can assume that the directory does exist, because we just ran the previous program that updated the creation of the directories.

But, why not give users a default web page, if one does not already exist? The following `PROCESS1` statements creates a default web page for the student if and only if that student does not already have a web page. That is, we don't want to erase any of their previous work.

```
IFF NOT EXIST D:\WEB\%U\index.htm THEN
     ECHO Making web page for %U as D:\WEB\%U\index.htm.
     CALL F:\USERS\makepage.btm >! D:\WEB\%U\index.htm
     ENDIFF
```

The batch file `makepage.btm` can assume that the environment variables `U` and `N` have been set properly.

The batch file `makepage.btm` appears as follows.

```
ECHO `<HTML>`
ECHO `<HEAD>`
ECHO `<TITLE>`%F %L`</TITLE>`
ECHO `</HEAD>`
```

```
ECHO `<BODY>`
ECHO `<H1>`%F %L`</H1>`
ECHO `<UL>`
ECHO `<LI>`UserID: %U
ECHO `</UL>`
ECHO `<HR>`
ECHO `<A HREF=../index.htm>Back to student list</A>`
ECHO `</BODY>`
ECHO `</HTML>`
```

The default home page for user TAMARA appears as follows.

```
<HTML>
<HEAD>
<TITLE>Snyder  Tamara</TITLE>
</HEAD>
<BODY>
<H1>Snyder  Tamara</H1>
<UL>
<LI>UserID: TAMARA
</UL>
<HR>
<A HREF=../index.htm>Back to student list</A>
</BODY>
</HTML>
```

CONCLUSIONS

4DOS is an easy and powerful to create an infrastructure that makes it easy for students to create and maintain HTML web pages. This article has just touched on the use of 4DOS to automate such tasks.

Using frames and JavaScript to automate teacher-side web page navigation for classroom presentations

Robin M. Snyder
Byrd School of Business
Shenandoah University
Winchester, VA 22601
rsnyder@su.edu

ABSTRACT

HTML provides a platform-independent way of creating and making multimedia presentations for classroom instruction and making that content available on the Internet. However, time in class is very valuable, so that any way to automate or otherwise assist the presenter in web page navigation during class can save valuable seconds. This paper (and presentation) will describe (and demonstrate) the use of frames and JavaScript in conjunction with VGA/TV project systems to automate teacher-side web page navigation for classroom presentations. The methods used are not visually distracting to the students during class, do not require web server support or an Internet connection (i.e., it can be used with only local hard drive access), and can be used independently of the student-side web page access so that students who may or may not have browsers that support the features used on the teacher side can still access the content of those pages.

THE WEB SYSTEM

We will assume that the the teacher has a way of creating presentations or other web-based material in HTML format that is to be presented in class using a web browser such that the students can see the presentation. There are many ways to display the screen such that it can be seen by students. The author uses the low-cost way of using one or more 27-inch TVs that display the screen output of a laptop computer via a SVGA-to-TV converter.

The entire web system is designed such that all files in the web system, except for an `index.htm` file in the root that requires special handling, are in directories that are immediately below the root. Thus, there is only one level of subdirectories, no other subdirectories. Thus, any HTML file can reference any other file within the web system by first going up to the parent directory (i.e., the root) via "..", then down to the desired directory. For technical reasons related to insuring web system integrity, this is done even if the referenced file is in the same directory as the referencing file. This design principle makes the entire web system simple and entirely relocatable. As there are no absolute file references, the web system on the local hard drive is the same as the web system on the network is the same as the web system as seen from the Internet. This provides convenience in developing and updating the web systems. For example, even if the Internet and the network go down, the class can proceed by using the copy on the local hard drive. In addition, throughput is improved since access to the local hard drive is always faster than access to files on the network which is always faster than access to files on the Internet.

FRAMES AND SCREENS

A frame provides a way to divide a browser window into several panes. Thus, more than one web page can be displayed at the same time. More importantly, events in one frame can cause other frames to be reloaded with new pages.

There are many way to arrange frames. The author uses a SVGA-to- TV converter that takes the 800 by 600 SVGA output from the laptop computer and converts it for TV output, which can handle a size of about 600 by 440, slightly smaller than a standard 640 by 480 VGA display. Unless the entire 800 by 600 screen is to be displayed to the students, making most of the text on the screen hard to read, the 600 by 440 area in the upper right of the 800 by 600 screen is displayed to the students. This leaves a margin of 200 pixels on the left and 160 pixels on the bottom that can be used for special navigation features.

The Windows 95 taskbar provides a move convenient way to switch between tasks in a classroom setting over using the `Alt-Tab` feature which can be visually distracting. This leaves about 110 pixels on the bottom part of the screen for navigation controls.

Application programs such as Excel and Word can be customized such that toolbars and buttons are arranged in this navigation area. But, the topic of this paper is how frames and JavaScript can be used to provide automate teacher-side web page navigation for classroom presentations.

The following is a simple HTML file.

```
<HTML>
<HEAD>
<TITLE>Title bar caption goes here</TITLE>
</HEAD>
<BODY>
Text, images, etc., to be browsed goes here.
</BODY>
</HTML>
```

The following is the frame file used by the author. This file is specified as the home page file in NetScape and Internet Explorer. Note that there is no `BODY` section when using frames.

```
<HTML>
<HEAD>
<TITLE>Kithara System, by Robin Snyder</TITLE>
</HEAD>
<FRAMESET COLS="15%,85%">
   <FRAME NAME="LEFTFRAME"
      SRC="../RMS1.WEB/left.htm"
      SCROLLING="no" MARGINWIDTH=0>
   <FRAMESET ROWS="80%,20%">
      <FRAME NAME="MAINFRAME"
      SRC="../RMS.WEB/index.htm">
      <FRAME NAME="BOTFRAME"
      SRC="../RMS1.WEB/bottom.htm"
      SCROLLING="no" MARGINWIDTH=0>
   </FRAMESET>
</FRAMESET>
</HTML>
```

This setup allocates the frame called LEFTFRAME to the leftmost 15% of the screen area, and the remaining 75% is allocated to the frameset that consists of the frame called MAINFRAME which has the upper 80% of that area and the frame called BOTFRAME that has the lower 20% of that area.

Note that the title

```
Kithara System, by Robin Snyder
```

is the title that appears on the title bar. Such a title can be useful in locating, via program code, the appropriate window to switch to if such switching is to be automated.

The MAINFRAME is used to display the presentation to the students. Command buttons that use JavaScript to provide navigation among the author's web system are placed into the LEFTFRAME and BOTFRAME frames. Initially, LEFTFRAME will display the HTML file left.htm, BOTFRAME will display the HTML file bottom.htm, while MAINFRAME will display index.htm, the author's home page.

To conserve precious screen space, neither LEFTFRAME or BOTFRAME have SCROLLING set to NO (i.e., no scrollbars) and the MARGINWIDTH is set to 0. The teacher needs to insure that it is not necessary to scroll these frames. This brings up an important design consideration. While the pages displayed in MAINFRAME will be accessed by students and should therefore contain portable HTML formatting code, everything in LEFTFRAME or BOTFRAME will be used only by the teacher. Therefore, these frames need not be portable but need only work on the teacher's computer. Thus, if the JavaScript code used works only with NetScape and not with Internet Explorer, this is not a problem since the teacher will be using only NetScape for presentations and not Internet Explorer.

The TARGET feature of HTML links can be used to change a page in a frame. However, as we will see, it turns out to be more generally useful to use input buttons and JavaScript with NetScape, since functionality in addition to just loading a page can be added. Similar things could be done using VBScript with Internet Explorer, but that is beyond the scope of this paper.

HTML FORMS

Input buttons and text boxes must be placed on a form. Since these are only for navigation purposes, we do not need a SUBMIT button. The FORM tag has the following form and is placed within the BODY section of the HTML file.

```
<FORM
   NAME="FORM1"
   ACTION="mailto:rsnyder@su.edu"
   METHOD=POST
   >

</FORM>
```

Since we are not using the ACTION or METHOD attributes, the preceding can be simplified to the following.

```
<FORM NAME="identifier">

</FORM>
```

The NAME attribute specifies the "**identifier**" by which JavaScript refers to the form. The author uses the name FORM1, since this is the only form of concern here.

The INPUT tag is used to place an input button on the form and has the following form.

```
<INPUT
   TYPE="BUTTON"
   VALUE="caption"
   NAME="identifier"
   ONCLICK="function"
   >
```

The NAME attribute specifies the "**identifier**" by which JavaScript refers to the button. The VALUE attribute specifies the "**caption**" text which is displayed on the button. The ONCLICK attribute specifies the JavaScript "**function**", with parameters, that is called when the button is clicked.

The INPUT tag is also used to place a text box on the form and has the following form.

```
<INPUT
   TYPE="TEXT"
   NAME="identifier"
   SIZE=36
   >
```

The NAME attribute specifies the "**identifier**" by which JavaScript refers to the text box. The SIZE attribute specifies the width of the text box.

SCRIPTS

JavaScript is a programming language whose commands work within a browser to perform tasks involving HTML web pages and the Internet. For security reasons, both JavaScript is extremely limited as to what can be written to the the local hard drive.

The SCRIPT tag is used to specify the use of JavaScript.

```
<SCRIPT LANGUAGE="JAVASCRIPT"
   SRC="../RMS1.WEB/tswitch.js">
</SCRIPT>
```

Since the navigation capabilities to be added are being used by the teacher and not the students, it is only necessary that the LEFT and BOTTOM frames work with whatever browser the teacher is using. Since both of these frames will use the same JavaScript run-time system, the SCR attribute is used to refer to the JavaScript text file `tswitch.js` that contains the JavaScript code. Since the code consists only of function definitions and, thus, need not be executed as the page is loading, the `SCRIPT` tag can be placed either in the `HEAD` or `BODY` of the HTML file.

JAVASCRIPT

The functions that comprise the JavaScript runtime system are now presented, along with an example, where appropriate, of the function call that uses that function from the HTML page.

Whenever a page is loaded into another pane of a frame, the user must click on that pane to give it the focus so that the page up/down keys can be used for navigation purposes. The `set_focus` function is used, where deemed appropriate, to set the focus to the `MAINFRAME` pane, saving a mouse click on that pane whenever a button is clicked.

```
function set_focus() {
   top.MAINFRAME.focus()
   }
```

WARNING: JavaScript is case sensitive, so `focus` is not the same as `Focus`. Beware.

Notice that `top` is the top-level page, `MAINFRAME` is one of the panes in the frame, and `focus` is a method of the frame pane.

A common use of a button is to go to a specified page. The function `goto_page` is used to go to page `s`.

```
function goto_page(s) {
   top.MAINFRAME.document.location = s
   set_focus()
   }
```

The `document.location` of `MAINFRAME` is set to `s`, loading that page, and then the `MAINFRAME` is given the focus. For security reasons, this loading only works from the local hard drive, but that is the intended use of the function.

An example of an `INPUT` button tag is as follows.

```
<INPUT
   TYPE="button"
   VALUE="Home"
   NAME="Home"
   ONCLICK="goto_page('../RMS.WEB/index.htm')"
   >
```

When clicked, this button will go to the author's local home page at `../RMS.WEB/index.htm` and set the focus to the `MAINFRAME` pane.

When using frames, it can be difficult to determine what file is loaded into the `MAINFRAME`. A text box is used to MARK the currently loaded page, displaying the URL in a text box. A GO button provides a way that the teacher can use to return to this page. The HTML tags to create a MARK button, a text box, and a GO button are as follows:

```
<INPUT
   TYPE="button"
   VALUE="Mark"
   NAME="MARK1"
   ONCLICK="mark_url1()">

<INPUT
   TYPE="TEXT"
   NAME="TEXT1"
   SIZE=36>

<INPUT
   TYPE="button"
   VALUE="Go"
   NAME="GOTO1"
   ONCLICK="goto_url1()"
   >
```

The author actually uses three sets of these, but only one is covered here. For security reasons, one can only MARK the URL of a file from the local hard drive.

The JavaScript code for `mark_url1` is as follows.

```
function mark_url(t) {
   t.value = top.MAINFRAME.document.location.href
   set_focus()
   }

function mark_url1() {
   mark_url(top.BOTFRAME.document.FORM1.TEXT1)
   }
```

Since the text box requires a lot of horizontal space, it is placed in the `BOTFRAME` and not the `LEFTFRAME`. When the MARK button is pressed, the caption of the text box `TEXT1` is set to the current URL. The function `mark_url1` is used to make it easy to add additional capability.

The JavaScript code for `goto_url1` is as follows.

```
function goto_url(fs) {
   if (fs == "") {
      set_focus()
      }
   else {
      goto_page(fs)
      }
   }

function goto_url1() {
   goto_url(top.BOTFRAME.document.FORM1.TEXT1.value)
 }
```

If the text box is empty, then the focus is set to the MAINFRAME, and no page is loaded. Note that the function goto_url is used to make it easy to add additional capability.

The author's word processor generates HTML. A word processor macro was written to create the $goto$.htm file that loads the just-generated HTML file into the browser and then switch to the browser. That is, clicking on the goto button calls the goto_link function that loads the $goto$.htm file into the MAINFRAME pane. The goto_link function is used to go to the page $goto$.htm on the local hard drive and appears as follows.

```
function goto_link() {
  top.MAINFRAME.location.replace('file:///D|/RMS1.WEB/$goto$.htm')
  set_focus()
      }
```

An example of an INPUT button tag is as follows.

```
<INPUT
   TYPE="button"
   VALUE="goto"
   NAME="goto"
   ONCLICK="goto_link()"
   >
```

An example of the file $goto$.htm is as follows.

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JAVASCRIPT">
top.MAINFRAME.location.replace('file:///D|/RMS1.WEB/index.ht
m#3h')
top.MAINFRAME.focus()
</SCRIPT>
</BODY>
</HTML>
```

Note that, in this case, the JavaScript code appears inline and is not included as a separate file. In addition, a target is added to the file by the word processor macro, in this case "#3h", so that the location in the web page just loaded is approximately the corresponding place in the document that was used to generate the HTML. This method makes it easy to generate HTML and then switch to the browser to see what the HTML looks like.

Due to security restrictions, however, it is not easy to go from an HTML page in the browser to the word processing document that was used to create that file. The best the author has done is to create a button that fills a text box with the URL of the page. This text box can then be copied to the clipboard, the word processor can be made active (e.g., via Alt-Tab), and a macro used to use the contents of the clipboard to determine and load the appropriate document. This is cumbersome, but when dealing with over a thousand document files, it is better than manually searching for the desired file.

The author's word processor is used to generate both an HTML file that can be used for classroom presentation (i.e., it is slide- based with larger-sized fonts and contains questions and answers to be used in class) or to an HTML file for student use (i.e., it uses smaller fonts and does not contain the answers to questions asked in class). For example, if `F:\IS207\java-08.spr` is the document file, then the HTML presentation file would be `D:\IS207.HTM\java-08.htm` and the student accessible HTML file would be `D:\IS207.WEB\java-08.htm`. The presentation files (with selected answers to questions) remain on the author's local hard drive while the student accessible files are automatically updated to the network and, therefore, the Internet.

The following is a more involved function called `talk_switch` that can be used to automatically switch between the presentation version and the student accessible version of the generated HTML files.

```
function talk_switch() {

   var d = top.MAINFRAME.document
   var fs = d.location.href

   var i = fs.indexOf('.WEB/')
   var j = fs.indexOf('.HTM/')
   var k = fs.length

   if (i != -1) {
      fs = fs.substring(0,i) + ".HTM/" + fs.substring(i+5,k)
      d.location = fs
      }

   if (j != -1) {
      fs = fs.substring(0,j) + ".WEB/" + fs.substring(j+5,k)
      d.location = fs
      }

   set_focus()

   mark_url2()
   }
```

WARNING: Again, JavaScript is case sensitive, so `indexof` is not the same as `indexOf` is not the same thing as `IndexOf`. Beware!

Briefly, the string variable `fs` contains the file. If the directory extension is `.WEB/`, it is changed to `.HTM/`. If the directory extension is `.HTM/`, it is changed to `.WEB/`. Then that other file is loaded and the focus is set.

The `INPUT` tag is as follows.

```
<INPUT
   TYPE="button"
   VALUE="Talk"
   NAME="Talk"
   ONCLICK="talk_switch()"
   >
```

CONCLUSIONS

This paper has covered just a few of the possible uses of JavaScript as a way to automate teacher-side web page navigation for classroom presentations. Future enhancements include investigating the use of Java for providing useful capabilities and creating a search mechanism for quickly locating topics in the author's local web system. The JavaScript code presented has worked well in practice, but there are surely better ways to write the code. These and other methods not covered here have improved and continue to improve the bringing of technology into the classroom. The only limitation is your imagination and the security restrictions imposed by the browser.

Accessing ITP: Accommodating the Disabled

R. Wayne Stewart, M. Ed., M. A.
Disability Services/504 Coordinator, ADA Compliance Officer and
English Instructor
Clayton College & State University
P. O. Box 285 Morrow, GA 30260
770.961-3719 – Voice and TDD
770.960-2166 – Fax
stewart@dd.clayton.edu
http://www.clayton.edu/disability_services

INTRODUCTION

On March 10, 1997, the Board of Regents of the University System of Georgia voted their unanimous approval of a proposal put forward by the presidents of Clayton College & State University (CCSU) and Floyd College (a two-year institution in Rome, GA) to make information technology a centerpiece of every aspect of the institutions' mission and operations. This initiative, **The Information Technology Project** or ITP, is the first such effort within the University System of Georgia and one of the first of its kind in the nation.

ITP focuses on information technology as a central part of teaching and learning - primarily through UPITA™ (Universal Personal Information Technology Access). UPITA equips each student (and faculty member) with a powerful multimedia notebook computer with remote communications capability, including "anywhere" dial-in access to the CCSU & Floyd campuses, GALILEO (the state's digital library), the Internet, the World Wide Web and e-mail.

Through UPITA, CCSU provides every student with a notebook computer on a quarterly basis. The notebook is used in class or across campus, at home, or anywhere the student takes it. Students pay a quarterly fee to cover a portion of the costs of the notebook, remote access and support services. Any student who prepays tuition and fees when preregistering for the next quarter can retain use of the notebook computer through the break between terms and continue to utilize the computer and its accompanying services (e.g., walk-up and call-in help). Since the University (instead of the students) is purchasing the notebook computers, the University received major educational discounts on computing software. This discount allows the student to receive the best software and hardware available at a moderate cost.

CCSU also is a "Going the Distance" school. This means that the University offers the associate degree in Integrative Studies entirely through telecourses and other distance learning methods in collaboration with the Public Broadcasting Service (PBS), Georgia Public Broadcasting (GPB) and local cable providers. These courses are enhanced by faculty-student interaction using the notebook computers to communicate in real time and by e-mail.

Universal Personal Information Technology Access (UPITA), is a program to provide each CCSU student with a notebook computer and Internet access. Costs of support, assistance and repair are shared between students and CCSU, and matches for funds generated from the state lottery and allocated to the institution by the Regents.

CCSU's Goals

-Provide students with state-of-the-art technology to enhance their current and future careers;
-Improve dramatically the quality of the education offered students;
-Expand and improve on- and off-campus services to all students;
-Improve institutional flexibility and capability to acquire, use, support and improve continuous information technology, and
-Reduce institutional reliance on state funding for information technology.

Each student will have personal use of a current, powerful multimedia computer, Internet access and user support services at a cost well below an individual's lease or purchase price for the machine alone.

The Faculty's Role

Information technology and notebook computers are central to teaching at Clayton College & State University. Each faculty member has been provided with Ws/her own notebook computer and training in the use of the notebook computer. All newly hired faculty are expected to bring to their positions experiences with instructional technology.

Using the Computers in the Classroom and in the Learning Center

Classrooms have been renovated to allow students to plug in the notebook computer and have access to the campus network. The recently renovated Learning Center contains more than 275 such connections.

Training students in the use of their notebooks and software

Every quarter, courses are provided during the day, at night and on weekends to train students in notebook use. These training sessions are available at the beginning of each quarter as well as after the quarter is in progress. They are convenient for students who work.

Other Institutions' and ITP

Among public colleges and universities, only the University of Minnesota at Crookston, a small baccalaureate institution, has much experience with such an institution-wide effort.

THE CHALLENGES

Prior to the implementation of ITP at CCSU, issues regarding access by our disabled student population began to emerge. According to Section 504 of *The Rehabilitation Act of 1973,* and the *Americans with Disabilities Act, 1990 (ADA)*, individuals with disabilities cannot be excluded from participation in programs covered by Title Two of the ADA, which includes public colleges and universities. As CCSU began to implement ITP, then, the university also had to address the problem of providing equal access to disabled students. Obviously, several issues and problems appeared quickly. Primarily, the problems were directly related to the various disabilities seen on college campuses. For example, the learning disabled comprise a fair portion of a disabled student population, and this group has specific needs with regards to academic/programmatic accommodations. Low vision students, blind students, students with mobility and neurological impairments, amputees, and other special types of disabilities or combinations such as deaf and blind also present specific and many times highly individualized needs. Furthermore, each disability type may require accommodations that vary widely from other students' needs. Of course the university could not wait until a particular need arose to address it since most likely a good portion of the quarter would lapse before the appropriate accommodation could be put in place. Likewise, obtaining special equipment, software, hardware, etc. on an individual basis would present not only an undue financial hardship considering that much such equipment is so individualized that use by other students is not reasonable. Consequently, the Disability Services Coordinator/ADA Compliance Officer sought solutions to these problems proactively with ITP personnel and Student Services administration. The problem was addressed by considering generalized and somewhat homogenous groups of disabilities as they were related with regards to the type of accommodation that would most effectively serve the group's majority. These groups split into two distinct sets.

The Challenge of Accommodating Low Vision/Blind Students

This group set presents a unique as well as difficult challenge. So much of the essence and heart of using computer technology rests with the output on the display that low vision or blindness in fact is at once a barrier to access. In a program like ITP students and faculty will be required to use their computers for a variety of tasks. Naturally, the university intranet is essential for certain key information used for registering, selecting a class, submitting and viewing grades, searching, and other functions. Consequently, denied access to the technology would fundamentally discriminate against the individual whose disability prohibits them from seeing the contents of a computer display, and indeed, even low vision presents a problem when the small screen of a notebook is used. Another group that needs to be considered with the blind/low vision group is the learning disabled, who often experience visual perception disorders that cause problems not unlike those faced by blind/low vision students – fortunately, to a limited degree, some similar modifications can also be used.

The Challenge of Accommodating the Physically Impaired

Those students and faculty members with physical impairments,, as well as low vision/blindness, that would otherwise prevent them from entering input on the computer's keyboard would also be subject to discrimination if reasonable and appropriate accommodations were not provided. While most individuals in this group can receive and use requested information as it appears on screen, getting data there poses the greatest challenge for individuals who have had amputations, strokes, spinal cord injuries or birth defects. The smaller size of notebook computers compound a problem that is present with desktops. Again, CCSU sought solutions.

The following conclusions were reached by CCSU:

The problems of access to ITP seemed to be problems with input and output of data for specific groups of individuals with disabilities.

Software and/or hardware that provides alternative methods for input and output was indicated.

A stock of software/hardware was needed that would allow individualized customization of notebook computers immediately upon the recommendation of the Disability Services Coordinator.

THE OPPORTUNITIES

Faced with these challenges, CCSU's Disability Services Coordinator (DSC) was compelled to seek acceptable, reasonable, appropriate and cost effective strategies for accommodating the disabled. The initial examination of the challenges having revealed data input and output as the two primary modalities requiring modification, the DSC began researching various products to ensure total and equal access to ITP.

Input Modifications

The most challenging task was deciding which of the many data input products would accommodate the widest range of disabilities in a timely, on-demand and cost effective manner. After reviewing and using several different products, the DSC chose IBM's Simply Speaking Gold. This choice has been questioned a few times, but when compared to other voice recognition programs in consideration of the number of students who were projected to request this modification, the IBM product met all the criteria more completely.. Six copies were purchased from a local Comp USA, and all copies are retained in the DSC's office. After the DSC has determined that voice recognition is a reasonable and appropriate accommodation and makes such a recommendation, one of the software packages will be installed on the student's notebook computer. At approximately 100 dollars each, with the ability to assist a wide range of disabled student needs, the IBM product, for the present, is serving students well.

Output Modifications

Less challenging, but nevertheless equally as significant, was purchasing a product that would modify output, which consist mostly of text on the display. Some of the programs considered were Arkenstone's OPENBOOK, Henter-Joyce's JAWS and AI Squared's ZOOMTEXT EXTRA. Since the Zoomtext program uses the notebook's pre-installed sound card rather than a separate device like DECTALK, the AI Squared product seem to most readily meet the criteria. Cost, ease of installation and use as well as navigation and screenreading capabilities placed this product in the lead soon, and when the screen enlargement aspect was considered, Zoomtext was obviously the product with the most versatility and user friendly functionality in addition to meeting our criteria. Thus far, students requiring modified data output have been adequately served.

Supplemental accommodations

In addition to readily available software for installation on disabled students' notebooks, existing desktops in the Disability Services Lab also provide and even enhance students' equal access. The Lab has a Pentium with 21" monitor, scanner and printer that is used with the new version of Arkenstone's OPENBOOK, which is an optical character recognition system that allows students to hear text as well as view or print the text after scanning the page(s). In addition, AI Squared's VISIBILITY program provides enlargement up to 30x for scanned text, which can also be read via the voice synthesizer or printed in the enlarged format.

CONCLUSION

Students with practically any type of disability that limits their ability to access data so vital to ITP and to their learning at CCSU can now are accommodated. With students now being able, and in some instances required, to navigate the campus intranet to register, check on class status and grades, financial aid status, class standing and so on *ad infinitum*, total access is a must. So much of the coursework at CCSU is either already provided on-line or is scheduled to be offered electronically soon, that the campus intranet is often much more comfortable and friendly than the internet, especially for computer user newbies. Furthermore, successful navigation of the campus intranet is often students' springboard to cyberspace. Clearly, denying students with disabilities access to computer technology by not modifying input and output modalities indeed denies them participation in any program at CCSU given the institution's commitment and implementation of such an extreme and innovative technological directive. Of course whether more learning actually happens will have to be determined during the post-evaluative phase of institutional measurement; however, students with disabilities achieve more learning immediately. Disabled students not only *have* access to information that previously remained obscure, they now are *learning* various ways *to access* that information.

The OTHER Dimension of Computer Competence for Teachers

M. J. Stockton, Ed.D.
Frances S. Evans Professor of Education
and Director of Teacher Certification
Department of Education
William Jewell College
Liberty, MO 64068
(816) 781 7700 x5499
stocktonm@willliam.jewell.edu
mjstock@sky.net

Introduction

This study chronicles the long term development of a conceptual framework for the training of prospective teachers in the area of computer competence. Today that conceptual framework drives the curriculum of EDU 215: Information Technology, a course required of students in all teacher certification programs at William Jewell College. Although computing activities appropriate for teachers are integrated into several courses in the teacher education programs, EDU 215: Information Technology is the course in which prospective teachers are expected to learn the basic computer skills needed by practitioners in their profession.

The search for a conceptual framework for the training of teachers in computer competence at William Jewell College can be dated to the work of a faculty committee whose initial mission had been the analysis of the College's general education curriculum, *i.e.*, the Quality Undergraduate Education Committee, locally called the QUE Committee. "Quality Undergraduate Education" was a nationwide initiative of the Council for the Advancement of Small Colleges (CASC), later renamed Council of Independent Colleges (CIC).

Defining Computer Literacy

The QUE Committee had begun its work in the middle 1970's, prior to the advent of microcomputers. As the committee continued its work into the late 1970's, its attention turned to the propriety of computing skills in a liberal arts general education curriculum. It was quickly concluded that a good liberal arts education for the 1980's and 1990's would have to include some kind of "computer literacy." Exactly what skills would constitute computer literacy remained obscure. For instance, must one be able to master one or more computer programming languages to be "computer literate?" Does the ability to use a word processor authenticate one's "computer literacy?" Is there a core of computing skills that would be appropriate for all students?

The QUE Committee eventually concluded that although "computer literacy" must ultimately be defined within the context of one's major field of study, there are some basic types of applications that should receive serious consideration regardless of one's college major. That is, what constitutes computer literacy for an English major may involve a vastly different profile of skills than that necessary for a biology major, a business major, a mathematics major, *etc.* Yet, there would be some basic skills essential to all. It would be incumbent upon the respective major departments to develop specific computing skills profiles for their own majors. Subsequent conversations resulted in the campus-wide adoption of the following definition of "global computer literacy" to serve as a core around which the respective departmental, discipline-specific profile of essential computing skills would be determined.

Global computer literacy refers to the acquisition of those computing skills that can reasonably be assumed to facilitate the pursuit of one's academic tasks within her/his own field of study. Minimally, it consists of an awareness of four types of computer applications (*i.e.*, word processing applications, data base managers, draw/paint software, and electronic spreadsheets) and a working knowledge of the applications determined by the respective departmental faculties to be most relevant to the discipline.

Word Processing	Draw and Paint Applications
Database Management	Spreadsheet Applications

Figure 1. An Initial Conceptual Framework for Computer Literacy

The conceptual framework is remarkably consistent with the "Fundamental Skills and Concepts for Today's Teachers" set forth by the International Society for Technology in Education (ITSE, 1992), which identifies the same four basic applications in which students must "...demonstrate skill in using productivity tools for personal and professional use..."

Funding Issues

The acquisition of a substantial number of microcomputers for student and faculty use was a task that was beyond the capacity of the operating budget and raiding the endowment to support a program that some still considered "trendy" was out of the question. Funding would have to be sought elsewhere.

Fortunately, some philanthropic foundations had also recognized the wisdom of encouraging computer literacy on college campuses. The Speas Foundation awarded a grant that funded the purchase of twenty Apple][computers for the establishment of the first microcomputer lab on campus. The Francis S. Parker Foundation funded proposals for purchasing computers for faculty offices. Restrictions on the grants were minimal. Basically, if the proposal asserted that the faculty member intended to use the computer in any way related to her/his job as an instructor, it was granted.

Faculty Development

Having dealt with the philosophical and financial considerations by way of the usual academic agendum, the faculty soon acquiesced to its own essential role in the campaign to integrate computer literacy into the curriculum. How could a faculty whose collective concept of educational technology equated to using electric typewriters instead of manual ones be expected to model computer literacy for their students? For a computer literacy program to have any sense of credibility it would have to be based on a basic principle: "...if it is essential for the students to be computer literate, then the faculty must be computer literate as well."

To be candid it was recognized that not all faculty members were computer illiterate. Computer competence on campus tended to reside in the faculties of mathematics, business, and the natural sciences, since the computer had been a long-standing tool of research in those disciplines. However, even among those instructors the actual personal experiences in computing had been limited to the use of "main frame" computers at the large universities where they had acquired their advanced degrees or in business or industrial settings where some had worked before beginning their careers in college teaching.

Reactions among faculty members to their anticipated roles in the computer literacy program were predictably mixed. While some quickly took advantage of the opportunity to write proposals for grants to purchase computers for their offices, others attended off campus workshops on basic computing skills and used personal financial resources to buy their own hardware, software, and books to increase their own skills. Still others checked their calendars and calculated the time that would have to elapse before they were eligible for early retirement! One English professor was quoted as having said, "I shall never commit my deathless prose to any such banal electronic gadget." One would hope that it was said tongue-in-cheek; but, nevertheless, the comment accurately reflected an existing point of view among some faculty members.

The QUE Committee, whose name had been changed by this time to Computer Facilitating Committee to more accurately reflect its work, took on the challenge of faculty development. Assuming that word processing would be the type of computer application that would hold the greatest appeal to the largest group of faculty members, the committee arranged for a workshop on the use of the AppleWriter][.

The subcommittee charged with the actual execution of the workshop consisted of a biology professor, a business professor, and an education professor. They decided that the education professor would lead the presentation and demonstration of AppleWriter][. [To this day, I contend that the logic behind this decision was to demonstrate that if an education professor (yours truly) could manage to use the computer, surely the faculty from all other disciplines could master it!]

Enter "GUI"

The place of computer programming in the discipline-specific profiles of essential computing skills was still a point of contention. This question was largely rendered moot with the arrival of the Macintosh with its graphical user interface (GUI) and accompanying mouse. Since pointing, clicking, and dragging could accomplish so many of the practical computing tasks, writing code in

BASIC, AppleSoft BASIC, or any of the more sophisticated programming languages seemed less important as an "essential skill" in most disciplines.

In 1993, funded by a grant from Southwestern Bell, another computer lab was opened. It consisted of twenty networked student computing stations, a server, two laser printers, and an instructor's station connected to an LCD panel and overhead projector. The central processing units were Macintosh Centris 610 models. Although IBM-compatible computers had been considered as possible operating platforms, Windows 95 had not yet arrived allowing IBM-compatible computers the "user-friendliness" afforded by the Macintosh.

"Computer applications" courses, in which students were introduced to the software appropriate to the campus definition of computer literacy, and graphic arts courses were scheduled for regular class meetings in the new lab. Other classes could schedule meetings in the room for special occasions-- usually to gain access to the overhead projection of computer images-- on a first come first served basis at times when the room was not in use by its regular clientele.

Migration from LAN's to the Internet

Several local area networks had appeared on the campus as campus-wide microcomputing grew. The purpose of the LAN's was typically to facilitate the sharing of laser printers and, in some cases, databases used by several members of a departmental workgroup.

Throughout the 1994-95 academic year the basic cabling, hardware, and software were installed for connecting to the internet. All working microcomputers owned by the college were "networked" and a drop was installed in almost every room of every building on the campus. Workshops for students and faculty were conducted to introduce the new users of the internet to email management.

The issues related to the process of getting the campus "network-ready" are numerous enough to deserve their own discussion but are beyond the scope of this paper. The purpose of this section of this paper is to establish the fact that the internet had by this time become a real part of campus life and would be expected to have an impact on the local definition of computer literacy.

EDU 215: Information Technology

EDU 215: Information Technology is the current name of the course at William Jewell College that serves prospective teachers. It, like similar courses on other campuses, has evolved from the traditional "audiovisual education" course for teachers. Among the goals and objectives of the original audiovisual education course were operating a filmstrip projector, threading a 16mm projector, designing a bulletin board, laminating instructional materials, VCR operation, using an overhead projector, *etc.* Computing and internet technologies had no place in its curriculum. Computing was something done by those scientific geniuses in horn-rimmed glasses with the plastic protectors in their shirt pockets and, of course, nobody had even heard of the internet.

When microcomputers began appearing in classrooms and talk of their representing only a fad that will soon go away died down, it became clear that teacher training in the use of computers was imperative. It seemed reasonable that the course in the teacher education curriculum most suited to take on the responsibility was the audiovisual education course.

In light the campus definition of computer literacy (see Figure 1), the new course syllabus included goals related to word processing, database applications, draw and paint software, and electronic spreadsheet applications. Students would be expected to gain an awareness as well as a working knowledge of the four basic types of applications. Due to the pervasiveness of computing in the course curriculum, it was given permanent resident status in the computer lab along side of the computer applications and graphic arts courses.

Except for the necessary elimination of some of the course's traditional goals in order to make room for the computer literacy goals, the plan seemed satisfactory to all interested personnel. The name of the course was changed from "Instructional Media" to "Instructional Technology." It was taught without any major modifications for several years.

The Other Dimension of Computer Literacy

In 1994, when Vice President Al Gore made his famous speech about "a different kind of super highway" many listeners had no idea what he was talking about. According to a government survey in 1996, fifty per cent of U.S. schools have access to the internet (Grabe and Grabe, 1998). The Telecommunications Act of 1996, which contains attempts to remove some current barriers to getting schools connected to the internet, may facilitate the goal of having all classrooms connected by the year 2000.

The internet represents the new challenge to teacher preparation. Skills in the use of internet technologies have to be integrated into teacher training and into our conceptual framework for computer literacy. Internet technologies constitute a new dimension that demands a place alongside of the four basic application types of our model identified earlier. (Figure 2)

Word Processing

Draw and Paint Applications

Internet Technologies

Database Management

Spreadsheet Applications

Figure 2. An Emerging Conceptual Framework for Computer Literacy

Implications of the "Other" Dimension for EDU 215: Information Technology

Of course the short answer to the question suggested by the title of this section is that the curriculum had to be changed again. But which specific internet technologies should be included posed a more complicated question.

Using the 'net to teach the 'net is a compelling idea. Numerous examples of the internet's use as a delivery system for instructional information in several disciplines (Thomson *et al,* 1997) have been cited. Examples range from simply posting the syllabus or a single assignment on a web server to complete, stand alone on-line courses.

The instructor set out to design some activities that would at least address the "awareness" level of the computer literacy model regarding internet technologies appropriate for teachers. Conversations with other technology instructors along with information (and inspiration) gained through participation in workshops and conferences like ASCUE proved to be the most valuable bases for curricular decision making in this area.

Modeling expected behavior. It seemed logical that internet assignments would have more credibility for prospective teachers as legitimate instructional skills that they could use in their own teaching if the instructor modeled them in his teaching. Some attempts to demonstrate the use of the internet in the teaching of the course follow.

1. The course syllabus was put in *html* (hypertext markup language) form and posted on the college's web server. Other education professors were encouraged to do the same.

http://www.jewell.edu/~education/syllabuses/media.html

2. Students were encouraged to visit the instructor's personal web pages.

http://www.jewell.edu/~education/stocktonm
http://www.sky.net/~mjstock/
http://members.aol.com/mjstockton/

3. A self-paced tutorial for one of the class assignments was placed on the served along with some graphics that they would use in making their own web pages.

http://www.jewell.edu/~education/stocktonm/webmaking.html
http://www.jewell.edu/~education/stocktonm/icons.html

4. Students' pictures were taken with a digital camera and displayed on the class's home page with links to some resources for the class projects. The captions under their pictures served as hyperlinks to their own web pages which they would be creating in the class.

http://www.jewell.edu/~education/EDU215/

Student projects. The student projects designed to introduce them to the internet included the following:

1. Email projects. Students were given the instructor's email addresses on the first day of class and asked to use them for routine communication during the course. At least one email with an attachment was required.
2. Web page project. Each student was required to design a personal web page to be posted on the college's web server and used in their formal presentation of the project in class.
3. Internet search projects. Students were asked to use various search engines to find information that would be needed in completing other class projects.
4. Downloading project. In connection with other projects, especially the web page project, students were required to download shareware or freeware that would be needed (*e.g.*, html editors).

Reflections and Future Plans. This story is not without its down side. As new material is introduced into any course it crowds other material out, unless the credit hours' value of the course is increased. One wonders how prospective teachers will acquire the technology skills needed for their work that have been eliminated from the course in deference to computing and internet skills.

Grading the activities described here does not seem to fit the traditional academic mode. Rubrics that have been created for some of the projects need to be expanded to define the standards of an "A" project, a "B" project, *etc.*

Electronic plagiarism has been a problem. Often students underestimated the time that their web page project would require. Occasionally, as panic set in near the project deadline a computer savvy friend was called on to design the web page. Perhaps progress reports required at various intervals can help to avoid this problem.

Among the positive outcomes of the emphasis on internet skills has been the classic phenomenon that motivates all teachers: the look of wonder in the face of a student who has mastered a skill that s/he thought impossible just a short time before.

Even with the modest beginning described in this paper, the plan to integrate internet technologies into this course and others is destined to grow. Some additions to the course anticipated in the near future include the following:
1. Using *webforms* for tests and practice activities;
2. Establishing a *listserv* for routine course communication;
3. Video conferencing, perhaps with another information technology class;
4. Uploading articles to newsgroups; and
5. Adding on-line tutorials for other class projects.

Other ideas will almost certainly develop as the result of the instructor's participation in the 1998 ASCUE Summer Conference.

Print References

ISTE. *Curriculum Guidelines for Accreditation of Computing and Technology Programs.* Eugene, Oregon: The International Society for Technology in Education, 1992.

Grabe, Mark and Grabe, Cindy. *Learning with Internet Tools.* New York: Houghton Mifflin, 1998

Thomson, Nancy, Echeverria, DonnaRose, and Mills, Bob. MIS On-Line. *Proceedings of the 1997 ASCUE Conference.* Myrtle Beach, South Carolina: ASCUE, 1997.

Internet References (in order as cited)

Syllabus for EDU 215: Information Technology. http://www.jewell.edu/~education/syllabuses/media.html

Instructor's Personal Web Pages http://www.jewell.edu/~education/stocktonm http://www.sky.net/~mjstock/ http://members.aol.com/mjstockton/

Tutorial on Web Page Design http://www.jewell.edu/~education/stocktonm/webmaking.html

An Instrument for Projection of Resource Requirements in a Client/Server Environment: A Research Report

Winston Tellis Ph D
Fairfield University
School of Business
Fairfield, CT 06430
WINSTON@FAIR1.FAIRFIELD.EDU

Goal

The goal of this study is to determine the information technology needs at Fairfield University, in a client/server environment.

Objectives

The primary objectives of this study are:

1. To develop an instrument to assess the information technology needs of the user community. Particular emphasis will be placed on client/server computing and the Internet as a result of widespread access to the World Wide Web (WWW).

2. To establish a basis for understanding current and future economic issues of information technology acquisition.

Research Questions Generated by the Objectives

1. What patterns of acquisition emerge from the current computing environment and the perceived needs for computing?

2. What characteristics of the categories of computing use contribute to the patterns of acquisition?

   The Urban Information Systems Project (URBIS) conducted by the University of California, Irvine, provided the logical categories, adapted by King and Kraemer (1985) and used by Levy(1988): (a) Technological development (b)Structural arrangements (c) Socio-technical interface (d) Political economic environment, and (e) Benefits / problems.

3. How will the institution balance the need for technological changes with the need to continue the accomplishment of routine tasks?

Methodology

The methodology used in this study will follow the recommendation of Yin (1994) and has four stages: 1) Design the case study, 2) Conduct the case study, 3) Analyze the case study evidence, and 4) Develop the conclusions, recommendations and implications.

Case study research is not sampling research; that is a fact asserted by all the major researchers in the field, including Yin, Stake, Feagin and others. The unit of analysis is a critical factor in the case study. Case studies are multi-perspectival analyses. This means that the researcher considers not just the voice and perspective of the actors, but also of the relevant groups of actors and the interaction between them.

Case study is known as a triangulated research strategy. Snow and Anderson (Feagin et al, 1990) asserted that triangulation can occur with data, investigators, theories, and even methodologies. Stake (1995) stated that the protocols that are used to ensure accuracy and alternative explanations are called triangulation. The need for triangulation arises from the ethical need to confirm the validity of the processes. In case studies, this could be done by using multiple sources of data (Yin, 1984). The problem in case studies is to establish meaning rather than location.

For this case study, the researcher replicated Levy's (1988) study, but also adds to the field by examining aspects of client/server computing, the Internet, and the WWW. It is based on a modification of the methodology devised by Yin (1984).

1. Design the case study protocol:
   determine the required skills
   develop and review the protocol
2. Conduct the case study:
   prepare for data collection
   distribute questionnaire
   conduct interviews
3. Analyze case study evidence:
   analytic strategy
4. Develop conclusions, recommendations, and implications based on the evidence

Results

The results of the survey were tabulated using SPSSx version 7 running on a Pentium PC 75 megahertz under Windows 95. The results are excerpted from the original study to conform the requirements of this publication.

Table 1
Survey Response Characteristics

Survey Type	# Distributed	# Respondents	% Response
Faculty	191	88	46
Administrators	22	14	64

It is clear from the data above that the response rate was sufficient to conduct the planned statistical tests. The response was representative of the faculty and the administrators and was considered adequate for this study.

Table 2
Projected Faculty Computing Use
N=88

Item	Question	%Increase	%Decrease	%Same
3	Number of Applications	93	0	7
4	Amount of time spent	86	1	13
6	Data communications	87	1	12

Table 2 indicates the potential for increased resource requirements in the near future. The projection is for significantly increased use of information technology.

Table 3 below shows that the respondents expect their need for specific information technology items to increase over the next five years. Items relating to database access and the Internet are particularly important to the users.

Table 3
Important in Next 5 Years (Faculty)
N=88

		All		* Cross Tabulations *							
				A&S		Business		Nursing		GSEAP	
Item	Question	A	D	A	D	A	D	A	D	A	D
39	Dept support for net PC	51	28	44	34	64	7	40	20	100	0
63	More LANs	70	4	70	4	71	7	75	0	80	0
64	Search library holdings	95	5	92	0	100	0	100	0	100	0
65	Database Search	98	0	96	0	100	0	100	0	100	0
66	Off campus computing	82	2	79	4	100	0	80	0	100	0
67	Email	85	1	88	2	57	0	100	0	100	0
68	Students PC	78	4	75	4	79	7	100	0	67	0
69	Off campus email	82	2	83	2	64	7	14	50	100	0
70	Laser printing	95	2	92	4	100	0	100	0	100	0
71	Test scanning	45	13	39	14	50	14	100	0	83	0
72	Upgraded PC	93	5	90	6	93	7	100	0	100	0
73	Video conference	57	8	54	12	50	7	100	0	67	0
74	OCR	75	4	67	6	93	0	100	0	100	0
75	Voice recognition	33	12	34	16	21	7	50	0	50	0
76	Database browsing	83	1	83	0	19	0	75	0	100	0
77	Video capture	59	7	54	8	50	14	75	0	83	0
78	Access to WWW	95	5	94	0	93	0	100	0	100	0
79	Class access networked CD	77	24	71	0	86	0	100	0	83	0
80	Class material on WWW	74	2	71	4	62	0	100	0	67	0

(A = % Agree; D = % Disagree; Neutral = A - D; A&S = Arts & Sciences; %Business = School of Business; %Nursing = School of Nursing; GSEAP = Graduate School Of Education & Allied Professions)

Context of Computing Use

The King and Kraemer (1985) categories were adapted by Levy (1988) for his study. The survey items in the questionnaires used by Levy (1988) and in this study also fell into those categories as follows:

In the Faculty Survey the items that fell into each category were:

Technological Development, items 39,63-80,82-102,107-116
Structural Arrangements, items 16-17,38
Socio-Technical Interface, items 18,51-62,117,120
Political/Economic Environment, items 19,40,42-50,104-105,118-119
Benefits/Problems, items 25-37,106

Technological Developments

Table 4
High Priority Should be Placed on (Faculty)
N=88

		*		Cross Tabulations				*			
		All	*	A&S		Business		Nursing		GSEAP	
Item	Description	A	D	A	D	A	D	A	D	A	D
114	Access to WWW	80	7	76	10	93	0	80	0	67	17
115	Access to Instructional labs	71	5	67	6	77	8	80	0	50	0
116	Access to Student labs	70	6	69	6	64	14	100	0	33	0

(A = % Agree; D = % Disagree; Neutral = A - D; A&S = Arts & Sciences; %Business = School of Business; %Nursing = School of Nursing; GSEAP = Graduate School Of Education & Allied Professions)

Table 4 shows that instructional uses are expected increase and that the users are expecting additional resources to be available.

Structural Arrangements

Table 5
University policies (Faculty)
N=88

			*	Cross Tabulations						*	
		All	*	A&S		Business		Nursing		GSEAP	
Item	Description	A	D	A	D	A	D	A	D	A	D
16	Univ has effective guidelines	12	63	11	68	7	43	20	80	33	33
17	Univ allocates resources equitably	19	55	19	61	0	36	0	80	50	17
38	Satisfied with computing decisions	7	68	11	66	0	71	0	67	0	50

In Table 5 the respondents indicate their dissatisfaction with the institution's computing policies and their ability to influence decisions regarding their computing needs.

Political/Economic Environment

Table 6 below shows the rejection of all the choices for funding information technology acquisition. The users project increased usage and the need for additional resources, but cannot accept a reduction in any area that might provide the funds for acquisition of resources.

Table 6
Sources for funding (Faculty)
N = 88
*** Cross Tabulations ***

		All	*	A&S		Business		Nursing		GSEAP	
Item	**Description**	**A**	**D**	**A**	**D**	**A**	**D**	**A**	**D**	**A**	**D**
19	All student access computers	97	1	96	2	100	0	100	0	83	0
40	Frequently approached by vend	14	73	15	77	7	79	0	50	17	50
42	From Faculty positions	3	90	0	92	0	85	0	100	0	100
43	From Support positions	29	56	32	55	23	46	67	33	25	50
44	From other equipment	36	44	29	44	54	31	75	25	50	50
45	From Professional Travel	11	72	6	71	23	62	0	100	0	100
46	From Plant Maintenance	26	53	23	51	31	54	25	50	75	25
47	From New Programs	43	42	47	35	39	54	33	67	50	50
48	From Salary Increase	5	76	2	73	15	69	0	100	0	100
49	From Current Instruction Programs	27	60	28	55	23	69	33	33	50	50
50	Current Support Programs	23	65	37	49	15	46	60	20	60	20

Future Research

A factor analysis was run on the Fairfield University data, on each of the five King and Kraemer (1985) groupings of variables that were adapted for use in a study of the University of Arizona by Levy (1988). New factors emerged in each of the categories. The new factors that were selected had factor loadings of .6 and higher. Further analysis could be carried out using the new variables as part of a cross tabulation, or some other statistical test.

Conclusions

Some of the conclusions from the data analysis, interviews, and literature are:

1. Institutional planning for information technology is inadequate.
2. A shorter planning cycle is needed for information technology.
3. Allocation of resources is not equitable among users.
4. Users are dissatisfied with their ability to influence computing decisions.
5. Faculty and administrators did not accept any potential sources of funding for Info. Tech.
6. Faculty and administrators felt that computing enhanced the scope of their work.
7. The expenditures and procedures for implementation of client/server computing were not carried out in a systematic and documented manner.
8. The equipment acquisition procedures are not responsive to user needs either in terms of pricing or timeliness.
9. There is a low level of user confidence in network integrity.
10. The faculty expect to use networked PC's in the classrooms.
11. User productivity is lowered due to resource allocation problems, and other technology issues.

12. There will be a significant increase in the use of the Internet and WWW by faculty over the next five years, which will require a well-designed client/server environment.
13. The shift to client/server computing will result in higher financial burdens.
14. There is no formal procedure to configure the servers using capacity planning procedures.
15. Multimedia classrooms for instruction and support will be needed in the near future.

Implications

1. In a client/server computing environment formal capacity planning procedures need to be instituted, to ensure properly configured servers and adequately equipped client systems.
2. As the pace of technology advance accelerates, desktop systems are likely to become more capable than the server. This could present problems in the delivery of service and result in bottlenecks. The client/server environment must be continually monitored for efficiency.
3. A budget item must be included for information technology so that the expenditure for acquisition is part of the institutional planning process.
4. The information technology planning cycle should be shortened so that the institution is in a position to respond to the rapid pace of technology change.

References:

Blumenstyk, G. (1994, November). Colleges struggle to develop formal strategies to pay for computing. Chronicle of Higher Education, 41(10), A44-45.

Deitch, K. (1994, May 1). A price war for higher education. Change, 26(3), 53-54.

DeLoughry, T. (1992). 36 percent of colleges and universities plan to increase spending on computing. Chronicle of Higher Education, 39(11), A17.

DePalma, A. (1991, November 27). As a deficit looms, 26 threaten to quit key Columbia posts. New York Times, 141(48797), A1.

Evangelauf, J. (1992, March 11). At Public Colleges more double digit tuition growth. Chronicle of Higher Education, 38(27), 29,32.

Feagin, J., Orum, A., Sjoberg, G. (Eds.), (1991). A case for case study. Chapel Hill:University of North Carolina Press.

King, J. & Kraemer, K. (1985). The dynamics of computing. New York: Columbia University Press.

Levy, S. (1988). Information Technologies in Universities: An Institutional Case Study. Northern Arizona University.

Long-Range Planning Committee. (1992). The Long Range Plan 1993-1998. Fairfield, CT:Fairfield University.

Management Information Office. (1995). Undergraduate Admissions Trend 1970-1995. Fairfield University.

Stake, R. (1995). The Art of Case Research. CA:Sage Publications.

Yin, R. (1984). Case Study Research: Design and methods. Beverly Hills, CA:Sage Publishing.

Yin, R. (1994). Case Study Research: Design and methods. Beverly Hills, CA:Sage Publishing.

Appendix I
Survey Instrument

Faculty Assessments of Computing

Thank you for participating in this survey. Additional comments would be greatly appreciated, and space is provided at the end of the survey. To return, simply fold the survey the opposite way, and return through campus mail.

1. Estimate your time spent in the following areas (total 100%)
   ____% Instruction ____%Public Service
   ____% Research ____%Administration
   ____% Academic Support ____%Univ Service

2. Do you use computing or have the knowledge of computing activities at the university? ___ yes ____ no

If you answered no, please respond to items 17, 25-31, and 41-49 and return the survey.

3. In the next year, the number of computing uses/applications you use will: ____ increase ____ decrease ____ remain the same

4. In the next year, the amount of time you spend using computing will: ____ increase ____ decrease ____ remain the same

5. Which of the following best describes you as a computer user? (If more than one is appropriate, please rank)

____ I use the computer for Word-processing most of the time

____ Use Package software or software provided by others to access data or use applications through a menu-driven format or another set of procedures.

____ Understand the use of database and able to specify, access, and manipulate information or instructional Applications.

____ Utilize computer languages directly for your own Information needs. Develop your own applications, some of which are used by others.

____ Support other computer users within their areas. Though not a professional programmer or data processing professional., you are called upon by others for assistance.

____ While not a direct computer user, you benefit from computer applications in your work through conceptualization of work to be performed, or the direction of co-workers and subordinates.

____ Employed, at least in part, for computer expertise. Formally support end user activities, and perhaps involved in information systems management, computer instruction/training, and programming.

6. In the next year, your data communications needs will: ____ increase ____ decrease ____ remain the same

Please check any of the following that describe your computing uses or needs:

	Currently use	Could use now	Would enhance future work
7. Internet resources (Gopher, FTP etc)	____	____	____
8. World Wide Web (WWW) resources Netscape etc	____	____	____

9. Networked PC access from classroom ____ ____ ____
10. Artificial intelligence/expert systems ____ ____ ____
11. Enhanced or complex graphics abilities ____ ____ ____

12. Would you use a computer during instruction if it were available on campus with consulting support?
(Please check) ____ yes ____ no ____ need information

13. Please check any of the following computing design and acquisition activities in which you have been involved:

1. ____ Review of designs for new applications
2. ____ Providing test data for an application
3. ____ Approve or sign off on an application
4. ____ Working as a member of a technical group in designing an application
5. ____ Sitting on a policy board/committee overseeing computing use/resources
6. ____ Participating in assigning priority of data processing projects
7. ____ Participating in decision making about types/brands of hardware and/or software to be acquired
8. ____ Participating in determining allocation of resources for the acquisition of computing

14. If you currently use the VAX 6430, what would you prefer to use if the system were no longer available? (Please check)

1. ____ Not a user of the VAX 6430
2. ____ Networked Multimedia Microcomputer
3. ____ IBM system (other than PC)
4. ____ Technical Workstation (Sun/Other)
5. ____ Alpha
6. ____ Remote computer

15. Please describe the way you do your computing work by the number of hours each week in each category:

System Type	StandalonePC /Mac	Mainframe	Networked PC/Mac
Personal			
Departmental			
College/School			
University			
Off Campus			

Please circle the appropriate option from the scale of each item.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
16. University policy has provided effective guidelines for computing use in the university.	SA	A	N	D	SD
17. The University's central administration has been equitable in allocating available resources for computing.	SA	A	N	D	SD
18. Hands -on workshops designed specifically for faculty and research uses of information technology tools would be useful for me.	SA	A	N	D	SD
19. All students should have access to computing, regardless of the courses which they are enrolled in.	SA	A	N	D	SD

The following have strongly influenced my views about the use of information technologies in universities:

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
20. Personal Experience	SA	A	N	D	SD
21. Professional journals and conferences	SA	A	N	D	SD
22. Opinions of peers	SA	A	N	D	SD
23. News media and popular literature	SA	A	N	D	SD
24. Advice from vendors/consultants	SA	A	N	D	SD

25. The scope of the work I am able to undertake is directly increased by the use of computing.	SA	A	N	D	SD

The current computing resources of the university are an asset in:

26. Attracting undergraduate students	SA	A	N	D	SD
27. Attracting graduate students	SA	A	N	D	SD
28. Attracting faculty	SA	A	N	D	SD
29. Attracting sponsored research	SA	A	N	D	SD
30. Attracting alumni support	SA	A	N	D	SD

31.	Attracting corporate donations/grants	SA	A	N	D	SD
32.	Forming joint ventures with private sector	SA	A	N	D	SD

As a user of university mainframe computing resources, I am:

If you are not a user please check the box on the right and proceed to question 39: ____ not a user

33.	Able to effectively discuss needs with support staff	SA	A	N	D	SD
34.	Satisfied with available applications	SA	A	N	D	SD
35.	Satisfied with system response time	SA	A	N	D	SD
36.	Satisfied with the access to data for which I have clearance	SA	A	N	D	SD
37.	Satisfied with institutional data sets available for analysis	SA	A	N	D	SD
38.	Satisfied with our level of computing decisions	SA	A	N	D	SD

39.	There is considerable support for the acquisition of PC networks within my department/unit	SA	A	N	D	SD

40.	I am frequently approached by computer vendors and/or outside consultants	SA	A	N	D	SD

41.	In my area, the computing resources of the University compare favorably with computing resources in our peer universities	SA	A	N	D	SD

Resources for the acquisition and maintenance of computing would come from the reallocation of funds from:

42.	Faculty positions	SA	A	N	D	SD
43.	Support positions	SA	A	N	D	SD
44.	Other equipment and supplies	SA	A	N	D	SD
45.	Professional travel/Conferences	SA	A	N	D	SD
46.	Plant and equipment maintenance	SA	A	N	D	SD
47.	New programs	SA	A	N	D	SD
48.	Promotions and salary increases	SA	A	N	D	SD
49.	Current instructional programs	SA	A	N	D	SD

The following contribute to the effectiveness of my current computing work:

50.	Current support programs	SA	A	N	D	SD
51.	Frequently upgraded personal computer	SA	A	N	D	SD
52.	Sufficient data communications capabilities	SA	A	N	D	SD
53.	Appropriate computing resources	SA	A	N	D	SD
54.	Appropriate software	SA	A	N	D	SD
55.	Good documentation	SA	A	N	D	SD
56.	Sufficient training	SA	A	N	D	SD
57.	Sufficient consulting	SA	A	N	D	SD
58.	Sufficient support staffing	SA	A	N	D	SD
59.	Effective support staffing	SA	A	N	D	SD
60.	Access to the Internet, WWW, E-Mail, from the Office	SA	A	N	D	SD
61.	Access to the Internet, WWW, E-Mail from the classroom	SA	A	N	D	SD
62.	Access to the Internet, WWW, E-Mail from Home	SA	A	N	D	SD

The following computing developments are or will be important to the Fairfield University within the next five years:

63.	More Local area networks	SA	A	N	D	SD
64.	On-line search of library holdings from the office	SA	A	N	D	SD
65.	On-line search of national databases from the office	SA	A	N	D	SD

		Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
66.	Access to off-campus computers	SA	A	N	D	SD
67.	Access to on-campus electronic mail	SA	A	N	D	SD
68.	Require all students to have network ready personal computer	SA	A	N	D	SD
69.	Access to off-campus electronic mail or bulletin boards	SA	A	N	D	SD
70.	Access to convenient Laser printing	SA	A	N	D	SD
71.	Convenient access to scanned test scoring	SA	A	N	D	SD
72.	Frequently upgraded personal computers	SA	A	N	D	SD
73.	Video conference capability	SA	A	N	D	SD
74.	Optical scanning/character recognition devices	SA	A	N	D	SD
75.	Voice recognition and compound documents	SA	A	N	D	SD
76.	Software assistance for browsing databases	SA	A	N	D	SD
77.	Video capture/playback capability	SA	A	N	D	SD
78.	Access to the Internet and WWW	SA	A	N	D	SD
79.	Access to networked CD's from classroom	SA	A	N	D	SD

	SA	A	N	D	SD
80. Ability to create class material for use on the WWW	SA	A	N	D	SD
81. Other ______					

Fairfield University should place high priority on the following services:

82. Up-to-date microcomputer-based instructional labs	SA	A	N	D	SD
83. More mainframes	SA	A	N	D	SD
84. More disk capacity on mainframe (VAX) and servers	SA	A	N	D	SD
85. More powerful network servers	SA	A	N	D	SD
86. Microcomputer classrooms for instruction only	SA	A	N	D	SD
87. Multimedia classrooms for instruction only	SA	A	N	D	SD
88. More laserprinting	SA	A	N	D	SD
89. More documentation	SA	A	N	D	SD
90. More training	SA	A	N	D	SD
91. More consulting support for instruction	SA	A	N	D	SD
92. More consulting support for research	SA	A	N	D	SD
93. More communications (data/voice)	SA	A	N	D	SD
94. Programming for university supported systems	SA	A	N	D	SD
95. Programming for non-university systems	SA	A	N	D	SD
96. Maintenance of department-owned equipment	SA	A	N	D	SD
97. Software maintenance on department equipment	SA	A	N	D	SD
98. More classrooms connected to the networks	SA	A	N	D	SD
99. Support for WWW/multimedia course development	SA	A	N	D	SD
100. More instructional software	SA	A	N	D	SD
101. Ability to transfer large files with sound, images etc	SA	A	N	D	SD
102. Ability to scan and store documents on WWW for instructional use	SA	A	N	D	SD
103. Other(s) ______					

104. There is sufficient support for instructional computing in my department	SA	A	N	D	SD
105. There is sufficient support for instructional computing in the university	SA	A	N	D	SD
106. Within the next five years, computing could improve/enhance the functions associated with my instructional work.	SA	A	N	D	SD

Instructional uses of computing, where appropriate, are assisted by:

107. Sufficient amount of quality software/courseware	SA	A	N	D	SD
108. Sufficient number of available multimedia workstations	SA	A	N	D	SD
109. Sufficient training and development for faculty	SA	A	N	D	SD
110. Sufficient incentives for software development for faculty	SA	A	N	D	SD
111. Software at affordable prices for use on PC networks	SA	A	N	D	SD
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
112. Sufficient data communications capabilities	SA	A	N	D	SD
113. Current personal computer equipment	SA	A	N	D	SD
114. Access to the Internet and WWW	SA	A	N	D	SD
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
115. Access to labs for instruction	SA	A	N	D	SD
116. Access to labs for student to practice/assignments	SA	A	N	D	SD
117. I would use the services of an Instructional Computing group to help faculty use computing for instruction.	SA	A	N	D	SD
118. There is sufficient support for research computing in my department	SA	A	N	D	SD
119. There is sufficient support for research computing in the university	SA	A	N	D	SD
120. I would use the services of a Research Computing group to help researchers use computing in their research	SA	A	N	D	SD

121. I subscribe to Listserves/Bulletin Boards ___ Yes ___ No

122. I use the Internet for the following purposes on a daily basis:

Internet Activity	Percentage of Daily use
Instruction	
Research	
Professional Interest	
Email	
Personal Interest/Surfing	
Total	100%

Department ____________

School ____________

Gender ______

Web-Based Campus Directory

Nancy Thibeault
Computer Services Manager
Miami University
Middletown Campus
4200 E. University Blvd.
Middletown, OH 45042
(513) 727-3355

Abstract

Each semester, our secretaries struggle to update the Campus Directory database in order to print the campus telephone and office/class hours directory. This is a time consuming process and the directory is usually late and out of date by the time it is printed. I have solved these problems by creating a Web interface to their existing database. Now the directory is available on the Web. The information in the Web directory is current and our secretaries' workload has been reduced. Faculty and staff members can print a copy of the latest information as needed.

In the new system, faculty and staff members can update their database information using a Web form. The Web directory provides immediate access to the updated information. Each time the Web directory is accessed, the database is queried and the latest information is displayed.

The Web directory consists of the telephone listing with a link from each entry to a pop up window containing a picture of the faculty/staff member along with their current class/office hour information.

This paper explains the files used to generate the Web-based directory.

Microsoft Visual InterDev

The Web interface was developed using Microsoft Visual InterDev. Visual InterDev is an integrated development environment that provides all the tools you need to develop and manage a sophisticated Web application. It includes integrated programming, database development, site management and content editing tools. The Visual InterDev system requirements are shhown in Table 1.

Visual InterDev is a tool for developing Active Server Pages (ASP). ASP files are HTML files that include scripts that are processed on a Microsoft Windows NT/95 server running Internet Information Server (IIS) with Active Server Pages Extensions. ASP files include VBScript code enclosed in <% %> tags, which is converted to HTML by the server before being sent to the browser. The result is a pure HTML page, which is browser and platform independent.

Visual InterDev provides a complete development environment for creating Active Server applications, including tools that automatically generate much of the script. The tools generate the code to establish the database connection, perform database queries and integrate the data into dynamically generated HTML pages.

System Requirements	Server	Developer	Client
Hardware	Minimum: 486/66 w/16MB RAM Recommended: Pentium w/ 32MB RAM	Minimum: 486/66 w/16MB RAM Recommended: Pentium w/ 32MB RAM	Any computer capable of running a graphical Web browser
Operating System	Windows NT (preferred) Windows 95	Windows NT Windows 95	Any
Software	IIS Active Server Extensions FrontPage Extensions	Visual InterDev	Web browser such as Netscape or Internet Explorer

Table 1. Visual InterDev Systme Requirements

The Database Tables

The Directory Database is a Microsoft Access database that is used by the Campus Administrator's office to produce the campus directory and generate mailing labels, email lists, contracts, etc. The Web interface uses two of the existing database tables: Directory and Class_Hours, shown in Table 2. The two tables are linked on the EmployeeID field. The directory table contains an entry for each faculty/staff member. The Class_Hours table includes office and class hour entries for each faculty member for every time slot. The Type field indicates whether the particular time is an office hour or a class hour. If it is a class hour, then the Course field contains the abbreviation for the course.

Directory	Class_Hours
EmployeeID	EmployeeID
LastName	Type
FirstName	Course
Prefix	StartTime
EmailName	EndTime
DeptArea	Mon
Office_no	Tue
Building	Wed
Extension	Thu
	Fri

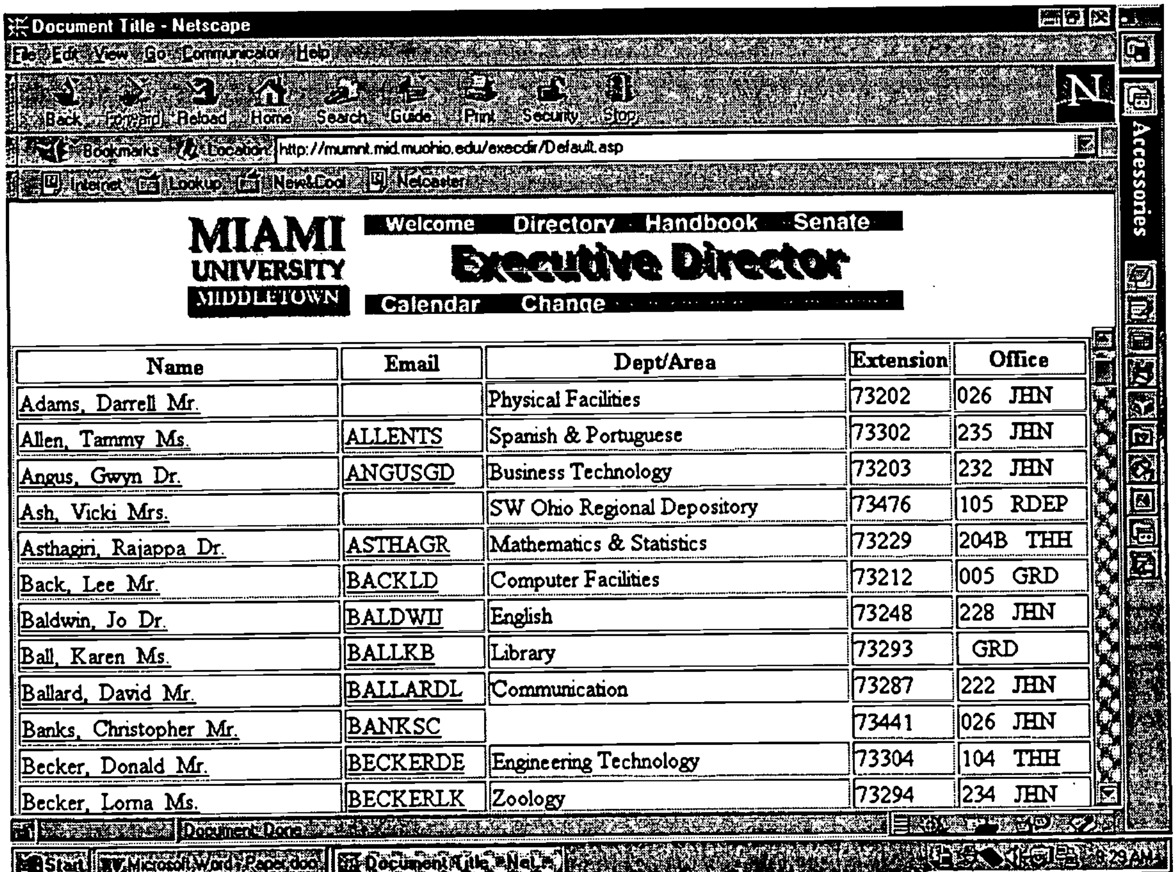
Table 2. Microsoft Access Directory Database Tables

The Web Application

The Web interface to the Directory Database is shown in Figure 1. Each name is a hyperlink to a pop-up window (see Figure 4) containing the faculty/staff member's picture, directory data, and class and office hours information. The Email Name is a mailto link that allows the user to send email to a faculty/staff member simply by clicking on the Email Name.

The Web interface is generated using two ASP files: Table.ASP and OfficeHrs.ASP. Table.ASP consists of a main program that makes the database connection, executes a query on the database then passes the resulting recordset to subroutine GenerateTable which produces the HTML table shown in Figure 1. When a user clicks on a hyperlinked name, the javaScript function openWin is called. The EmployeeID for that name is passed to the openWin function, which in turn calls OfficeHrs.ASP to populate the Pop-Up window shown in Figure 4.

Table.ASP ⟶ GenerateTable ⟶ openWin ⟶ OfficeHrs.ASP



Name	Email	Dept/Area	Extension	Office
Adams, Darrell Mr.		Physical Facilities	73202	026 JHN
Allen, Tammy Ms.	ALLENTS	Spanish & Portuguese	73302	235 JHN
Angus, Gwyn Dr.	ANGUSGD	Business Technology	73203	232 JHN
Ash, Vicki Mrs.		SW Ohio Regional Depository	73476	105 RDEP
Asthagiri, Rajappa Dr.	ASTHAGR	Mathematics & Statistics	73229	204B THH
Back, Lee Mr.	BACKLD	Computer Facilities	73212	005 GRD
Baldwin, Jo Dr.	BALDWIJ	English	73248	228 JHN
Ball, Karen Ms.	BALLKB	Library	73293	GRD
Ballard, David Mr.	BALLARDL	Communication	73287	222 JHN
Banks, Christopher Mr.	BANKSC		73441	026 JHN
Becker, Donald Mr.	BECKERDE	Engineering Technology	73304	104 THH
Becker, Lorna Ms.	BECKERLK	Zoology	73294	234 JHN

Figure 1. Web Interface to Microsoft Access Directory Database

Table.ASP

Table.ASP generates the Web interface shown in Figure 1. The "main program" ASP code is shown in Figure 2. It creates and opens a connection to the Microsoft Access Directory database, then sends an instruction to the database indicating the name of the database query to be executed. The resultant data is returned to the recordset variable rs. Subroutine GenerateTable is then called to generate the HTML page using the data from the recordset parameter rs.

BEST COPY AVAILABLE

```
<%set conn = Server.CreateObject("ADODB.Connection")
set cmd = Server.CreateObject("ADODB.Command")
Conn.Open Session("ExDirConn_ConnectionString")
cmd.ActiveConnection = conn
cmd.CommandText = "Nancy_query"
set rs = cmd.execute
GenerateTable rs%>
```

Figure 2. Table.ASP Main Program

GenerateTable

The GenerateTable subroutine shown in Figure 3 produces the HTML table. Note how the ASP code is interspersed with the HTML tags. The ASP Extensions of the IIS Web server will execute the ASP script, replacing all the rs variables with the actual data.

The GenerateTable subroutine first creates the Table and Headings from the HTML tags. Each pass through the while loop, a record from the recordset is used to produce a table row.

```
<% Sub GenerateTable( rs )%>
<TABLE BORDER=1>
<TR><TH>Name<BR></TH><TH>Email</TH><TH>Dept/Area</TH><TH>Extension</TH>
<TH>Office</TH></TR>
 <%  while not rs.EOF%>
   <TR>
  <td><A HREF="javascript: openWin(<%=rs(0)%>)"> <%=rs("LastName")%>,
           <%=rs("Firstname")%>  <%=rs("Prefix")%></A></td>
          <% if isNull(rs("EmailName")) then%>
                    <td> </td>
          <%else%>
                    <TD>
                    <A HREF="mailto:<%=rs("EmailName")%>@MUOhio.edu">
                    <%=rs("EmailName")%></A>
                    </TD>
          <%end if%>
          <TD><%=rs("DeptArea")%></TD>
          <TD><%=rs("Extension")%></TD>
          <TD><%=rs("office_no")%>   <%=rs("Building")%></TD>
          </TR>
   <%rs.MoveNext
  wend
  Response.Write( "</TABLE>" )
 End Sub%>
```

Figure 3. Subroutine to Generate HTML for Campus Directory

The LastName, FirstName becomes a hyperlink to a javaScript function that opens a pop-up window. When a user clicks on a name a new window is opened displaying the details for that individual as shown in Figure 4.

When the code is executed on the server, each of the <%=rs(x)%> references is replaced with the actual data in the current record of the recordset.

For example, the record

EmpID	LastName	FirstName	Prefix	EmailName
1796	Hurn	Janet	Ms.	HrunJE

will replace the rs variables in the following code segment:

```
<A HREF="javascript: openWin(<%=rs(0)%>)">
<%=rs("LastName")%>,  <%=rs("Firstname")%>  <%=rs("Prefix")%></A>
```

generating the following HTML:

```
<A HREF="javascript: openWin(1796)"> Hurn,   Janet   Ms.</A>
```

A user can simply click on the Email Name in the table to send email to any faculty/staff member.

The Email Name is a mailto link formed by concatenating the EmailName from the recordset with @MUOhio.Edu.

For example:

```
        <A HREF="mailto:<%=rs("EmailName")%>@MUOhio.edu">
```

is replaced with

```
        <A HREF="mailto:HurnJE@MUOhio.edu">
```

The table row is completed using the remaining data from the recordset. This one subroutine writes the HTML for the entire directory table.

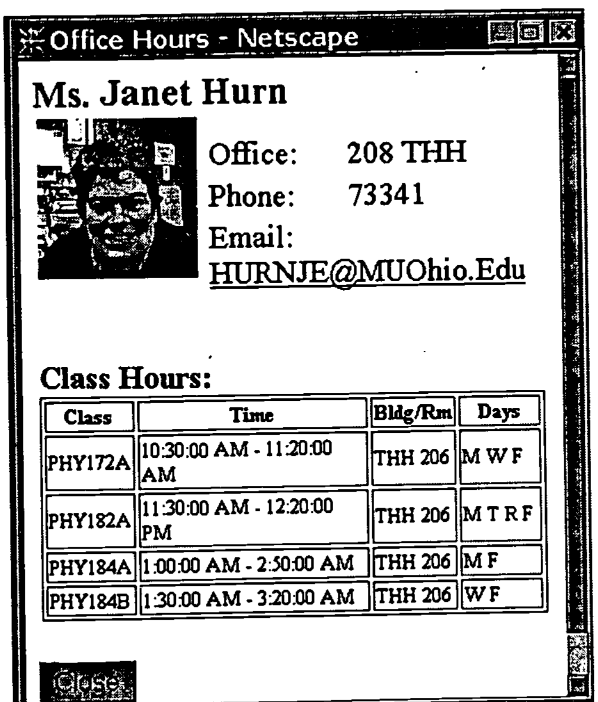


Figure 4. Pop-Up Window

openWin

The javaScript function shown in Figure 5 creates a new browser window which is populated by OfficeHrs.ASP. The resultant window is shown in Figure 4.

"OfficeHrs.asp?EmpID="+EmpID sends a request to the server for the OfficeHrs.ASP page passing the EmpID as a query string parameter. OfficeHrs.ASP then uses the EmpID to select the data for that faculty/staff member.

```
function openWin (EmpID)
        {
        var newWin = window.open("OfficeHrs.asp?EmpID=" + EmpID,
        OfficeHrsWin","toolbar=no,location=no,directories=no,status=no,
        scrollbars=yes, resizable=yes,copyhistory=no,width=350,height=400")
        }
```

Figure 5. JavaScript Function to open Pop-Up Window

OfficeHrs.ASP

The ASP code shown in Figure 6 displays the faculty/staff member's name, picture, office location, email address and phone extension. The ASP code makes the database connection, then uses the EmpID passed in as a query string parameter to create the Select query. The query is executed against the database and the resulting information is displayed in the window. Notice how concatenating the EmailName with the .jpg extension forms the file name for the picture.

```
<img src="images/<%=rs("EmailName")%>.jpg">
```

If the EmailName in the current record of rs is HurnJE, then <%=rs("EmailName")%> will be replaced by HurnJE resulting in

```
<img src="images/HurnJE.jpg">
```

```
<%set conn = Server.CreateObject("ADODB.Connection")
set cmd = Server.CreateObject("ADODB.Command")
Conn.Open Session("ExDirConn_ConnectionString")
cmd.ActiveConnection = conn
cmd.CommandText = "Select Prefix, LastName, FirstName, EmailName, Extension, Office_no, Building from Directory
where EmployeeID = " & request.querystring("EmpID")
set rs = cmd.execute%>
<b><Font Size=+1>
<%=rs("Prefix")%> <%=rs("FirstName")%> <%=rs("LastName")%></Font></b><BR>
<table border=0>
<tr><td>
<img src="images/<%=rs("EmailName")%>.jpg" Width=100 Height=100
ALT="<%=rs("Prefix")%> <%=rs("FirstName")%> <%=rs("LastName")%>">
<BR><BR></td>
<td>
<table border=0>
<tr><td>Office:</td><td> <%=rs("Office_no")%> <%=rs("Building")%></td></tr>
<tr><td>Phone:</td><td> <%=rs("Extension")%></td></tr>
<%if not isNull (rs("EmailName")) then%>
        <tr>
        <td colspan=2>Email: <BR>
        <A Href="mailto:<%=rs("EmailName")%>@MUOhio.Edu"><%=rs("EmailName")%>
        @MUOhio.Edu</A></td></tr>
<%end if%></table></td></tr>
</table>
```

Figure 6. OficeHrs.ASP code to display picture, name, office, email and phone number

Figure 7 shows the ASP code used to generate the Office Hours table in the pop-up window. A query is executed to retrieve the office hours for the selected EmployeeID then displays the table shown at the bottom of Figure 4.

```
<%cmd.CommandText = "Select * from Class_Hours where EmployeeID =
" &request.querystring("EmpID")& " AND Type = 'Office'"
set rs1 = cmd.execute%>
<table border=0>
<tr>
<%if not rs1.eof and not rs1.bof then%>

        <td><B>Office Hours:</B><BR>
        <table border=1>
        <TR><TH><Font Size="1">Time</font></TH><TH>
        <Font Size="1">Days</font></TH></TR>
        <%while not rs1.eof%>
        <tr><td><Font Size="1">
        <%=rs1("StartTime")%> - <BR><%=rs1("EndTime")%></font></td>
        <td><Font Size="1">
        <%val=rs1("Mon") if not val = 0 then%>M<% end if%>
        <%val=rs1("Tue") if not val = 0 then%>T<% end if%>
        <%val=rs1("Wed") if not val = 0 then%>W<% end if%>
        <%val=rs1("Thu") if not val = 0 then%>R<% end if%>
        <%val=rs1("Fri") if not isnull(val) then%>F<% end if%>
        </font></td></tr>
        <%rs1.MoveNext%>
        <%wend%>
        </table>
        </td>
<%end if%>
```

Figure 7. OfficeHrs.ASP code to generate office hours table

The query retrieves all office hour records for the selected EmployeeID. The resultant recordset is stored in rs1. Each record of rs1 is then displayed in table format.

Summary

Visual InterDev is an excellent tool for developing database driven Web applications. It contains all the tools needed to create and test your application in a single integrated environment; however, it has a fairly steep learning curve. The wizards and visual tools are great for generating basic applications, but I found that you must understand HTML and ASP in order to fine-tune and create customized applications. The wizards generate too much unnecessary code that makes the applications difficult to understand and debug. I prefer to write my own code. It is a great development environment, but you need a background in HTML, programming and databases in order to use it effectively.

References

Teach Yourself Active Web Database Programming in 21 Days, Sams Net, 1997. Fleet, Warren, Chen, and Stojanovic.

Professional Active Server Pages, Wrox Press, 1997. Alex Homer, etal.

Special Edition Using Visual InterDev, Que, 1997. Mike Morrison

Inside Microsoft Visual InterDev, Microsoft Press, 1997. Ken Miller, Kenneth Spencer, and Eric Vincent.

Microsoft Visual InterDev Web Site. www.microsoft.com/vinterdev.

Open Platform Software System for Internet Interactive Education

Dave Thomas
I-CARE
dthomas@pride-12.poly.edu

I-CARE = Internet - Cyberspace Assisted Responsive Education

An Innovative and Flexible Internet-Based System for Education and Training

I - CARE is an educational solution and has three main components:

Course Development
Course Delivery
Course Evaluation

I-CARE is the Internet-based version of CARE (Cyberspace Assisted Responsive Education) and is a computer mediated system for bringing teachers together in a virtual learning environment. It provides new, interactive tools, a stimulating multimedia environment, and group learning techniques.

The I-CARE course is modularized with each module covering a subject domain at a given starting level. Hypertext lectures and interactive exercises with instructors and peers are the vehicles of teaching and learning. Students are able to learn at their own pace but are also expected to progress along with the group. Groups made up of faculty will teach student groups globally. Students are able to forge ahead and extend their learning beyond the scope of the present module, but are also encouraged to take the lead and to help others to reinforce what they have learned.

Tools available to students include: a bulletin board, chat session, e-mail, faculty appointment book, search tools, on-line library, group discussion and homework assignment depository.

The I-CARE system can be viewed at http://pride-sun.poly.edu/icare/

An Integrated Computerized Instructional System for Classroom and Lab

Ray Treadway
Department of Mathematics and Computer Science
Professor of Mathematics
Bennett College
900 E. Washington Ave.
Greensboro NC 27401
336-370-8739
RAYT@PALS.GUILFORD.EDU

Bennett College has developed the use of an integrated computerized instructional system featuring the use of large networked test-banks, a sophisticated test management system, application software, and multimedia. These materials are used both for interactive learning experiences in the classroom and for practice, drill, and tutorial in the computer laboratory. The system involves students in the learning process by providing interactive questions from the test-bank and visual presentation of concepts. Thus far the materials have been used mostly in first-year courses where specific information and techniques for problem-solving are emphasized.

The integration of test-banks for computer-based testing, textbooks, and electronic lecture notes in first-year courses allows the delivery of a variety of teaching methods and hence caters to a variety of learning styles. The system is not complicated, nor does it require that the use of Internet and sophisticated software and program development, but it is effective. At Bennett College, it is used on local area networks in two computer classrooms, with about thirty computers and a projection system in each. An integrated computerized instructional system (ICISS) has been set up in both labs so that a variety of teaching/learning approaches can be used. In the classroom direct lecturing, writing on the board, class discussion, and seat work is combined with computer drill and practice, and class projection of multimedia lecture materials. In addition learning lab activities and individualized use of these materials round out the entire teaching/learning package.

At Bennett, the main use of the materials has been in teaching a variety of first and second year mathematics courses. Currently the package of materials is also available in first-year chemistry and biology. These multimedia materials are produced by EDUCO International, Inc. of McDonough GA. In other disciplines multimedia software is often available on the market from textbook publishers. While the integration of topics among the textbooks, test-banks, and electronic lecture notes is convenient for the student and the lecturer, it is not crucial. What is distinctive and significant is the computerized set of practice/test questions with hints and solutions.

INTEGRATED COMPUTERIZED INSTRUCTIONAL SUPPORT SYSTEM

The Integrated Computerized Instructional Support System was developed by Dr. Man Sharma, at Clark-Atlanta University. This project has been supported by several grants from the US Department of Education through the Minority Science Improvement Program. Materials are distributed by EDUCO International, Inc. McDonough GA. Approximately 100 institutions are using or have used these materials. Bennett College has been involved with the ICISS project since 1988.

Question Banks and the Graphics Editor

The first component of the system is a set of extensive multiple-choice test-banks for each course. Most of the questions have hints and solutions so that students can use the test-banks for tutorials outside class or they can be used for short timed quizzes during the class period. Such frequent short quizzes are particularly useful in courses such as elementary algebra or foreign languages where quick recall of facts and procedures are tested. Students who do not do well on the first try may be allowed to make up the quiz at the discretion of the instructor. The questions banks have been installed on the network and records of the students' scores on quizzes are kept in "result" files. The instructor can also generate multiple versions of the test to be printed out and used as hard-copy tests. These can be saved as a text file and modified. A graphics editor (MEDIT.EXE) with many mathematical symbols available is used to add questions or modify questions in the test-banks.

The use of these tests as practice and then for quizzes is a valuable feature, particularly when learning the basic skills and facts of the content of a course. Since the questions of the testing system are multiple choice, the types of questions that can be asked are limited. However, the capability to include hints and complete solutions greatly bolsters the use of the materials for practice. Short-answer printed tests can easily be created, as indicated above, by modifying the text-file tests by deleting the multiple choices and leaving sufficient blank spaces for the student to write out the answer.

Crucial to the system is the development of a large set of questions with several similar versions. This can be done over a period of time as tests and final exams are routinely prepared. Questions are chosen by a set of codes and by randomization.

File Management and the Test / Menu Generator

An 8-digit coding system is used and this allows the instructor to select the types of questions selected at random by the computer. Usually the coding system has the form:

<Course Chapter Section Concept Question No (2 digits) Version Number (2 digits).>

Hence a question with code 21323003 represents the second course (e.g, Elementary Algebra, Biology 102), chapter 1, section 3, topic 2, question 30, and version 03. A practice test consists of a collection of four digit codes, so that the instructor can select questions within a particular topic, but the questions and versions are selected at random for presentation to the student.

The ICISS system includes several programs that support the testing/tutorial process. It is not necessary that the instructor knows how to use these programs, but the laboratory technician will need to spend some time learning the procedures. A file management program (FMS) helps in the management of the question banks, by indexing the questions, checking technical errors, and creating various subsets and summaries of the test-banks. A single program (MENUGEN) is used to generate the individual tests and to create the menu which is displayed when students come for practice and quizzing. When the a student takes the test, one of the many questions with the appropriated 4-digit or 6-digit code is selected at random. With a suitably large test bank, the students can practice several times and will see relatively few questions twice. The instructor can "activate" tests for selected periods of time. An electronic class roll prevents unauthorized students from taking the test.

When MENUGEN is run, a list of possible actions is shown. To create a practice test, one version of each 4-digit question is displayed and the particular questions to be included in the practice can be selected. Likewise to create a quiz, one version of each 6-digit question is displayed. When the process of creating a test is completed, the codes for the test are saved and the test is added to the test menu for the student.

As indicated above, tests may be prepared for students on two levels: review questions with no score and quizzes for which the score is saved. For the review questions (called practice tests), students may practice any number of times and each time different questions will be provided, depending on the number of similar versions in the test-bank. The time spent on the test is recorded but the score is not. Hence the students may work on questions and study the hints and solutions without worrying about the stigma of getting a low grade.

When the student takes the review test (tutorial), hints and solutions are provided, and the student gets immediate feedback. An actual quiz consists of a collection of six digit codes, so that the instructor can select specific types of questions; the only difference between the tests for one student and another is the version of the question. The degree of difficulty of the same quiz given to different students will be the same. Moreover, the amount of time allowed to complete the test is specified in setting up the quiz. When a student takes an actual test, answers can be changed as on a paper test, but the student will not be able to review missed questions and look over the solutions until he or she finishes the test. If the instructor wishes to see the student's work while taking a test on the computer, the student can be directed to show work on paper while taking a test on the computer. The computer simply provides questions easily and conveniently.

Course Management System, Record Management System, and Item Analysis

The Course Management System (CMS.EXE) produces an electronic grade book, so that scores can be entered into the grade book quickly and easily directly from the file server. This is a powerful feature for large classes. This automatic transfer feature saves a great deal of time for classes, provides complete accuracy in recording scores, and allows for frequent short quizzes. Besides electronic transfer of scores on tests to the class roll, the course management system allows for manual entering of grades, a wide variety of methods to determine grades, and periodic progress reports to students. These periodic progress reports have proved popular among the students, since the reports indicate to the students exactly where they stand in the class and allow the students to make corrections if some scores are inaccurate. Another program provides an item analysis of the test results for each type of question.

Electronic Lecture Notes

Electronic Lecture Notes have been installed for Basic Mathematics, Beginning Algebra, and Pre-calculus (including College algebra and Trigonometry). Chemistry and Biology Notes are also available for first-year college courses. The electronic lecture notes, (also known as EDUCO Electronic Tutors) are based on Asymmetric's Toolbook. They provide extensive use of dynamic graphs, animation, attractive screen designs, and colorful display of concepts and strategies for solving problems. (See examples of typical screens at the end of these notes). Textbook publishers are often including similar multimedia materials to the instructor upon the selection of a textbook, and commercial materials are available, sometimes with a substantial price tag. Text and diagrams are presented a little bit at a time, and the student can "click" on various links or proceed in a step-by-step manner. Examples are worked out in detail with explanations, again one step at a time. The electronic lecture notes can be used by the instructor for display to the entire class as a substitute for the chalk and talk approach. They are particularly effective when animation or visualization is required. Use of the electronic lecture notes allows teachers to stay well organized and informed and to reduce time spent in writing on the board. Since the electronic lecture notes are available to students after the class is over, the students can concentrate on the discussion and do not need to take notes. Electronic Lecture Notes allow the instructor to present a variety of kinds of activities in class. I use them in place of lecture for twenty or twenty-five minutes once or twice a week.

The electronic lecture notes can also be used by students in the computer lab as tutorials complementing class lectures. If a student misses a class, she or he can easily make up the work by using the software. The material corresponds to text book materials. They can also be used as enrichment for material not covered in class for the advanced student. Guided instructions for students to use the electronic lecture notes can be prepared and distributed. These instructions can indicate in detail what the students should cover on the computer, ask them to write down the basic results and examples shown, and direct them to work problems similar to those displayed. This approach is particularly useful when the instructor is away from campus or after a quiz. In the latter situation, these guided instructions are used when students finish at various times and can begin studying for the next topic while others finish the quiz.

Conclusion

Much of the education hype today is directed to the use of the Internet for the new focus in teaching, sometimes exclusively. At the same time, many instructors continue to teach in traditional ways, writing notes on a blackboard and lecturing without technology. The narrative above tries to show that one does not have choose between the chalk-and-talk methodology and the telecomputing methodology but can and should mix a variety a teaching strategies involving lecture and discussion, electronic lecture notes or other visual materials with a projection system, and individualized tutorials and explorations on the computer as part of classroom and learning lab activities. The software packages, particularly the ICISS programs, provide powerful but simple tools for practice and drill, tutorials, and multimedia materials. Students who need support appreciate the chance to practice and to receive help. It is not difficult to place these materials on a local area network.. The combination of materials allows the students to take a great deal of control for their own learning, and this should be one of the goals of any educational program.

References

ICISS Software, textbooks, and the Electronic Tutor Version 2.0 (which includes courses in Basic Mathematics, Algebra Part I (Elementary Algebra), Pre-calculus, Calculus I, Calculus II, Calculus III, General Chemistry, and General Biology) are available from EDUCO International Inc., 1948 Chambers Rd., McDonough GA 30352. Phone 1-800-963-3826, FAX 1-770-954-9060.

ICISS Manual (MEDIT, FMS, RMS, IAN) EDUCO International Inc. and Clark-Atlanta University, (McDonough GA: 1998)

PCTEST and Course Management System Manual, EDUCO International Inc. and Clark-Atlanta University (McDonough GA: 1998)

"Multimedia in Education", a series of articles in Syllabus 9 (8), May 1996.

When Teaching Becomes Instructional Support

William P. Wilson
Director of Instructional Technology and Training
Gettysburg College
Gettysburg, PA
(717) 337-6933
wilson@gettysburg.edu

Over the last 20 years we at Gettysburg College have tried a lot of models to support instructional integration of computing and/or technology into coursework. As the nature of the delivery system has changed, so have the models. With the filling out of staff positions in our ITT group, it has given me the opportunity to get back in the classroom to teach in my academic discipline, Psychology. I elected to teach the introductory psychology course for a number of reasons. First, it wouldn't be so easy for students to see how much I have forgotten. More seriously, I elected to teach this course because I saw in it the opportunity to better prepare psychology students for the interactions they would have with technology in their future psychology courses. I also saw a chance to provide a model that people could take pieces from in many introductory courses throughout the curriculum.

Process

We will have a traditional textbook with non-traditional study guide. Study guide materials will be loaded into course delivery software (TopClass). TopClass provides a full-featured, web-based instructional environment which includes delivery of course material, class discussion forums, test-taking, test scoring, and results-based branching to additional learning material. Papers will be electronically collected, marked up, and comments will be electronically returned. They will learn how to use relevant library resources, how to search and deal with electronic journals, they will learn to graph data, and use our network to full advantage. One project will be for the class to create a website that will be the basis for an organically developing Gettysburg College introductory psychology site that can be added to as future introductory classes add content. These latter activities will be active learning exercises that will take place in a computer lab one time a week as one of the regular meetings of the course.

Faculty Learning

As I prepare for this class, I will be meeting with the rest of the Psychology department to be sure the skills I am targeting are the right ones and will be used later in the curriculum. I will also hold monthly meetings with interested faculty, probably in a brown bag setting, to review what is happening in class and to show faculty what I am using and some of the outcomes we are getting from the students. One of the important potential outcomes is to allow faculty to pick and choose elements of what I am doing in class. For example, almost everyone has students word process their papers, but do they use the special comment and mark up features of the word processor to provide feedback to the students?

I am planning some high-tech features to this class and faculty need to be aware of some of the advanced things we can provide over networks now. I will be preparing realvideo clips of many psychologists talking about specific topic areas that will bring experts from across the US into virtual contact with my students in Gettysburg, PA! I will show faculty how to take their PowerPoint presentations and add narration and turn the presentation into a virtual lecture or review. Faculty need to be exposed to a variety of ideas that will in turn get them thinking about their courses and beginning to take what they see and modify it here and there to fit into their level of comfort and their course.

Outcomes

I will take advantage of a Friday Faculty lunch series that we have on campus to report on this course and to hopefully show our faculty some of the successes of the course and be ready to help people adapt some of the approaches I used into their own courses. We will base some of our faculty training program on some of the technologies I used to try to leverage the course and any interest it generates among the faculty. We will also try to assess whether this is an effective model to help get faculty to adopt some technology into their courses compared to other approaches.

The remainder of my presentation will introduce some of the software I will be using and some of the student assignments I foresee having these student tackle.

Presenters Index

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket")..