DOCUMENT RESUME

ED 405 861 IR 054 771

TITLE	Proceedings of the ASCUE Summer Conference (26th Annual, Myrtle Beach, South Carolina, June 20-24, 1993).
INSTITUTION	Association of Small Computer Users in Education, Greencastle, IN.
PUB DATE	93
NOTE	247p.; Pages 45-49 withdrawn by ASCUE.
PUB TYPE	Collected Works - Conference Proceedings (021)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	Computer Oriented Programs; *Computer Uses in Education; *Educational Technology; Elementary Secondary Education; Information Networks; Information Technology; *Microcomputers; Multimedia Materials; Postsecondary Education; Professional Development; Strategic Planning; Telecommunications
IDENTIFIERS	Interactive Courseware

ABSTRACT

This proceedings report includes 37 papers presented at the 1993 presented on the following topics: information technology in college recruiting; introductory networks in the classroom; Total Quality Management in higher education and a computing services organization; a High Tech Student Workstation; network communication for students and faculty; telecommunications using the Apple Global Education network; a computer component in a college core program; Spreadsheet versus general ledger accounting; solutions to "CD-ROM anxiety"; technologies planning in small colleges; multimedia in the classroom; Artificial Intelligence versus Information Processing in melody recognition; a local area network college bulletin board; "for credit" lab technicians; touch screen building directories; computer uses in professional education and staff development; a computer academic enrichment program; building a client/server environment with technology; teaching CS2 with Ada; teaching Matrix Algebra with an interactive software package; a multimedia tool for instructional development; configuring a computer lab for multimedia applications; laptop use by professors in the classroom; a capstone computer course; a computer literacy course; system maps and embedded cognitive strategies in educational hypermedia; computer technology for elementary and middle school students; a microcomputer purchasing program; comparison of database systems; a computer-based differential equations course; an interdisciplinary technology classroom; a Faculty Innovation Center; networking classrooms and learning centers; math visualization with Spreadsheets; and case studies of technology for instruction. This volume also includes information about ASCUE, a list of ASCUE board members, and information on the keynote speakers and preconference seminars. (Author/AEF)

ASCUE

ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION

"Our Second Quarter Century of Resource Sharing"

PROCEEDINGS OF THE 1993 ASCUE SUMMER CONFERENCE

26th Annual Conference

June 20-24, 1993

Myrtle Beach, South Carolina

ASCUETUG

CUETUG (College and University Eleven Thirty Users Group) 1967-1975

ASCUE

ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION

"Our Second Quarter Century of Resource Sharing"

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 Users' Group, with an initial membership requirement of sharing at least one piece of software each year with the other members, ASCUE has a strong tradition of bringing its members together to pool their resources to help each other. It no longer requires members to share homegrown software, nor does it have ties to a particular hardware platform. However, ASCUE continues the tradition of sharing through its national conference held each June, its conference proceedings, its newsletter, and its sponsorship of NECC, the National Educational Computing Conference. ASCUE proudly affirms this tradition in its motto "Our Second Quarter Century of Resource Sharing."

ASCUE's ASCUEL LISTSERVER

Subscribe by sending the email message SUBSCRIBE ASCUEL yourname to listserv@purccvm. The listserver itself is ascuel@purccvm.

NEED MORE INFORMATION?

Direct questions about the contents of the 1993 Conference to Rick Huston, Program Chair, ASCUE '93, rhuston@univscvm.csd.scarolina.edu, University of South Carolina - Aiken, 803-648-6851 ext. 3422. Also, subscribe to ASCUE's listserver, ASCUEL. Details about future conferences will be sent to the listserver.

ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION
ASCUE 1992-1993 BOARD MEMBERS
(years remaining in office)

PRESIDENT
Carl P. Singer (1 year)
Computing Services
DePauw University
Greencastle IN 46135
317-658-4294
runner@depauw

PROGRAM CHAIR
F.P. (Rick) Huston,III (1 year)
Computer Services Division
University of South Carolina - Aiken
171 University Parkway
Aiken SC 29801
803-641-3422
8036413494 (FAX)
rhuston@univscvm.csd.scarolina.edu

TREASURER
Thomas A. Pollack (2 years)
Associate Dean/Director of
Undergraduate Studies
Duquesne University
600 Forbes Avenue
Pittsburgh PA 152820104
412-434-1639
pollack@duq3.duq.edu

SECRETARY
Dagrun Bennett (1 year)
Computing Services
Franklin College
501 East Monroe
Franklin IN 46131
317-738-8150
bennett@minnie.franklincoll.edu

IMMEDIATE PAST PRESIDENT
William P. Wilson (1 year)
Academic Computer Center
Gettysburg College, Box 439
Gettysburg PA 173251486
717-337-6933
wilson@gettysburg.edu

BOARD MEMBER AT LARGE
Mary V. Connolly (1 year)
Mathematics Department
Saint Mary's College
Notre Dame IN 46556
219-284-4497
connolly@saintmarys.edu

BOARD MEMBER AT LARGE
Thomas E. Gusler (2 years)
Assistant Academic Vice President
Clarion University
103 Carrier Hall
Clarion PA 16214 p?3
814-226-2676

NEWSLETTER EDITOR
Don Armel (1 year)
School of Technology
Eastern Illinois University
Charleston IL 61920
217-581-2216
cfdja@eiu.edu

PUBLIC RELATIONS
R. Waldo (Wally) Roth (1 year)
Computer and System Science Dept.
Taylor University
Upland IN 46989
317-998-5269
wlroth@tayloru

LIBRARIAN/HISTORIAN
Jack Cundiff (1 year)
Horry Georgetown Technical College
P.O. Box 1966
Conway SC 29526
803-347-3186
r010125@univscvm.csd.scarolina.edu

TABLE OF CONTENTS

About ASCUE 1
ASCUEL listserver 1
Need More Information? 1
Board Members 2
Keynote Speakers 7
Fred T. Hofstetter -- Multimedia
George Bateman -- TQM
Preconference Seminars 7
Networks -- Bill Hancock
Toolbook -- Lorinda L. Brader
Problem Solving -- Mary English and Carl Singer
Internet -- Ernie Ackerman and Bill Wilson

Papers

Information Technology and Recruiting 10
John A. Anderson
Loras College

Experiences of Introductory Network
Use in the Class Setting: Development and Implementation 20
Donald Armel
Eastern Illinois University
Constantine Roussos
Lynchburg College

Total Quality Management in Colleges and Universities 25
Thomas B. Auer
Murray State University

High Tech Student Workstation in the Biomedical Sciences 27
Thomas G. Basler
Nancy McKeehan
Nancy Smith
W. Curtis Wise
Medical University of South Carolina

Unique Communication Opportunities for Nursing Students and Faculty 34
Kimberly D. Bone
Tamara L. Simpson
University of South Carolina

The World of AGE, a Telecommunications Project 39
Dr. Marvin Bratt, Director
The Ohio State University

Jump Start the Use of Computers in the Classroom 42
Kim S. Breighner
Gettysburg College

Spreadsheet vs General Ledger Accounting Program in the Accounting Curriculum 50
Clarence Coleman
Charles Doryland
Francis Marion University
Randy Reid
Central Carolina Technical College

CD-ROM Anxiety: Finding Problems, Searching Solutions 55
Anne L. Fliotsos
Winthrop University

"CONNECTing" with Campus Staff 62
Thomas L. Hallman
University of South Carolina - Aiken

Technologies Planning and the Small College 65
David C. Bonner
John Hammer
Jane C. Neale
Kevin A. Synnott.
Russell Sage College

Multimedia in the Classroom 76
Carl W. Helms
Doris R. Helms
Clemson University

Machine Models of Human Pattern Recognition - The Case of Melody Recognition 83
William J. House
University of South Carolina - Aiken

Networking Made Easy: The Bulletin Board to the Rescue! 87
Louis M. Janelle, Jr.
The University of Akron - Wayne College

Managing Computer Labs with "For Credit" Lab Technicians 89
Dan W. Krupp
Indiana Vocational Technical College

Lander's Touch Screen Building Directory 95
Terry W. Lancaster
Lander University

Computer Integration in Professional Education: A Case Study of A Regional Institution of Higher Education 98
Charles D. Manges
Larry Harris
Wayne State College

Computer Usage in an Academic Enrichment Center 101
Mabel Jones Matthews
Saint Augustine's College

Moving to Client/Server Computing: How Do You Get There? 116
David K. Moldoff
Applied Business Technologies, Inc.

Teaching CS2 With Ada 130
Dr. Patricia T. Montague
Metropolitan State College of Denver

Teaching Matrix Algebra Using MATLAB 151
Dr. Patricia T. Montague
Metropolitan State College of Denver

LanguageWriter: A Multimedia Tool for Instructional Development 167
Robert L. Oakman
Jay A. Waller
Reginald A. Riser
Fred Fenimore
University of South Carolina

Configuring A Computer Lab For Multimedia Applications 172
Thomas A. Pollack
Duquesne University

The Use of Laptops by Professors
in the Computer Science and Systems Classroom 178
R. Waldo Roth, Chairman
Taylor University

The Senior Capstone Course for Computer Science
and Computer Information Systems Majors 183
Donna Satterfield
Franklin College of Indiana

Computer Literacy - An Electronic, Paperless Course 187
Donna Satterfield
Franklin College of Indiana

Using Hypermapping and Embedded Cognitive
Strategies in the Development of Educational Hypermedia 191
Gary J. Senn
University of South Carolina - Aiken
Phillip B. Horton
Florida Institute of Technology

Computer Technology For Elementary and Middle School Students 195
Gary J. Senn
Jeffrey M. Priest
University of South Carolina - Aiken

The Development of a College/University Microcomputer Purchasing Program: A Conceptual Model 198
Lisa J. Shepherd
William J. Moressi
Don E. Gardner
Winthrop University

Comparison of Database Systems 214
Thomas J. Smith
Manhattan College

A Computer-Based First Course In Differential Equations 218
Thomas Smith
Charles Stolze
Kathryn Weld
Manhattan College

Development, Implementation and Incorporation of an Interdisciplinary Technology Classroom for Science and Math 223
Paul S. Tabor
Clarke College

The Faculty Innovation Center Ideas, Training, & Development Workshop 226
Robert Teague
Terrance Keys
Monroe Community College

Networking Classrooms and Learning Centers Better Facilities With Fewer Resources 229
Robert Teague
Terrance Keys
Brett Thompson
Monroe Community College

Math Visualization With Spreadsheets 232
David S. Tucker
Midwestern State University

Does Technology Improve Instruction? Case Studies in Networking and Multimedia 249
Mary Wall
Atlantic Community College
Barry Centini, SRI

Working Smarter: TQM Principles Applied to an Overworked Computing Services Organization 254
William P. Wilson
Gettysburg College

KEYNOTE SPEAKERS

Fred T. Hofstetter is Director of the Instructional Technology Center at the University of Delaware. A specialist in multimedia computing for music composition, desktop publishing, interactive video, digital television, compact disc, handheld scanning, computer-based instruction, and classroom media projection, he developed the PODIUM hypermedia program for IBM's multimedia solutions. As a full professor of Music and Educational Studies, he authored the GUIDO EarTraining Lessons, the Random House book Making Music on Micros, and the Prentice Hall book Computer Literacy for Musicians. Professor Hofstetter is the principle investigator of the Videodisc Music Series, funded by the National Endowment for the Humanities and winner of a gold CINDY award.

George Bateman, Senior Lecturer in Quality Management and Statistics in the Graduate School of Business at the University of Chicago, has thirty years of experience consulting in the area of quality. With his twenty-five years of university teaching and his fifteen years of senior management responsibilities in information technology at the University of Chicago, Professor Bateman is uniquely qualified to discuss the blending of Total Quality Management, Higher Education, and Information Technology. In the information technology area, he has served on advisory boards to Apple, Microsoft, NeXT, Claris, and IBM.

PRECONFERENCE SEMINARS

Preconference Seminar 1
Network Design and Implementation for Campus Networks

Presented by:
Bill Hancock, Vice President, Network1, Inc.

This fullday seminar covered the essential information for designing and implementing campus networks in the educational environment. Topics included general design problems and solutions, current state-of-the-art network implementation techniques, cable plant design, cable selection criteria, software protocol issues and selection criteria, performance, management of campus networks, and many other related topics necessary to properly implement campus networks. Case studies on successful campus network designs were provided as well as methods in starting with a small network and expanding it to a full campus-wide network.

Intended Audience:
This seminar was intended for anyone implementing, expanding or contemplating the installation of a campus network at an educational institution.

About the Presenter:
Bill Hancock is an internationally respected consultant and engineer with many years of experience in network design, management, troubleshooting, security, and standards. A prolific network designer and product architect, Mr. Hancock has designed over 4000 networks ranging in size from two nodes to over 400,000 nodes on a single network. Mr. Hancock is the recipient of the Digital Equipment Computer Users Society (DECUS) Technical Excel-

lence Award, the top technical achievement award given by the U.S. Chapter. A former ANSI and ISO standards committee member, Mr. Hancock has worked on many international network standards and authored numerous network articles and books on network technologies and standards. Mr. Hancock brings a unique first hand insight to network architecture, management and troubleshooting techniques, performance and tool requirements.

Preconference Seminar 2
Introduction to ToolBook (Multimedia and More in the Windows Environment)

Presented by:
Lorinda L. Brader, Faculty Computing Support Center Manager
Academic Computing, East Carolina University

This workshop provided an overview of Asymetrix Corporation's ToolBook development system and provided hands-on experience using ToolBook to create personal applications. ToolBook is a graphical development environment that enables non-technical users to build Windows applications. The workshop introduced and discussed the basic concepts and features of ToolBook, including books, pages, backgrounds, fields, buttons, hotwords, and graphics. To demonstrate the range of capabilities available, examples of ToolBook applications, including advanced multimedia applications, were shown. Workshop participants developed the skills necessary to manipulate the objects of ToolBook to create their own simple applications for presentations, training or personal use.

About the Presenter:
Lorinda Brader worked as an instructional courseware developer for Penn State for 5 years. In 1991 she received three awards for courseware she developed with teams at Penn State 2 EDUCOM Higher Education Software Awards and the Association for Educational Communications and Technology (AECT) award for Outstanding Practice in Instructional Development. The EDUCOM awards were for the courseware "Background and Analysis of Martin Luther King's Letter From Birmingham Jail" and "Evolution of Spanish Sounds Part II: Consonants in Combination" while the AECT award was for the courseware "Diagnosing Reading Abilities: Case Study Simulations".

Preconference Seminar 3
Using Computing to Enhance the Learning Process: Problem Solving and Pedagogy

Presented by:

Mary P. English	*Carl P. Singer*
Asst. Professor of Economics	*Professor of Computer Science*
DePauw University	*DePauw University*

This was a hands-on, interactive seminar about developing better methods to teach through the use of computers. The workshop focused on exciting ways to use new and traditional software to enhance the learning process. Several packages were used in the seminar, including PCSolve, an easy to use numerical mathematics package. Many different examples of

PCSolve related course material were presented. The examples were based on actual assignments in the areas of economics, physics and an introductory course in quantitative reasoning (problem solving). After an introductory tutorial, participants were given the opportunity to learn PCSolve in the same manner as the student (through content related assignments). This format provided the participants with the opportunity to experience many different pedagogical techniques that might be found in courses having a computer component. The workshop also provided examples using software packages such as QuattroPro, Turbo Pascal, Derive, WordPerfect, and Vax Notes. Courses which use the computer in different ways were discussed: a laboratory centered course (computer science), a lecture oriented course with a lab component (physics, mathematics, chemistry, and economics), and a lecture course without a lab component (economics, English).

About the Presenters:
Although this was Mary English's first year attending ASCUE, Carl Singer has been a member of ASCUE and its Board of Directors for years. Carl was Program Chairman for the 1992 ASCUE Conference and is ASCUE's President this year.

Preconference Seminar 4
Internet Tools and Services

Presenters:

Bill Wilson
Advanced Software Support
Gettysburg College

Ernie Ackermann
Coordinator, Computing Services
Mary Washington College

This workshop was a hands-on session that took participants through the many services available on the Internet and demonstrated various ways of accessing these tools. Included were discussions of interest lists, electronic mail, telnet, ftp, gopher, wais, and World Wide Web. Hints for how to locate interesting sites on the network were also discussed. The presenters compared and contrasted some of the software tools now in use to assist the Internet voyager, with an eye toward what it takes to maintain some of them locally.

Intended Audience:
This seminar was designed for people new to internet and for people looking for tools to make navigating the networks easier for their users as well as for people looking for information to help them justify moving from BITNET or augmenting their BITNET service.

About the Presenters:
Ernie Ackermann and Bill Wilson are ASCUE members who gave a brief overview of the internet in a joint talk at the 1992 ASCUE Conference. Bill has also served on ASCUE's Board of Directors for years, was the Program Chairman for ASCUE's 1991 Conference, ASCUE's 1992 President, and is ASCUE's Immediate Past President this year.

Information Technology and Recruiting

John A. Anderson
Director of Information Services
Loras College
1450 Alta Vista, Dubuque IA 52001
319-588-7212
hawks@1cac1.loras.edu

Background

Loras College is a four-year, private, coeducational, liberal arts college affiliated with the Catholic Church. It is located on a bluff high above the Mississippi River where the states of Iowa, Illinois, and Wisconsin meet at one point, in the city of Dubuque, Iowa. The 1800 to 1900 men and women students typically come from Iowa, Illinois, Wisconsin, and Minnesota. Usually 25 or more states are represented along with 10 or more countries. A majority of students are from either the Dubuque or Chicago areas.

Prior to the last three years, Loras would routinely recruit approximately 500 freshmen per year. We were able to utilize the old standard methods of recruiting sons and daughters of alumni and targeting the reliable recruiting territories of Chicago and Dubuque. At that time, we obtained many prospects by purchasing names of high school students in the high volume targeted areas. For a variety of reasons, the effectiveness of this technique declined drastically. As a result, we started buying less and less names from areas that we felt we were not going to be able to compete, either because distance made Loras basically unknown or weak recruiting history in those areas. Our pool of prospects gradually dwindled from approximately 8,000 prospective students to only a little over 5,000 in a period of three years. This situation was caused by a number of factors besides efficiency concerns, among these are:

- dwindling numbers of college eligible high school graduates due to population trends of 18 year olds
- less funds to purchase names of students and communicate effectively with all of them
- increased competition between private colleges, public universities, and community colleges
- desire for higher quality students as students are becoming more selective about their academic future

As a result of this smaller pool of prospects to choose from and an Admissions department that decided to resign one-by-one until there was only one original person left, we ended up having less than 500 new students including less than 400 new freshmen for the fall term of 1991. This followed a recruiting year of slightly more than 600 new students and almost 500 new freshmen for the fall term of 1990.

Probably the most important factor was the resignation of our Director of Admissions after more than 10 years of service at our institution. In the next six months, our Admissions staff saw everyone leave the institution from Admissions counselors to data processing clerks. It became obvious that our recruiting effort at the college needed some major overhauling and now was the time to do it, with a need to retrain personnel, anyway.

Recruiting Assessment

The record low numbers of new students in the fall of 1991 provided Loras College with a wakeup call of what was to become the new world of Admissions recruiting. It was apparent that the methods being employed by Loras in recruiting students was archaic and totally ineffective in the face of the new factors. As colleges and universities found it necessary to streamline and become more efficient in all their operations, they also found that the recruiting game also had the rules changed. The approach taken prior to this time appeared too simplistic for the complexities involved in demonstrating the best qualities of our institution to prospective students and convincing them that attendance at Loras was the right choice for their future. What we were seeking was a new marketing plan that would allow us to change the direction of our recruiting effort as swiftly as it had declined.

This paper will deal with the role played by the Department of Information Services in the revitalization of the Admissions department. It was important to evaluate all aspects of information flow that existed and to determine the strengths and weaknesses in all stages of a prospect's development into a confirmed student. This project also marked a new view of the role that our programming staff would play in many other sectors of the administration. It is becoming increasingly important for the computing staff to provide users with more than just the tools for entering data and providing reporting, but also to provide consultation and direction in the area of procedures and the linking of information needs with other non-technical aspects of an administrative department.

The lack of veteran staff in Admissions made the evaluation process much more difficult due to a lack of knowledge of procedures prior to fall of 1991. This was the basic reason why the administrative computing staff placed Admissions and Recruiting as a major priority. With the turnover in the Data Coordinator position within Admissions, it became necessary for establishment of formal procedures where little had existed and any that did exist were basically unknown. We soon discovered a major delay between the time that information from students was received and the time that the information was entered onto the computer system. This resulted in very inaccurate information being supplied to upper level administration. For this reason, the administration was not forewarned about the terrible recruiting year, until it was to late to fix the problems.

The Admissions department at this point was a six year user of Datatel's Colleague software, recording all information on Prospects and/or Applicants files. The software provided a good means of recording information, but there was little development in reporting, in spite of an easy-to-use query language. Procedures were not well documented, so that was the first major step in getting the Admissions department back on its feet.

In developing a strategy for rebuilding our recruiting effort and making if more effective, we discovered many requirements;

- we needed to provide the administration with timely and accurate reporting of the status of recruiting throughout the year
- we needed to improve all publications and better utilize existing materials
- we needed to develop a means of controlling what communications would
- we must have a system eliminating duplicate correspondence and missing correspondence
- we want to mass produce written communications but want individualized customization

- we need to improve the print quality of all written correspondence and utilize the collected data on our database in creation of letters, labels, and envelopes
- all data on the Colleague database must be able to be accessed quickly by all Admissions counselors
- an effective telemarketing program must be developed to improve verbal contacts the prospect pool must be increased in spite of any declining trends
- data must be maintained and accessible for multiple recruiting years for c comparison purposes
- all information must be integrated and based on the Colleague foundation already established
- software must provide methods of categorizing all prospective students by their seriousness about attending our institution
- we need effective handling of rejections, special acceptance programs, withdrawals, and students no longer interested in Loras

What our investigation found was that all aspects of recruiting fit into two categories, either verbal or written communication and the proper managing of each.

The Verbal Communication Solution

We decided that the response-time available with the Prime hardware currently installed was not able to meet the requirements for quick access for telemarketing purposes. To improve this performance, it would be a major investment to acquire additional hardware at this time. We installed a token-ring PCnetwork utilizing Novell Netware 3.11 and software from NoelLevitz called Dialog. The server is an IBM PS/2 model 60 with nine PS/2 model 30 computers. The nodes are located in each office within the Admissions area. What Dialog provides is fast retrieval of information and categorization of prospects for telemarketing purposes. All comments associated with telephone contacts with prospective students can be organized and recorded on the Dialog files.

Dialog only provides the tools to keep efficient records of all verbal correspondence, it does not provide you with all of the information necessary to diagnose your best methods of recruiting. It is still up to the institution to determine what the typical student at Loras will be like. Dialog software, however will enable the telemarketers to tag or "grade" prospects so the right amount of time and money can be applied to the right prospects. The primary goal is to build a large prospect pool and then proceed systematically in eliminating prospects from that pool that show little or no interest in attending Loras College. The primary purpose of any telemarketing program must be to introduce your institution to eligible prospects that ordinarily may not every hear about your college.

The Dialog system allows us to highly individualize our recruiting telephone calls and remind us when each student needs to be contacted again. Prospect records on the network can be given special flags to indicate when information on your database needs changing or when the prospect indicates that they are not interested in Loras. The system allows the creation of standard call dialogues that key on various student comments or interests. The script can change as the telephone conversation changes and a record of items discussed can be recorded for reference during follow-up phone calls. Other features of the Dialog system are:

- multiple color coded windows to display information while a caller is on the telephone
- history of all telephone calls made by a caller and all calls made to a particular student
- step-by-step record of a student's progress through the admissions process
- reports for daily activity and prospects contacted by almost any criteria
- establishment of recruiting strategies through pattern reporting and data analysis

While the Dialog system provides everything necessary for a complete Admissions database, it is still essential for our institution to maintain our Colleague database for integration with all other modules of the administration of the institution. For this reason, all pertinent information on the Colleague Admissions files is downloaded to the Dialog network. This download provides all information that has changed on our Admissions database on a weekly basis. We simply download information as an ASCII file onto an IBM PS/2. From this download, we use Noel/Levitz custom conversion programs to upload into the network server in the format required by the Dialog software.

The Written Communication Solution

The Dialog system fulfills most of the requirements for all verbal communications, but we felt that we needed a system to control and record all incoming and outgoing written correspondence between our institution and it's prospective students. There are many products on the market, but we had already acquired the Correspondence Control module from Datatel's Colleague. The module was provided to us about three years ago, but was never implemented because it was not understood by Admissions staff, nor was it thought that it was required for successful recruitment at that time.

Like Dialog, Datatel's Correspondence Control provides only the tools necessary for assigning correspondence, processing of correspondence, and the recording of all history of correspondence activity. These are the primary functions in any letter flow system, but the true view of the capabilities doesn't appear until you design the overall flow of correspondence and link it with a word processing package. As we found out very quickly, the design phase can easily be the longest and most complex task in implementing letter flow within the institution.

We started by determining what we really wanted to do to improve the communication with our prospective students. We felt very strongly that you could not improve recruiting substantially by simply adding more students to the prospect pool and continue with the same communication methods that had been used for the last five to ten years.

There were some definite goals that became obvious to us in our preliminary planning. These were:

- send no correspondence unless it is useful
- don't send any correspondence if the timing is inappropriate
- the appearance of the letters must be professional and consistent
- appropriate personnel need to have letters signed and not use signature stamps
- only use gummed labels when absolutely necessary

- limit the size of letters (typically a single page)
- eliminate duplicate mailings and enclosures
- keep correspondence from continuing when students are rejected or withdrawn
- allow separate types of correspondence for special acceptance programs (i.e., learning disability, one semester review)
- individualized correspondence for a more personal image (includes insertion of custom text)
- maintain history of all incoming and outgoing communication
- no additional Admissions staff
- unique correspondence flow for different levels of prospects (i.e., valedictorians, ACT score greater than 24, Nation Merit Scholars, etc.)

We began the design process in the Summer of 1992. It took approximately 4 months for us to design, discuss, and agree on a good structure for our letter flow. All this time was not spent only on the design. We started testing the versatility of the package very early in the process by running large groups of letters to ease the pressure in our word processing center. This allowed us to test the software, basically in live production.

The planning staff included the Admissions Director, the Head Basketball Coach (assigned in off-season to lead the project), and the Director of Information Services. The project coordinator, in consultation with the other two members of the team, drew charts of the communications that were needed at each level in the recruiting of a prospective student. Until these charts were complete (no loose ends), the actual implementation could not proceed. One of the most critical factors in making the system work, is that the Admissions staff must know what they want and the design must be workable within the framework of the software available. But as we have found out, the complexity of the design has not been limited by the software.

Since the software provides the basic tools for a lett·r flow system, each institution will have a totally different design for their letter flow, and in fact, we expect our design to change extensively every new recruiting year.

Of the written documents that are being designed into sequences of communication that make up the correspondence charts, many have existed for the last year or more. Most have, however, been completely rewritten. These letters had, until the Winter of 1992, been produced entirely by our Word Processing center on their IBM System/36 with Displaywrite software. The names, addresses, and other information utilized in the correspondence had been typed manually for each letter produced. In addition, there was absolutely no way for anyone to know what letters had been sent to which students or any idea about which students had not received any correspondence at all. Duplicate correspondence was another likely result of this lack of correspondence history.

As the design stage progressed, existing and new documents were charted into sequences with the timing between letters recorded on the chart. New documents were created, entered into the word processor, and created in Correspondence Control, and assigned to one or more correspondence sequences called tracks. This process can take a considerable amount of time, depending on the number of documents used. For letters that are only different by a few words or a few phrases, it will need to be decided whether you want to create individual documents for each version of the letter or if only one letter will be created and the differing text will be inserted each time the letter is produced. For each document, a number of items must be decided:

- what is the name of the document (limited to only eight characters)
- what is the description of the document
- will history be recorded when the document is produced
- who will author and sign the letter
- will duplicates of the letter be allowed for a single prospect
- are there enclosures with the document
- is this correspondence a printed document or just a reminder
- will the correspondence be produced on a specific day of the week or daily
- will a laser envelope be created with the document or will a gummed label be produced
- do file copies of the document need to be created
- will the document be dated

Once the charts depicting the actual flow of documents are completed. It will be relatively easy to transform the charts into many tracks of one or more documents. A prospective student can be on any number of tracks at the same time. Tracks of correspondence can contain one or more documents to be sent at designated times. The timing is specified either by the number of days from the assignment of the prospect on the track to delay or the exact date of printing. We started with a fairly complex plan that consisted of charts for inquiries, limbo (reminders until all information is received), applicants (first inquiry and nonfirst inquiry), and confirmed students.

As sections of the design were approved, the correspondence was requested from the Word Processing Center until it could be implemented on our system. We were, however, using the information in the Admissions database to monitor production of various documents, insuring that no prospect would be omitted by accident. Even without adding any new correspondence, the volume of letters began to overwhelm the word processing staff. It became obvious that if written communication was computerized, the volume of correspondence would increase drastically. This situation actually became the driving force behind getting Correspondence Control implemented sooner than we had projected. It was very clear that only the new Correspondence Control module could possibly provide the resources that was needed for the Admissions documents.

We had planned to have the entire Correspondence Control module in operation by January 1, but by mid-November a serious backlog developed in producing letters in the Word Processing Center. Many of the letters were not able to be produced in the time frame necessary for the letter to be appropriate or effective. As we started producing more and more letters through Correspondence Control, we also prepared the module for a "live" date of December 1.

The primary problem with this arrangement, as we found out later, is that we did not prepare adequately to handle the inquiries, applicants, and confirmed students prior to the December 1st date. This led to the majority of problems in dealing with the new module. It was necessary to determine by database inquiry and the stage of development of a recruit which letters needed to be sent and when. After a couple of months and records changing from one track to another as they progress through the admissions process, this manual procedure was no longer necessary.

As tracks of documents are assigned to prospective student records, a record for each document is placed into a file that serves as a queue, holding all information necessary for each document to be printed correctly. These records remain in this file until the batches of documents processed are deleted. These records can be used to print documents directly if letters are lost or misprinted. Documents with similar printing characteristics are grouped into batches, processed, and then deleted. Usually all letters with envelopes are printed together and all letters with labels are printed as a group. History records are updated either automatically or manually during the processing stage, whichever is desired. A separate file is maintained for each prospect that records every document processed and track assigned for that person.

Batch processing of documents usually is performed daily. Assignment of tracks is done continuously as communication is received from the prospects or decisions about a prospect are made by Admissions staff. This process continues throughout the recruiting year. This software allows the Admissions Office to manage every piece of mail that arrives or departs.

Operational Solutions

Software, hardware, and procedures are all necessary components of a successful Admissions recruiting solution for any institution. While these parts are extremely important, the most important ingredient is a skillful, energetic, and dedicated staff. Each person involved in the recruiting of students plays a critical role in the overall program.

The current regular admissions staff consists of:

Director of Admissions
(6) full-time Admissions Counselors
(2) part-time Admissions Counselors
Data Coordinator
Receptionist/Secretary

The Director of Admissions plays the largest role in the design process and provides very necessary public relations with community, institution, parents, and prospects. The Director is the individual with the most responsibility to design the correct quantity and quality of communication between the institution and it's potential students.

The Admissions Counselors assist the data entry process by supplying the input necessary to record all personal visits and other verbal contacts with the prospective students. It is important for the counselors to understand how to extract information from the Colleague database and how to retrieve information from the Dialog files. Many of the letters in Correspondence Control will have custom text entered and be signed by the counselors. One counselor remains on campus to coordinate the Dialog and Correspondence Control systems. Three leading administrators have been assigned to be special counselors in the three high schools of Dubuque to concentrate recruiting efforts on our own city's students.

The Data Coordinator provides the leadership necessary to insure expedient entry of data into the Colleague data base and the Correspondence Control module. All reporting is provided by this person in addition to supervising of all student data entry staff.

The Receptionist/Secretary supervises all student workers in correspondence mailing activities, coordinates the schedules of appointments for all Admissions staff, and distributes all telephone calls to the appropriate persons. This person also provides some backup data entry and corrections to name and address information on returned mail.

With the counselors out of the office a great amount of time and the massive number of tasks necessary to provide support for all new recruiting endeavors, it is necessary for us to employ many student workers to fill the gaps. With proper supervision, our students have been able to provide some of the most important assistance in both technical and public relations areas. Some of the student positions are described below.

Telemarketer -- make appropriate calls to prospects; record important information on Dialog
Ambassador -- specially selected students to guide campus tours, coordinate student/parent visits, provide leadership to prospective students
Coder -- code on input source documents to aid in accurate data entry
Data Entry Operator -- entry of coded and non-coded information onto Prospects file and letter flow system
Data Proofer -- checks coding and entry of data for accuracy, proof letters that are produced for accurate names, addresses, and custom text
Output Operator -- batching of pending documents, processing and printing documents, envelopes, and labels on laser printers, and sorting documents to match with labels and envelopes
Mail Preparer -- prepares finished output from the letter flow system so it can be mailed (sorting, stuffing, distributing)

Obviously, without reliable student employees, our operations would never be done in a timely manner. This, I feel is the key to the success of such an endeavor. Besides teaching the students some valuable skills, the primary by-product is loyal workers that can most accurately assess the values of the institution and demonstrate their feelings to other prospective students and parents. Besides this, the students get paid, allowing many to financially be able to remain in school.

Statistical Reporting

With the Correspondence Control module and the Dialog system being relatively new, there are not very many factors that can be analyzed at this point. I expect that at the end of this recruiting year, we will begin the task of developing some standardized reporting on these methods of contact management. We are still dedicated to concentrating on fully analyzing the data that is gathered into the Colleague Admissions module.

With the Colleague system, prior to the last recruiting year, there was very little statistical analysis performed, beyond simple hand written reports. The query language provided more than enough capabilities to provide this information, but the staff didn't feel it was necessary. So, as part of the rebuilding process in Admissions, we developed some reports emphasizing comparisons between the current recruiting year and one or more previous years. Some of these reports are:

Weekly Progress Report -- compares the weekly progress of inquiries, applicants, acceptances, and confirmations for freshmen and transfers to the same week the previous recruiting year

Area Analysis Report -- provides statistics for each geographical area assigned to an Admissions counselor for inquiries, applicants, acceptances, and confirmations with percentages of each category in comparison to all areas for the last two recruiting years

State Analysis Report -- provides counts of inquiries, applicants, acceptances, and confirmations by each state for the last two recruiting years

High School Target Report -- selected high schools that have typically been good recruiting sources are listed with either inquiries, applicants, acceptances, or confirmations for the last two recruiting years

Campus Visit Report -- displays the number of prospects that have made personal visits to our campus and counts those that are still active inquiries, have applied, are accepted, and have confirmed

High School Rank Distribution -- tabulates freshmen, transfers, all new students, full acceptances, learning disability students, and one semester review students for each level of high school rank (i.e., top 10%, top 20%, top 50% etc.)

New Student Performance Report -- divides all new students into freshmen and transfers, provides the number and average ACT score, HS GPA, and high school rank for full acceptances, learning disability, semester review, and all types

Summary of Contacts Report -- gives totals for the last two recruiting years for 40 different types of contacts with prospective students with the categories of inquiries, applicants, and confirmations

Transfer Analysis -- for inquiries, applicants, and confirmations for the last 4 years, the totals for each month of each recruiting year

Many statistical reports are still in development using a four year history of trends to predict confirmations.

The Future of Admissions Recruiting

Probably the most important change for the future is the much needed close relationship between Admissions and the Financial Planning Office. This was foreseen over a year ago, so these two offices were placed together physically in one wing of our administration building. This tie will strengthen greatly in the next few years. As work has developed in the recruiting area, timing problems between these offices has clearly surfaced. For this reason, the computer staff is emphasizing the financial aid area as a high priority in the next fiscal year, while still continuing with our work in the Admissions area.

In Dialog, I really don't expect much new development except in statistical reporting and trend analysis areas.

With the Correspondence module, we are already planning on developing a more categorized approach. Initial tracks of correspondence will be assigned according to the prospect's performance in high school, with each category having a unique series of correspondence. New and revised letters will continue to enhance the effectiveness of the communications and more and more custom text inserts will personalize the letters to give the appearance of being created individually. The operations involved with this software will increasingly be accomplished by student workers, with supervision of regular Admissions staff.

I feel that we have in place, all the software and hardware that we need to provide a solution for every information need that Admissions will require for the next few years. It is important for the massive amount of data being collected to be transformed through statistical analysis and general reporting into meaningful knowledge for the use of the recruiting specialists and administration of our institution. We have created a very solid foundation but there is still much refinement that continues to be needed as we try to become better at the everchanging techniques of college recruiting.

Experiences of Introductory Network Use in the Class Setting: Development and implementation

Donald Armel
School of Technology
Eastern Illinois University
Charleston IL 61920
217-581-2216
cfdja@eiu.edu

Constantine Roussos
Computer Science Department
Lynchburg College
Lynchburg VA 24501
804-522-8395
roussos@acavax.lynchburg.edu

A year ago, just after the spring '92 semester, my colleague (Ray Richardson) and myself (Don Armel) were discussing new activities for our freshman technology course. Our class covers past, present and future aspects of technology and I generally present the communication technologies. While discussing new activities to teach the present technologies of communication we wanted to incorporate the use of the University's network into the course. So I came to the '92 ASCUE conference with a mission, to find another teacher interested in allowing my students to contact theirs. Through contacts made last summer our Technological Systems class at Eastern Illinois University began to network with the Computer Systems class at Lynchburg College, taught by Constantine Roussos.

This is a synopsis of the experiences, organization and activities associated with the networking of the two classes. Hopefully, this will inspire others to teach the network as content and use the network as process.

The technology class integrates the network as a visual media and activity to augment the instruction of communication technology. Computers and networks are not the main subjects of the technology course, but the computer science class would naturally be concerned with involving the student more fully with a broad range of computer experiences and topics. The computer science class covers instruction on word processing, concepts of operating systems and commands, file and directory management, logging into other computers, and nonDOS operating system commands (UNIX). The instruction of computer activities for network are similar for both classes: email, ftp, telnet, bulletin boards, gopher, archie, usenet, and knowing where to find additional network information.

Actively engaging the students in the use of the network satisfies several goals. First, students are introduced the world of electronic communication. Second, computer use is promoted, leading to computer competency and literacy. Third, students are introduced to the electronic educational services available.

Most of the students in both classes are novice computer users which may lead one to assume the instruction is similar, however the manner and depth of achieving the goals are different. The two classes have different orientations and reflect those differences through the basic content of each course and in the instruction of the network activities. The collaboration between the two school is done to the mutual benefit of each other, despite the differences. The collaboration provides opportunities for enriched network use not otherwise possible.

Class Organization

At the beginning of the semester we want a few things in place. So before the semester begins a list of generic user-id's is requested, and the class newsgroup is reset. The technology class has 40 to 50 students each semester, so to prevent each student from going to Academic Computing and requesting accounts, we are given a consecutively numbered list of user-id's. The user-id's are assigned in class on the day network instruction is given.

The newsgroup prepared for the technology class is moderated to limit article submission to only the class members. The system administrator is notified of the new user-id's and faculty id's to reset the newsgroup for a new semester. Resetting also includes purging the previous semester's articles so the class starts with what appears to be a new newsgroup.

As the semester begins information is shared between the two schools concerning the kind of activities we have planned for the respective classes and the coordination of assignment deadlines. University calendar dates are also shared to avoid scheduling conflicts due to breaks or campus activities.

Activities

Electronic mail is a concept of computer networks that most students are familiar with at least by name. and therefore students interest are already primed to learn to use email. Students are asked to demonstrate basic email proficiency by mailing directly to faculty. forwarding students messages from the other school and sending or copying a reply to a message.

Students use the class newsgroup to discuss group projects. ask general questions and conduct discussions (argue) about topics related to the class. The assignment is to post to the newsgroup a minimum of three times.

Students are introduced to the kinds of information and resources available through bulletin boards. Their assignment is to review several legitimate informational articles and email the review to faculty.

The network is used for every aspect in managing and grading the students' assignments. The newsgroup is frequently check and student activity is recorded. Review articles are checked and accepted or returned with comments for corrections. A check off list is kept for completion of email communication. In case there is a question by any student of their work all student communication are kept for the duration of the semester.

Handout of userid's to students

by Userid			by Last		
first	last	userid	first	last	userid
Shane	Hague	cik7477	Keith	Anderson	cik7504
Pete	Homan	cik7478	Robert	Bailie	cik7493
Matt	Karls	cik7479	Bill	Bertram	cik7487
Rosa	Perez	cik7480	Bret	Boglino	cik7495
C.J.	Boon	cik7481	Del	Bohnenstiehl	cik7517
Jason	Boulanger	cik7482	C.J.	Boon	cik7481
Lonnie	Wright	cik7483	Jason	Boulanger	cik7482
Kate	Maloney	cik7484	Marshall	Canfield	cik7490
Nick	Kournetos	cik7485	John	Creamer	cik7511
Jason	Lee	cik7486	Charles	Croy	cik7502
Bill	Bertram	cik7487	Dan	David	cik7514
Ryan	Gardner	cik7488	Alex	Delulio	cik7497
Brian	Kealy	cik7489	Brian	DiCrescenzo	cik7507

Sample of record keeping sheet

Spring 1993	E-mail					TRN articles				TRN posts					
Keith Anderson															
Rob Bailie															
Bill Bertram															
Bret Boglino															
Del Bohnenstiehl															
CJ Boon															
Jason Boulanger															
Marshall Canfield															
John Creamer															
Chuck Croy															

Total Quality Management in Colleges and Universities

Thomas B. Auer
College of Industry and Technology
Murray State University
1 Murray Street
Murray KY 42071-3307
502-762-3391

Total Quality Management has been an integral part of the industrial scene for several years by now. The commitment to excellence of products and service has become an absolute necessity for those firms who compete in the free enterprise system at home and abroad. We Americans, as individuals, are accustomed to quality. We demand it in our automobiles, appliances and other consumer goods. In the health care industry we will accept nothing less than the best in life-saving measures for our families and we are willing to pay dearly for this essential service.

The price of providing quality has escalated rapidly in the last ten years. Whereas only the United States had the capability to produce the level and volume of quality expected in the early postwar years, now other countries and alliances of nations are driving world class completion for the dollar and the dedication of the consumer. We are no longer alone to determine convenient standards and profitable margins.

W. Edwards Deming was the first guru of quality control to effect a major change in the attitude of industry. It was not his recognition in America but the awakening of Japanese industry that adapted Dr. Deming's philosophy on excellence achieved by teamwork and continuous improvement. We have seen the results in the electronic and automotive fields.

American industry has since responded to the advice of consultants like Joseph Juran and Philip Crosby as well as Deming. But the resulting movement has far exceeded the application of mathematics and statistics pioneered by Eugene Grant and Walter Shewhart. It has given new life to the classic mid-century management guidance of Peter Drucker and Chester Barnard. The insight of these two authors long ago challenged the Taylor Model of authoritative "Top Down" management by showing that high productivity and top quality can only be consistently provided by motivated employees. Both emphasized the major differences between "efficiency" and "effectiveness" — more simply "doing the right thing" versus "doing things right." Both argued that the customer will be right in the long-run by voting with his feet and his wallet.

When "Total Quality Management" is translated into doable tasks it means asking "who is the customer?"; "what does he need?"; "what is the level of satisfaction that will lead to a long-term relationship?" Then TQM means solving immediate problems and building longer term plans with the involvement of the entire workforce and not simply "upper management." Permanent quality improvement and now long-term survival is dependent upon teamwork and the creation of an atmosphere of innovation.

Dr. Robert Cornesky is one of the leaders in the application of Total Quality Management to higher education. His writings show the parallel in thinking between our institutions of higher learning and our manufacturing and service industries. Dr. Cornesky's case studies and conclusions have led to strong recommendations with regard to customer identification and problem solving in higher education. Further, his contemporary work is leading the thinking of those who must assure the survival of our universities in these crucial economic circumstances.

Other practitioners are experiencing very positive results in this area. Today we will hear of two cases of the successful application of TQM to higher education. Patsy G. Lewellyn, University of South Carolina-Aiken, and Bill Wilson, Gettysburg College, will each tell of their personal involvement in this important new movement.

High Tech Student Workstation in the Biomedical Sciences

Thomas G. Basler
Nancy McKeehan
Nancy Smith
Library
W. Curtis Wise
Physiology
Medical University of South Carolina
171 Ashley Avenue, Charleston SC 29425-3001
803-792-2374
basler@musc.bitnet

Abstract

A shift in medical practice and teaching that emphasizes evidence-based medicine creates a need to teach this approach to information gathering to medical students. Hightech opportunities allow for the gathering of information from a variety of sources to a single microcomputer that, through navigational front ends, can access full-text references, databases, illustrations, interactive educational tools and networks to other information and individuals. This allows a latitude to incorporate study tools and decision making aids as a integral part of basic and continuing education in the biomedical sciences.

Introduction

The practice of medicine, and all of health sciences, is increasingly reliant on hightech diagnostic tools and information management. With that trend in mind it is not unexpected to find additional electronic and computer-based tools developing in the area of medical education. The Medical University of South Carolina is exploring many fronts in the advancement of computer and network use toward this end. Examples of this activity include access to full-text references, databases, networks to other information and individuals, medical educational and decision aiding tools.

The High Tech Student Workstation is the access and production tool in this developing environment. It contains the typical array of personal production software applications for word processing, file management, data creation and analysis, and graphics, and the vital information access navigational tools that allow the user to connect with departmental, hospital, campus, and worldwide resources.

The physical location of the information resources has become almost irrelevant to their access and transparent to the student physician user. Nonetheless, considering this from the operational/management point of view, this Workstation/microcomputer will usually have certain resources "directly" mounted on its own hard disks. Those resources that are especially large and either notably interactive or contain large graphic files that typify computer based educational resources are candidates for implementation directly on the workstation.

Other important local resources such as the library system, the hospital and patient information systems, electronic communication with colleagues and faculty, and campus directories and schedules are as immediately available as the resources that are physically located on the Workstation's hard drives. So, too, are the informational resources beyond the campus, including other colleges and universities, other institutional or agency computers, super computers, bulletin boards, electronic mail and file transfer worldwide through use of networks including the Internet.

The information resources accessed might fall into several rough categories such as 1) those resources that are information/data banks, 2) those resources that teach and 3) those resources that aid decision making. These are the same kinds of resources that all health professionals will continue to access throughout their careers. Because of this we believe that these resources should be made widely available to the student population.

We hope to present, not only a demonstration of a wide variety of resources easily accessible from the High Tech Student Workstation, but demonstrate the interactive opportunity lying within this arrangement. Further, we will demonstrate in detail one of the educational, interactive programs valuable as a teaching/learning tool as well as a clinical practice aid.

Medical Decision Making

The availability of diagnostic tools is affecting the way clinicians work. Decision-support systems like Iliad or DxPlain cover a variety of subspecialties of medicine based on actual patient records recognizing most diseases while offering interface with patient records. Comprehensive and clinical collections of electronic reference books are also accessible. Resources being developed in this area include one common search engine to multiple sources including drug information, medical diagnosis, medical dictionaries and diagnostic manuals.

Types of resources presently available at the Medical University of South Carolina, or expected shortly include:

- electronic mail
- library catalogs
- drugs databases
- decision aiding & diagnostic tools
- networks to other library's catalogs
- electronic reference materials
- file transfer
- bibliographic databases
- student aids and resources
- bulletin boards
- electronic textbooks
- electronic journals

Typical also of a health sciences university taking advantage of the recent advances in microcomputers and telecommunications, informational resources such as large databases and systems in the area of drugs and toxicology, medical environmental hazards assessment, epidemiologic studies and patient/hospital information systems are or will be available shortly.

Simulation of Patient Diagnoses and Treatment

Diagnostic Tools. The availability of clinical diagnostic tools is beginning to affect the way clinicians will work throughout their careers. MUSC trained health professionals must become familiar with this approach to clinical support and should begin doing so as students. Even though some diagnostic tools have been in existence for awhile, they have become more sophisticated and can be more efficiently delivered at this time.

One of the mainstays of the clinical simulation efforts appropriate for the Student Work Station is a HyperCard program titled Diagnostic Reasoning™ or DxR™. Developed at Southern Illinois University, School of Medicine by Hurley Myer, Kevin Dorsey and Eldon Benz, DxRsÆ purpose is to facilitate teaching, learning and evaluating the key skills related to medical clinical problem solving. The evaluation portion, discussed below, is very different from the usual expectations of CAI, even those recently developed.

Like earlier attempts of the last decade, DxR provides a simulated patient for the student, via the computer. Unlike earlier efforts; however, DxR records the student's patient handling process and then analyzes his strategy for review by both student and faculty. This gives opportunity for faculty insight into a student's diagnostic reasoning process and clinical problem solving approach and ability. With proper handling constructive feedback is obtained by the watchful instructor.

Generally the DxR evaluation is based on characteristics which have been identified in the literature as skills fundamental to successful problem solving:

High on this list of characteristics are:
- correct framing of the problem
- quickly derived hypotheses
- selection of key findings
- scanning for background information
- searching specific information
- efficient inquiry
- correct data interpretation
- derivation of additional hypotheses following new information
- elimination of competing hypotheses
- verification of correct hypotheses
- accurate diagnosis showing understanding of knowledge gained
- prescription of correct therapy for the patient.

One of the strengths of DxR is the ability to record each of the above characteristics for evaluation while aiding the student in learning how to think about diagnoses, and the faculty to evaluate them.

To further describe the value of this program let us look at two aspects of DxR: Gathering Patient Information and Using the Information Gathered. The Gathering Patient Information section very early on prompts the student physician to state his hypothesis or hypotheses concerning the evaluation of the patient. The program prompts the student to revise, change and add additional hypotheses as the inquiry progresses, thus influencing the student toward correct diagnostic habits.

The Gathering Patient Information aspect of DxR allows the student to learn or review the commonly accepted line of questioning for patients in general and for any case in particular. Such minimal information as name, age, race, gender and simplified statement of the problem is stated at the introduction of each case. A photograph of the patient meant to simulate the patient's physical contact is presented on the screen along with HyperCard buttons including: Interview, Exam, Lab, Hypothesis, Diagnosis, Treatment, Learning Resources, Notes, Consultant, and Interpretation. Initial patient interviews contain 250 questions that can be asked, grouped into seven broad categories: Present Illness, Associated Symptom, Lifestyles, Medical History, Psychosocial, Review of Systems and General.

The Exam button leads to a palette of doctor's Tools allowing for over 375 different physical exam procedures. Hypotheses can be modified as information is gathered during the exam. Another example of the extensiveness of this program is the opportunity available through the Lab button. The Lab allows for the 32 most commonly ordered laboratory tests, by highlighting them and thus bringing them to the student's attention, but it further allows for over 400 other tests that can be ordered upon request. Results for each test can each be compared to typical or normal results, upon request. Notes can be made by the student at any time for further study or as learning issues to discuss with faculty.

Typical of the newer generation of CAI, the tests and procedures are presented, not only in words in an easily obtained way, but as graphics, sounds and other simulations of the real thing. Electronic resources are also available. Simulated (individually for each case created) Heart Lab information, ECGonDisk and others are available.

This underscores another strength of this type of teaching/learning tool: the ability to allow for inquiries into information such as physical exams, lab tests, EKGs and other "real" information concerning any given patient. Each patient is real in the sense that he/she was constructed by the creating faculty. However, true to the concept of teaching diagnostic reasoning, no information is extractable from the information base without the student "connecting" it to one of the hypotheses being investigated. Much like most newly developed CAI programs, especially those designed for use in the Macintosh environment, pull down menus and multiple screens make the navigation of the program intuitive and generally easy to accomplish so that the student may place his interests and energies toward his diagnosis not learning how to use the computer.

Another feature of DxR is the capability of linking each case to additional, external information resources which can enhance the student's learning experience. The Learning Resources button can offer access to on-line medical texts for general clinical information such as description of symptoms, discussions of possible diagnoses, or therapeutic options. Differential diagnosis knowledge bases like Iliad, DxPlain, or Reconsider which accept symptoms as data and assist the student or clinician in establishing a diagnosis can also be called upon to improve efficiency and suggest or eliminate hypotheses. A quick search of the campus library for appropriate printed resources in the book or journal literature can also be performed.

Within the concept of Using the Information Gathered, progress can be measured in several ways. Since DxR is measuring the process as much as the outcome, efficiency in approach is encouraged. It is obvious that in real life this efficiency affects physician time, patient cost and correct treatment. Responses to the student who is not quite on track include: "You may ask more questions, but if you do, you cannot reach the highest performance levels." Areas within the program allow for "Notes" and "Issues" cards to be created and displayed for/to the student. A Consultant portion of DxR will give expert opinion and feedback on a question.

The final step of the computer interactive process is the Diagnosis. At this point a "dialog box" appears indicating to "Select the item from your diagnostic hypotheses list that most closely matches your desired final diagnosis. It need not be complete." A typical answer might look like: "Medications, Neuro Problems, Cardiac Problem, Atrial Fibrillation, PSVT." After entering this information another dialog box requests that the student "Edit and add to this information to finalize your diagnosis. Your goal is to create a complete but concise pathophysiologic diagnosis." Here the answer "Episodes of lightheadedness are related to the rapid heart beat associated with the PSVT. The PSVT is likely the result of the mitral valve prolapse," is a reasonable response. Thus, ends the diagnosis.

Patient Management is also included within DxR allowing for various treatments including medication, physical location (inpatient, outpatient), physical activity, diet, nursing orders, consultation with specialists, patient education and special procedures.

Once again, one of the most innovative aspects of DxR is the Performance Feedback. Here the computer councils both the student and the faculty. Issues such as "was there enough data for diagnosis?" "was this the correct diagnosis?", "were the Lab tests ordered the correct tests?", and other questions can be considered. Further, the faculty/designer of any patient case can follow the entire class of students through the same case in order to find areas constructed too simply, or those that seem too obscure. Or more to the point, on which parts of the diagnostic reasoning process do most students excel and on which parts do they tend to be weakest? For the inventive faculty a set of Case Construction Tools are available for building their own cases to incorporate into DxR.

Although DxR is an excellent example of the newly developed interactive programs designed to facilitate teaching, learning and evaluating the key skills related to medical clinical problem solving, it is only one of the first, that surely will be a precursor to many others, each becoming more sophisticated and effective as microcomputing technology advances.

Interactive digitized educational tools.

Although the following computer based tool doesn't feature the diagnostic values of DxR and other such programs, it brings to the Student Workstation a wide variety of educational opportunities.

Animated Dissection of Anatomy for Medicine™ (ADAM) . This microcomputer based application provides a study tool which allows a complete view of human anatomy. Further, it contains interactive patient education tools. Considered a step above the student's constant companion Grey's Anatomy, a standard tool since the early part of this Century, ADAM goes far beyond the once typical "page turning" CAI process so often tried and rejected in the 1970's and 1980's.

Study without the need for actual cadavers makes this program excellent for purposes of review, continuing education and perhaps even the original study of anatomy, since ADAM is fully interactive, multimedia, medical computer software which features on-screen dissection of the human body. There are those, including former Surgeon General C. Everet Koop, who believe that ADAM and other tools of this type may well prove to have almost as much impact on medical education as the original creation for which it is named.

Ten years in the making, a team of medical illustrators generally educated at the Medical College of Georgia's School of Medical Illustration have produced highly-accurate illustrations, blended with state-of-the-art computer interactivity allowing dissection via the computer or exploration of histology and radiology.

Its illustrations are clear and simulate multilevel dissection, thus it would be easy to see ADAM as a simplistic tool for viewing. However, some modest attempts at a very user friendly/instinctive interface reveals much to be seen. A student can simulate a real-life incision to reveal anatomically-precise details. By the touch of a button he can move layer by layer from the outer skin revealing the muscles, ligaments, blood vessels, nerves, bones and other tissues, to the center of the body. The user can peel the layers individually, or can navigate immediately to the chosen layer. There are choices of views: anterior, posterior, lateral and medial (the typical medical student cross-section). Each will address all layers as mentioned above. Gender may be changed in the ADAM image, once again with the press of a button.

A built in identification function, bringing labels to the screen, points to thousands of parts of the human body. Once again, the student touches the item (down to pixel level) to be identified and a dialog box brings the identification label to the screen. At that point another option can be explored. Since ADAM incorporates the studies of histology, pathology and all other anatomically oriented procedures, the user can request the proper histologic or pathologic representation of the item previously identified. Through the histology module, the user is allowed to compare the histology of different tissues. The Radiologic images are also excellent representations of the clinical images which each student will study during his/her clinical years. All movement to other levels of anatomy or to specific visual representations occurs either instantly or within seconds.

The Surgery module allows the student to choose a scalpel to simulate surgical incisions and dissections anywhere into the body, enabling the student to explore layer by layer. This supports the development of a better understanding of the relationship of one part of the body to another. An example of a newly developed and released module is the ObGyn Library, thus expanding ADAM's capabilities in a specialty area.

ADAM also includes a self-testing module. Our observations are that students enjoy it and look forward to using it. Unlike most educational tools, but somewhat typical of the most recent computer-based education, the implementation of ADAM at the MUSC has been student driven and as more and more faculty have discovered it, it has become college-level driven.

Since ADAM contains an excellent Author Module, this gives the professor the power to create a personal archive of anatomical images. The application was specifically designed so that faculty can easily author their own programs in order to demonstrate new procedures or incorporate existing slides and videotapes into the ADAM program. This opportunity has promoted the concept of in-house development of ADAM "add-ons" by medical colleges nationwide. An ADAM authors meeting in June, 1993 provided an opportunity for these educators

This first meeting in Charleston was to test the waters for an expected annual ADAM Conference, held in the same location each fall sponsored by ADAM, Apple Computer and MUSC.

Another of ADAM's features is motion illustrations or digitized video through the use of the newly developed QuickTime™ application. This allows for a full color, animated surgical procedure that demonstrates in a simple, constructive manner, the steps necessary to correct any hypothetical problem. Any illustrative materials can be incorporated.

The system can also be used by practicing physicians to demonstrate procedures to their patients, giving them a new sense of understanding about their own case. A physician or assistant can demonstrate in a very short period of time such procedures as a hip replacement, by first showing the general area of the body (hip) and moving down through the layers to the area affected. Then without changing screens, the physician can call for a digitized video that slowly shows the general problem, the procedure and the final result. Parts of this presentation can be incorporated automatically into the patient's record (including a patient's release form) but just as important, color printouts can be taken home with the patient for further reference. As a preventive measure ADAM can also be used by the physician to simulate the progression of an untreated pathology or to exhibit how a specific type can worsen, causing unnecessary damage.

Thus, ADAM is presently serving as a commercially available anatomy educational tool, an authoring system, and an aid to clinical practice. Other capabilities such as "hooks" to bibliographic and other databases will make this application a mainstay within the concept of the High Tech Student Workstation.

Conclusion

The examples of educational and informational tools available through MUSC's High Tech Student Workstation are excellent examples of what we believe is the future promise of computer-based learning. This type of materials used to be considered in the category of interesting "techie stuff". However, now we realize that these programs are precursors of what will become mainstream medical education. The future practice of medicine will rely heavily on computers and we as biomedical educators should adjust our educational strategies sooner rather than later, not only to accommodate, but to embrace this exciting development.

Thomas G. Basler, Ph.D., Professor and Director of Libraries and Learning Centers;
Nancy McKeehan, M.S., Associate Professor and Assistant Director of Libraries for Systems;
Nancy Smith, M.S., Assistant Professor and Head, Systems Operation and Instruction;
and W. Curtis Wise, Ph.D., Professor of Physiology and Academic & Research Computer Specialist.

Medical University of South Carolina, 171 Ashley Avenue,
Charleston, South Carolina 29425 3001

Unique Communication Opportunities for Nursing Students and Faculty

Kimberly D. Bone
Tamara L. Simpson
College of Nursing
University of South Carolina
Columbia SC 29208
803-777-3756
kimbone@univscvm.csd.scarolina.edu

The College of Nursing at The University of South Carolina has implemented a new approach to communication between Nursing students and faculty. Students access the College's two networks either from on campus via computers in the Information Resource Center or from off campus via portable computers checked out by the College. Topics covered are how communication was in the past, how it is now, the obstacles encountered in the planning, development, and implementation stages and future expectations.

A three-year plan was developed for the Information Resource Center. In this plan, goals were established to increase student usage of the lab resources, develop efficient electronic communication between faculty and students, transparently extend the lab hours while physically reducing them, increase student computer skills and knowledge because the Nursing Licensure Exam would be computerized by 1994, increase faculty computer skills and design an area that is conducive to student learning. It was planned to hire student assistants on the federal work study program versus graduate assistants who required additional wages for a tuition reduction.

One process in place before 1989 was the plan for a total renovation of the third floor which encompassed the student computer lab, offices, and three other learning labs. This project was approved at the State level as a Capital Improvement Project in 1987. This was a prime opportunity to design an area conducive to student learning. Modular sound proofing units were designed for each work station with disappearing and adjustable keyboard shelves. Each station was designed with privacy panels. Power poles were purchased to hide unsightly cabling for the network. Installing a false flooring was not an option, so the network topology was limited to that which could be free standing with limited cable exposure.

The Information Resource Center (IRC) at the College of Nursing before 1990 had ten stand alone IBM XT computers with 20 megabyte hard drives and 5 1/4" disk drives. The computers were placed on computer tables, each containing a dot matrix printer, with little or no privacy. Because of this equipment and these conditions student usage of the IRC was minimal. Three IBM convertible computers were also made available for check out to graduate students. These convertibles did not have hard drives and swapping diskettes was common. The use of these computers was also minimal.

Course requirements for students in the Information Resource Center were the completion of various computer-aided instruction (CAI) programs related to particular course content. Again, students were continually having to swap diskettes to run these programs because of the limited disk space of the XT computers in the center. The only resident program on the hard drives of the computers was the word processing program. WordPerfect 5.0 which was selected by college administrators as the supported word processor of the College of Nursing.

Although the IRC usage was low in the semesters preceding Fall 1991, students could access the lab seventy-seven hours during a week, which included Saturday and Sunday hours. These operating hours required the hiring of many student assistants and graduate students. One reason for the low usage of the IRC stemmed from the limited computer skill level of the college faculty. Their skill levels increased with the installation of the administrative network. The College of Nursing's administrators were dedicated to the efforts of the network installation and used electronic mail as their primary method of communication, replacing the inter-college hard copy memorandum. The faculty in turn were forced to become "computer literate." A series of instructional classes were held for faculty in the Information Resource Center to orient them to the new system. Individual consultation was also made available.

Due to the number of grants supported in the College of Nursing and a commitment by it's administration, we were able to purchase sixteen new IBM PS/2 Model 30 286 computers with 40 megabyte hard drives connected on a Local Area Network (LAN) for the lab. Money was also allotted to purchase all of the network hardware and software necessary for operation.

It was decided that the student lab network topology would be Ethernet using ThinNet cabling. The cards purchased were 3C503 which come with 3COM's lifetime warranty. A new file server was purchased for the administrative network. In turn, the administrative network's replaced file server was added to the new lab network. Netware 386 3.11 was purchased for both networks. Also, Word Perfect Office 3.1 was purchased as the menuing system and mail system for both networks and WordPerfect 5.0 was upgraded to 5.1.

Over six hundred nursing students were given individual accounts on the lab network. Each student was allotted 5000K of hard disk space for their class related or personal materials. Within a year, six of the new 286 computers were upgraded using Kingston Technologies to 386sx machines with four megabytes of RAM. Seventeen Zenith portable computers and four notebook computers with hard drives and 2400 baud rate internal modems were purchased. Fifteen Zenith portables and two notebooks were assigned for student checkout. Two Zenith portables and two notebook computers were assigned to department secretaries for faculty use.

A new computer to serve as an Asynchronous Communications Server was purchased. This server contained three data lines to accept outside calls. Ninety-six hundred Baud Rate Modems were connected to the three data lines of this server for faster processing for students and/or faculty with computers at home containing modems that operate at 9600 Baud Rate or faster.

Dot matrix printing is not an option. As a test, four dot matrix printers were installed in the Fall of 1992. Printing to these printers was fee. The two recycle bins were emptied at least twice daily. Students were abusing this privilege. As a result, dot matrix printing was eliminated. Everything printed in the IRC computer lab is routed to a Hewlett Packard LaserJet III. Students are required to pay ten cents per page for printing. Although students were required to pay for printing, the lab usage has remained steady and consistent with that of figures from the Fall 1992 semester.

Since the XT days, the lab usage has increased tremendously with an average of eighty student users per day. This can be attributed to better equipment and comfortable facilities after the renovation. Faculty's willingness to learn and to encourage student use of the facilities is another factor in the increased usage. Faculty also have increased assignments that require the use of the computers. Another encouraging aspect for students is the development of an American Psychological Association (APA) format macro. This macro prompts students to simply type in information and the macro automatically creates appropriate APA formatting in WordPerfect for their papers.

The physical lab hours have been reduced, but have been transparently extended due to the installation of the Asynchronous Communications Server allowing remote connections through personal home computers or portable computers checked out through the lab. This enables commuting and mature students to call into the network at any time eliminating the need for them to be physically in the lab.

Clinical faculty can also use the departmental portable computer modems to connect to area hospitals. This eliminated the requirement of having to go to the hospital and log in to their system to make clinical assignments. They simply used modems to connect and make assignments remotely. To facilitate the communications between the administrative network and the lab network, a bridge was installed. This process only required one 3COM card to be installed in the administrative file server (Token Ring) with the appropriate cable connection. This bridge afforded communication between the two different network topologies (Token Ring and Ethernet). After this bridge installation, faculty making their clinical assignments, transmitted them via email to the respective students in their user account mail boxes.

Group orientations for network training were scheduled for nursing students. The orientations encompassed network training as well as WordPerfect 5.1 instruction. Individual orientations were also set up to accommodate student schedules.

Hiring and training student assistants for the lab took a tremendous amount of time. The requirements were previous computer and audio/visual equipment experience. Fortunately, some very competent students with those qualifications were hired. Other positions were filled with undergraduate students without work study funding.

Communications between students and faculty before 1990 were limited to class time and/or faculty members office hours. As a result, students had limited opportunities to ask questions and receive input from faculty before submitting assignments. Consequently, most feedback from faculty was received once a paper, careplan, or project was submitted and returned for a grade.

Another form of communication between faculty and students was the College of Nursing Suggestion Box, located in the canteen area of the school. Suggestions were compiled by the Office of Academic and Student Services and physically given to the Dean of the College.

Presently, communication between student and faculty via electronic mail can be conducted twenty-four hours a day, seven days per week providing immediate feedback on issues, questions, assignments, etc... Once feedback is received on an assignment, students can quickly retrieve their work and make modifications.

Upon the local area network installation in the IRC, the faculty's new computer knowledge allowed them the means to communicate with students. Because of this means of communication, careplans, one key clinical assignment for students in the college could now be transmitted via e-mail. Careplans are used in all clinical settings and as assessment forms specific to course content and patient populations. They are used with nursing diagnoses approved by the North American Nursing Diagnosis Association (NANDA). Before 1990, students purchased careplan packets made available at a local print shop. These packets contained all the sample nursing careplan forms. Although students only needed a few of the forms, they purchased the entire packet which cost approximately $30.00. Students were encouraged to duplicate these forms, in order to preserve the originals for later patient/client situations.

Costs to the students did not end after purchasing the packet. Students selected particular forms in the packet and completed them by hand or transferred hand written notes to the form by typewriter. Once completed, the careplan was submitted to the assigning faculty member by hand, either in class or at the faculty member's office. The faculty in turn compiled all the careplan assignments and graded them by hand. This long and tedious process for both faculty and students absorbed valuable time that could be spent in interaction between faculty and students.

It was determined to use the clinical courses that required the completion of careplans as a prime opportunity to establish efficient communication between faculty and students. Six faculty, two staff, and two administrators were strategically picked to participate in what was labelled a "Pilot Project." The faculty chosen had to be constructive with their criticisms, not destructive. These faculty also had to have good computer skills and one to two who had no skills. The search was successful and wonderful group was organized. The initial meeting was a sub sandwich luncheon to introduce the participants to the idea and receive feedback. Nine of the ten participants were receptive to the project. The faculty member with doubts agreed to the project with the stipulation that students decide whether or not to use the method the group established. This participant was one with limited to no computer skills. After a decision was made to move forward with this group and this strategy, individual orientations were developed. The members of this group had computer accounts on the administrative network. The next step was for these faculty to have easy access to the student network. A student network mailbox option was placed on each participant's menu to send and receive student messages and/or assignments.

Class by class orientations were conducted where mock careplans were used and mailed to their participating instructor. The instructor would then reply to the student's careplan with feedback and suggestions on completing the mock form. The student then read the instructor's comments and immediately incorporated them into the careplan. In order to encourage student use, one instructor offered extra credit to students who sent her messages via email.

The careplans used by nursing students are available on the network for immediate access. The forms were adjusted so that movement was conducted by pressing the TAB key in WordPerfect 5.1 Tables. Access to the careplans is conducted through the use of a checked out portable or a home personal computers. These computerized forms can now be neatly completed on the computer and electronically mailed to the assigning faculty member. Faculty can now view and grade from home via communication server and departmental portable. One drawback to the computerized form is that the document can not be seen in its entirety on the computer screen. Students must be familiar with the form before attempting to complete it. Also, home computer users cannot use their mouse in the remote session. Students and faculty's learning curves were shortened by offering a variety of orientations and individual consultations.

Feedback from the participants at the end of the semester was provided during a pizza luncheon. Some suggestions were to stagger careplan assignments throughout the semester. The hesitant participant stated she would require students to complete at least one careplan on the computer in the future. Overall, the group had a positive experience opening the path for other classes and faculty to be added to this project. Other clinical courses and respective faculty will be added in the Fall of 1994. The majority of students participating in this project found completing careplans on the computer and having direct e-mail access to their professors extremely helpful.

Students enrolled in Nursing classes, but not admitted to the Nursing program did not receive accounts because the listing from the College of Nursing Office of Academic and Student Services did not include them. These accounts had to be set up on an individual basis. The estimated number of occurrences were approximately 100 to 150. The Nursing 110 course had five sections with approximately 100 students in each section. This course was the main contributor to having to establish individual accounts after the initial semester setup.

Students also had trouble remembering their user-ids and passwords. The IRC student assistants were trained on setting up student user-ids and passwords.

One complaint about the portable computer LCD screens was that they were smaller and more difficult to read. However, if the student wished, an external color monitor was connected to the portable for better vision.

A software concern existed with some nursing computer assisted instruction. The programs in question were not network compatible; therefore, multiple users were not allowed access simultaneously. However, these particular programs can reside on the file server and be used by one remote user at a time.

Some future expectations are to train nursing students to use the campus backbone system to communicate with other students on campus and subsequently other campuses, connect to the Thomas Cooper Library through the backbone, upgrade hardware for increased memory, upgrade software in order to assure network compatibility, increase data lines for the communications server from three to six, develop two multimedia stations and continue to purchase state of the art equipment for nursing student resources.

The World of AGE, a Telecommunications Project

Dr. Marvin Bratt, Director
The University Classroom of Tomorrow
The Ohio State University
1465 Mt Vernon Avenue, Marion OH 433025695
614-389-2361
hbratt@magnus.acs.ohiostate.edu

The University Classroom of Tomorrow (UCOT) is a teacher education project designed to assist upcoming teachers meet the challenges of technology. Funded by grants from the Ohio Board of Regents and The Ohio State University, UCOT has been in existence for four years. We have developed numerous activities for students to become involved with technology. Each of the UCOT students works with us for approximately four years. Kathy Rutkowski, editor, NetTEACH NEWS writes...

> "Teachers like their students need time to "play" with the technology. It is a new technology and most of the teachers out there teaching have not been exposed to this technology as college students. The younger teachers that are now graduating will be a different mindset and will come into the schools asking for the most advanced work stations and internet connectivity. However, most of our current teaching population has had limited exposure."

Learning is an intense experience. Those who have studied it find that there is a period in which we must "explore (vis play)." This exploration leads to more disciplined investigations and with appropriate encouragement and direction, becomes learning. This is a different role for teachers.

Each year for the past four years, UCOT has participated in cooperative projects with schools on the AppleLink network known as Apple Global Education. This year's project was designed to collect unique pieces of information about the participating sites. We were interested in gathering information that is not readily available in texts but was commonly known by the population surrounding a particular site. As Dave Allan at Marigold Elementary School in Victoria, B. C. Canada, director of the project put it, "We are building a new folklore... or doing our own living archaeology. Without ruling out well known information, we are looking for those things that are known to the people of the place."

The University Classroom of Tomorrow is similar to many other projects in technology based education. What makes it different is that it is directed at future teachers. I joined Apple Global education with the intention of introducing my students to experiences that they could not get any other way. This has become a reality and, indeed, gone much further. While we participate in several networks in order to do our telecommunications projects, AppleLink is the only one that allows us seamless transmission of text, graphics, digitized sound and video files among sites. This has been the case up until the introduction of Eudora and Gopher servers to the Internet. The internet still seems too foreboding to most teachers. They simply don't have the time nor the stamina to overcome the learning curve. In our experience, it has taken at least three quarters or an academic year of constant work and exposure to overcome the anxiety that besets most teacher education candidates. The process seems even more protracted with graduate students who don't have full time to devote to technology learning. This is exactly how the introduction of a project such as "The World of AGE" plays into the process. It provides a substrate, a vehicle, for students to work in and learn the skills of computing.

What I had reported only last year was that my students were using the computer to develop papers for their coursework. This was a justifiable goal for our project in the beginning, I thought, because it broke down the barrier between the student and the computer. We knew that there was more to using the technology than word processing. Ben Shneiderman of the Department of Computer Science, HumanComputer Interaction Laboratory & Institute for Systems Research at the University of Maryland wrote "I have learned many lessons about technology and education in the AT&T Teaching Theater, but my most compelling experience is the acceleration of my transformation as a teacher ... I see more clearly than before that the path to motivating students is the joy of creation, exploration, and discovery. I see also that these processes are social in nature and that shared experiences in class and through teamwork projects are vital." This is precisely the types of experiences we attempt to develop in AGE projects. Again, projects such as those we participate in on AGE are the cornerstones that cause us to probe technology deeper for more applications, more tools. Each of us on AGE could have participated in a project such as Dave Allen has proposed without computers. The computer provides quick efficient feedback that traditional mail services and telephone conversations do not.

We began by building a Hypercard stack of the authors pictures and short biographies. This is not a trivial exercise. Deciding what to tell about yourself in a small space is a challenge. Choosing a good digitized image (of yourself) is also interesting. Recording your voice is something new for many of us. This is real-time learning and education. It is demanding, requires sound decision making skills and requires careful thought. Having completed the task of collecting photos, voice recordings and biographical information, we next entered it into a hypercard stack.

Next, we found photographs of places of interest in our local area. This also created a good deal of discussion. What did Dave mean by this. How were the places to be chosen. We had to decide to make our own rules. At OSU Marion, we have students from several surrounding communities and so each student picked a place or places that was familiar to them. Next a short narrative describing each photo was written to accompany the photo. We were able to add quick time movies to some parts of the stack. Again, decision making was critical to choosing the material. This information was then sent to Mr. Allan and the students at Marigold to be incorporated into a large hypercard stack.

We have visions of expanding the stack to include high resolution images of the locations and participants as well as longer descriptions of the places they submit. We now are of the opinion that the information we have gathered can be an ongoing history of our research project. As new students come into the project, we will simply add their information to the stack.

In sum, the participants have learned several new technological skills. We all know how to collect and edit sound recordings with the audio hypercard stack and SoundEdit software. We know how to capture each other in quicktime movies. We have learned to edit the digitized information with Premiere software. We have developed some knowledge of hypercard and how to incorporate quicktime movies into it. We all collected new information on familiar places to share with everyone.

What we have found in our three years of experience in doing this sort of project is a renewed commitment to and a completely different vision of teaching and learning and different types of commitments from teachers and students. We have come to understand that education is not about books and tests; but about relationships. Both teachers and students develop a new and much more comprehensive vision of the use of technology after participation in such a project.

Jump Start the Use of Computers in the Classroom

Kim S. Breighner
Computing Services
Gettysburg College
Gettysburg PA 17325
717-337-6930
kbreighn@gettysburg.edu

Abstract

At Gettysburg College, we have successfully completed a training program for all first-year students. To train them in the basics of the computer, word processing, electronic mail, and network tools, we added a computer component to an already existing required core course, firstyear Colloquy. Sessions were offered on both Macintosh and IBM compatibles. This presentation discusses the planning and implementation of this course and the direction we are taking for computer training for the future.

Paper

The FirstYear Colloquy program was introduced into the curriculum of Gettysburg College during the 19831984 academic year. It was developed because of the faculty's desire to establish a common educational experience for all students. Its goal is to assist students to develop intellectual skills and habits (e.g., critical thinking, sound writing) which are basic to a liberal education, but until this year it did not include computing skills.

Over the past two years, Gettysburg College has invested a substantial amount of time and money networking the campus. This state-of-the-art technology provides many benefits for the College. Without the proper training, these resources probably would be underutilized. We decided to help the students take advantage of this technology by providing them with basic, comprehensive computer training. We designed a series of hands-on workshops for all firstyear students on computer basics, wordprocessing, electronic mail, and network resources.

There were approximately 18 sections of Colloquy with approximately 15 students in each section. The Provost required all adjunct professors to participate in the computer component of the project and urged all other professors to participate also. Nine members of the Computing Services department were involved in the training aspect of the project. Each member had anywhere from one to three sections to train. We held three mandatory sessions an hour and a half long and then an optional fourth session at the end if needed or desired. As soon as the sections were divided up and trainers were assigned, it was the responsibility of the trainer to contact the Colloquy professor to tell them exactly what all we would cover in the sessions. We felt very strongly about working as closely as possible with the professor so that our assignments meshed with what they were doing in class. We wanted the students to feel that providing this computer training would make their computer use more efficient and less frustrating and not have them feel that lab time was "wasted." The students were required by their professors to produce a paper in the first few weeks of class, so including wordprocessing in the training enabled them to handle this assignment with greater ease.

Training in electronic mail allowed them to communicate with their Colloquy professor for assignments, class syllabi, notes, or any other information they were concerned about or had questions about. It also gave them the capability of communicating as a whole by setting up an alias for the entire class. And training in network resources provided students the ability to access College information from Gopher, our electronic campus information system, and reach users and access information on other networks.

In the wordprocessing portion of the program, students are taught the basics needed to enable them to produce acceptable printouts for course assignments using such functions as page numbering, justification, formatting features like bolding, italicizing, underlining, and changing font types and sizes, footnotes and/or endnotes, editing text, moving text, spell checking, previewing their printout before printing, choosing the appropriate printer, saving their document after initializing their disk, and finally printing their document.

In the electronic mail portion of the program, students are taught how to send messages, receive messages, set up their own signature file, look up mail addresses of other people on campus, delete messages, send and receive attachments such as course assignments to and from their professors, and how to use the aliases that are developed for campus-wide use.

In the network resources portion of the program, students are taught how to use Gettysburg's Gopher system. In this system, they can access various information such as the local campus library catalog and other library catalogs at other institutions on the internet, campus information such as the campus calendar of social and academic activities, student schedules, all documentation supplied by Computing Services, the schedule of additional free-of-charge training sessions offered by Computing Services, news information from various sources such as USA Today and the Washington Post, the information they would need to join any of the various interest lists that are available to them on the internet, and any other information supplied by any department on campus. If any Colloquy professor (or any other professor on campus) requests that their course syllabi, assignments, or notes be posted on Gopher, this can also be done and the students can be trained to access all course information from this network tool. They also access gopher systems at other institutions most times as quickly as they can access information on our own campus and are shown advantages of having this capability. If students are interested, they can be instructed on various other network tools such as FTP, which stands for File Transfer Protocol, and would enable them to retrieve files from other institutions on the internet.

Upon completing the first successful year of this program, it is quite clear from student evaluations that the information and skills taught are very vital to the their four years at Gettysburg and beyond. But it is also quite clear that it is a great undertaking for the members of Computing Services that are involved in the training. A lot of time was involved in planning the courses, making sure all the members that were involved were familiar with all of the software involved, meeting with the Colloquy professors, and then finally conducting the actual sessions. A committee of four members of the Computing Services staff -- all four of which were major trainers in the program for the first year spent a lot of time in the preparation of this project. I was a member of this committee. This project, as any other, had it's bumps along the way. We had to work with the Registrar's office and the Colloquy Steering Committee from the very beginning. One of the biggest concerns was how to make sure that students were signed up for training sessions on the type of machine they wanted to use. From information on a form we were able to include in a mailing the Registrar's Office sent out over the summer, we were able to determine what type of machine they preferred (IBM or Mac). From this information, the Registrar's Office divided the students up into different sections according to this preference. Of course, there were a few students that either changed their preference between the time they filled the form out or for some other reason changed their preference, so there are always a few cases where a student needs to change training sections. There were also a few professors that did not want to get involved in the training, which caused some minor problems. Another problem arose because we started the sessions the very first week of classes. Some of the professors were not available until classes actually started, so their was a communication problem there. As a result, we are in the process of investigating new and different ways of getting this information to the students rather than continuing to conduct regular classroom sessions. A few ideas that are being discussed are to conduct the training sessions over the local campus TV station and to develop a video library of the sessions for checkout purposes so that the students could go at their own pace in their own networked dorm room using their own computer.

We see tremendous benefits of this program. Students will be more computer literate, which will enable them to do bigger and better things with computing resources while at Gettysburg and in their future. It will also mean that the faculty will be able to integrate more new and different computing applications into their classroom instruction so that both students and faculty will have more and more new tools at their disposal and be able to use them in a variety of ways. It also helps Computing Services, hopefully, in that there will be fewer basic questions. Because of these benefits, we feel our time investment is well worth it.

Spreadsheet vs General Ledger Accounting Program in the Accounting Curriculum

Clarence Coleman
Charles Doryland
School of Business
Francis Marion University
P.O. Box 1000547, Florence SC 29501-0547
803-661-1413

Randy Reid
Central Carolina Technical College
Sumter SC

Introduction

Basic Financial Accounting is one of the most challenging courses facing business students. It is also a challenge to the professor to communicate the nature of the accounting equation which is fundamental to understanding future accounting issues. Too often, the teaching of basic financial accounting is assigned to junior accounting faculty or graduate assistants. This assignment unwittingly diminishes the importance of the course in the eyes of students and sometimes fellow faculty. What does an accounting professor attempts to accomplish in the first course of accounting? Most authors begin with the fundamental accounting equation Assets = Equities. The equities portion of the equation is further divided into those belonging to creditors and those of the owners. During the early chapters, in the financial accounting class, students increase and decrease the various balance sheet accounts by using +'s and -'s. At this stage in their development, a "+" means to add and a "-" implies a reduction. Students generally understand this one to one correspondence and tend to do well. Students have a tendency to get behind in the transition from "+'s" and "-'s" to debits and credits. A debit does not always mean increase nor does a credit connotes decrease. The appropriate action (increase or decrease) is a function of the normal balance of the account in question. Most students are already familiar with the concept of normal balance (plus and minus numbers) from their algebra course. But very seldom do text book instructions or instructors draw upon this prior learning experience. The purpose of this article is to show how a spreadsheet can be used to link this prior learning experience to understanding basic financial accounting. Moreover the broad concept of Asset = Equities can be reinforced through graphic visualization. Transitory assignments are those that are intended to foster a student's learning in a collateral or advanced course. Some author refer to transitory assignments as "repetition". Transitory assignments are important in enhancing a student's self confidence in upper level courses. This paper will demonstrate two transitory assignment examples and report feed back from some senior level students.

SPREADSHEET ADVANTAGES

From a Teaching prospective, there are four identifiable benefits gained from using spreadsheets. The Spreadsheet display offers the students an instant pictorial display of the effect that transactions. The spreadsheet discloses immediate results as opposed to the delayed effect of posting. The spreadsheet provides visual conceptualization of accounting numbers. Because of the speed of the spreadsheet, more and diverse transactions can be incorporated in assigned problems. The fourth benefit of the spreadsheet is the algebraic methodology which links the student to prior learning experience.

Immediate Result versus Posting

The spreadsheet display offers the student an instant pictorial display of the effect that transaction entries have on the various accounts. The impact of entries show up immediately as changes in all the affected account balances. Students begin to grasp the different types of transaction as outlined by most authors in their basic accounting texts. The types of transactions are outlined below:

- 1 Transactions that increase both sides of the equation
- 2 Transactions that decrease both sides of the equation
- 3 Transactions that change the composition of Asset or Equity

Through careful pre-entry thinking, students can anticipate the affect of various transactions on certain account balances. The actual result are observed instantly and either confirms or corrects the student's understanding of the transaction effect. The spreadsheet is a natural tool for many of our students who have grown accustomed to visual learning via the great America past time, Television.

Graphic Conceptualization

Though it may surprise some accountants, the profession has long been known for its tabular orientation. Accountants tend to prefer data presented in tables as opposed to graphically orientated data. Accounting principles text have begun to use more graphs to emphasize various concepts. While static graphs represent a step forward in presenting what is often difficult subject matters to the students, dynamic graphs linked to changing data are superior teaching tools. Dynamic graphs provide an immediate pictorial representation of changes in asset and equities accounts. The student receives instant visual and tabular feedback. After appropriate instructions, students should be given assignments that incorporate dynamic graphs. These students will be better prepared to sell themselves to employers and communicate with customers.

More and Diverse Transactions per Problem

A third advantage of the spreadsheet lies in its ability to rapidly process transactions. Since students are not burdened with the rudimentary problem associated with calculation errors, they can concentrate on understanding the conceptual model. In addition, the elimination of calculation time allows the instructor to give more transaction per problem. A far wider variety of transactions can be addressed than could occur by working the problems manually under existing class time constraints. The variety of transactions and repetition fosters the students's learning process and make more efficient use of class time.

Algebraic Correlation

Most accounting students have had a course in algebra in high school or college prior to enrolling in Principles of Accounting. Unfortunately most accounting text ignore this prior learning experience which might enhance the student's understanding of the accounting equation. For example, students could be told that all assets normally have plus balance and all equity (liabilities and owner's contributions) accounts have minus balances. A student with an algebra background immediately knows how to increase or decrease these accounts. From a spreadsheet layout prospective, the rows represent transactions where the net of the pluses and minuses must equal zero. The columns connote accounts where the plus and minuses represents increases and decreases in the accounts with some resulting account balance shown at the bottom of each column. By skillfully using panes, tiles and graphs (Appendix) the instructor can provide the kind of emphasis on the basic accounting equation not found in most accounting texts.

Disadvantages

The disadvantages of the spreadsheet are often the advantages of the general ledger program. However, they should not be viewed as mutually exclusive tools, rather each tool has its place and purpose in the accounting curriculum. The ability to date, reference and describe transactions are not nearly as effective in a spreadsheet as in a general ledger program. Spreadsheet documentation tend to be more cryptic and customized to the individual user. The spreadsheet lacks the T account structure that permits side by side viewing of increases and decreases in accounts. The above short comings of the spreadsheet should not disqualify its use in the early part of an accounting principles course where understanding the basic accounting equation is paramount to a student's success in the course.

INTEGRATED GENERAL LEDGER PROGRAM

An integrated general ledger program is one in which the activity of the special journals (cash receipt, payments, sales) are automatically recorded in the general ledger. the special journals are automatically assigned reference codes for general ledger identification. Although the sum of the activity of the special journals is recorded in the general ledger, the amounts can be modified through changes made in the special journals prior to posting. Most accounting are not inclined to use this general ledger type program because a student using the special journal only need to know one side of an accounting entry. For example, if a student access the cash payment journal he only needs to know the account to debit. The Program automatically sum all debits recorded in this module and credits the cash account for the total. The use of special journals circumvents the reinforcement of the equality of transactions debits and credits. Accountants not familiar with these programs are not aware of the general journal entry mode provided by most of these accounting programs. There are several advantages to using the general journal entry mode following the spreadsheet reinforcement of the basic accounting equation. First, the student must determine whether he is increasing or decreasing the account in question. Second, the equality of debits and credits is stressed because most general ledger programs will not accept an unbalanced transaction. Third, All transactions may be entered on the same screen, hence reducing program learning time. This method of transaction entry is appropriate prior to the class discussion of special journals. After the class has discussed special journals, each transaction can be entered separately through the general journal mode with its own journal reference, date, description, etc. Finally, the professor should permit student to use the special journals (often called modules by software developers) after establishing a small group of customers (accounts receivables) and vendors (accounts payables). The general ledger accounting program exhibits many of the benefits obtained from using the spreadsheet. The student is freed from tedious calculations and thus should have more time to analyze transactions. Posting, which is often time consuming and fraught with student errors, is handled by the program, again providing the student additional time for analytical work. Some accounting programs have begun to provide reports and graphs ordinarily derived from spreadsheets (Appendix). The graphs and columnar reports were produced using an integrated general ledger accounting program. This particular accounting program is capable of graphing the account balance in up to fifteen different accounts. The graph format selection (bar, horizontal bar, pie) rivals that of some spreadsheets. Although the graphs are not dynamic, they provide continual visual learning stimulation in the later part of the introductory class.

Ledger and Report Generator Structure

A detailed discussion of general ledger and report generator structures is beyond the scope of this paper but the selection of the appropriate general ledger account structure is critical to the professor's future course objectives. General ledger account structures fall into two major categories, flat and tiered. A flat general ledger account structure is one which the account balances do not pour over into some category (marketing) or Classification (current assets) accounts. In this type of account structure, the report generator bears the burden of combining accounts to obtain subtotals for some desired account grouping. In some flat general ledger systems, the account numbering sequence determines its sub total grouping for reporting purposes. The accounts in a tiered general ledger structure pour over into specified classification accounts thus significantly eliminate the need for combining accounts to generate financial reports. The tiered account structure is better suited for those professors who would like to demonstrate report building or graphics. In addition, the tiered structure can be readily adapted for cost/managerial problems or ratio analysis.

Transitory Assignments

Transitory assignments prepare students for similar but more intense assignments later on in their course of study. The study of ratio analysis comes rather late in the accounting principles course. Conversely, the topic is introduced early in the basic finance class. Spreadsheet assignments emphasizing ratio analysis are excellent transitional course requirements (Appendix). Students in basic Managerial Accounting are often asked to prepare various schedules (eg. cash collection) and budgets (eg. cash) assuming certain monthly sales level are attained. In a small business development or entrepreneurship course a student is often required to prepare a twelve month cash flow budget. Cash budgeting, with its collateral schedules, is an excellent transitory assignment (Appendix)

SUMMARY

The spreadsheet and integrated general ledger are excellent tools that can enhance the teaching basic of financial or managerial accounting. The spreadsheet should be used in the early chapters of the accounting course to emphasize the conceptual foundation of the accounting equation. Properly used, the spreadsheet can ease the transition from +'s an -'s to debits and credits. The spreadsheet should be followed by an introduction to the general journal entry mode of an integrated accounting program. The use of a general ledger accounting program eliminate clerical errors and the tedious posting process. Students can then spend more of their time analyzing transactions and preparing financial reports. Analyzing transactions and communicating the results of those transactions is the essence of accounting, unfortunately too much of students' time is devoted to processing transactions. Students are often not responsible for their lack of spreadsheet or general ledger application training. Too often, accounting professors use the same teaching methodology that they have employed over the past fifteen to twenty years, usually the chalk board and lectures. Students normally adapted to this mode of teaching. Unfortunately, the quality of students who desire to major in accounting has declined over the years. Accounting professors must modify their teaching methodology for at least three reasons. One, academically capable students are not interested in transaction processing. They are more interested in the analytical and communicating aspect of accounting. Two, most students are easily bored with manual worksheets and practice sets and some of the brighter students will seek what they feel are more analytically challenging or creative majors. Three, accounting professors who do not meaningfully incorporate the use of the microcomputer in their classes have not adequately prepared their student for both the conceptual and applications demands that will be placed upon their students by future employers.

CD-ROM Anxiety: Finding Problems, Searching Solutions

Anne L. Fliotsos
Dacus Library
Winthrop University
Rock Hill, SC 29733
803-323-2131
fliotsos@admin.winthrop.edu

INTRODUCTION

There is no arguing that CD-ROM has arrived in today's institutions of higher education and will continue to be a presence in the near future. While compact disc technology is a boon to the speed and accuracy of searching for information, it can present problems for students, faculty, and staff. With the constant increase in computer technology, all information professionals must be aware of issues dealing with implementation of that technology. Libraries and Academic Computing Centers need to communicate and work together to solve the problems that persist as technology evolves. This paper is written from the perspective of an academic librarian, and focuses on specific anxieties accompanying the implementation of CD-ROM technology.

As of 1991 96% of all research libraries were using CD-ROM technology. Over half of those libraries own more than 20 titles (Tenopir 23). The majority of the CD-ROMs purchased serve as indexes to journals. Some libraries have networked their CD-ROMs to provide access to any disc from any CD terminal while others, like our library at Winthrop University, offer only single workstation access to each CD-ROM index. At some schools, departments may purchase specialized CD products that reside within the department. This is an expensive option without networking resources, and creates a decentralized information retrieval system. The newest, most exciting option in libraries is to eliminate the disc entirely and tapeload the indexes into the on-line catalog. While this is a desirable option, it is also quite expensive to initiate, so the stand alone CD-ROM workstation is still a reality to many of us.

LITERATURE REVIEW

In the 1980's the term" computer anxiety" became a buzz word as the general public found themselves increasingly faced with terminals in place of paper sources. Within the past five years, the terms technostress, technofear, technoanxiety, and computer hassles have evolved as studies attempted to define these anxieties further. There are basically two types of computer stress to consider: stress associated with the computer literate, and stress associated with the computer novice. While these anxieties have uncommon sources, both types of users experience stress when confronted with computer technology.

Technofear, or technoanxiety is the anxiety felt by the computer illiterate. It is often characterized by avoidance, fear, and intimidation. Craig Brod elaborates:

> This anxiety is expressed in many ways:...irritability, headaches, nightmares, resistance to learning about the computer, or outright rejection of the technology. Technoanxiety most commonly afflicts those who feel pressured—by employer, peers, or the general culture—to accept and use computers. (16)

Technostress, or computer hassles are associated more closely with the experienced end user. These users have demands on the system that are not met. Collin Ballance and Sydney Rogers conducted a study of 186 technical college students using Hudiberg's Computer Technology Hassles Scale. Of the ten most frequently cited hassles they found, several could relate to CD-ROM applications, including: "computer system is down, slow computer speed, lack of computer expertise, others not knowing how to use a computer, and keyboard typing errors" (541).

Change in itself can create anxiety for the user. A review of the literature in psychology indicates that humans are resistant to change, even if the change is perceived as positive (Fine). Psychologist Paul Faerstein states, "Arguing about making changes, showing resentment at suggestions, and not quickly learning what is required are manifestations of such anxiety" (14). He also identifies concerns that technological change can bring: confusion with equipment capabilities, information overload, raising skill requirements, organizational and job changes, work load, added responsibilities, and physical hazards (14,15).

CD-ROMS AND STUDENTS

One way to observe student reaction to CD products in the library is through a library instruction program. Each year Writing 101 classes come to Winthrop's library to learn research skills. Our faculty liaison in the English department surveyed the Writing 101 students after their instruction sessions. Overall, the students felt more comfortable using the library and on-line catalog after the instruction. The one area they still felt they needed help with was using the CD-ROM indexes. The unique complications associated with compact disc technology frustrate users on a daily basis.

Vendor differences create havoc for the librarian and student alike. Search strategy and language differences (such as truncation, proximity operators, and Boolean capabilities) make proficiency difficult at best. Unfortunately, libraries rarely restrict themselves to just one CD-ROM vendor, so software differences are always a problem. General search strategies vary greatly for common products like Wilsearch, Dialog OnDisc, and Silverplatter. With so many systems to master, it is no wonder the cry for standardization has been present almost as long as the technology itself.

Long lines at InfoTrac and other CD products indicate that students like and accept the new technology, but librarians who observe these students know that there are problems. To the student who's only computer experience is word processing, on-line catalogs and CD-ROMS do not offer the full-text capabilities he expects. Other patrons try to type in entire phrases and sentences to search a subject. Often novices assume that all of the computers in the library are the same. Once that assumption is made, the student accepts a computer "halo effect" that everything is in the computer; if the student does not find what he is searching for, it must not be available.

Admittedly, many of these mistakes stem from a basic lack of library knowledge. What is a periodical index? What is a periodical? These basic questions lie at the base of understanding how to use today's automated library. Once the index concept is mastered, students need to be taught the pros and cons of computer versus paper searching. For example, students do not take the years of coverage into consideration when they are searching for materials. Often the CD-ROMs only cover the last several years, and paper indexes must be searched as well. Lastly, the student must understand that library holdings are not guaranteed when they use a national index such as Reader's Guide. Vendors like UMI and InfoTrac have partially overcome that problem now with a serials holding field. However, those schools that use the serials holdings list find that loading and updating these lists is a major source of anxiety in itself.

CD-ROMS AND FACULTY

Faculty reaction to CD technology can run the gamut from fanaticism to frustration. Computer literate faculty and students find that stand alone CD indexes are not enough. They want increased search speed, better retrieval methods, networked access, and dial-in access 24 hours a day. Some want to purchase CD-ROM indexes for their departments, forcing librarians to closely examine the library's jurisdiction in such cases. As an example of this, Winthrop's School of Business wanted to pay for a subscription to Compact Disclosure. The library purchased a subscription, but several years later the School of Business cut back their contribution, forcing the library to pay for the majority of the subscription. Often CD-ROM policy problems are trickier than first anticipated.

There are also faculty members who are hesitant to use computers. Some professionals are self-proclaimed incompetents. Bad experiences with simple technology, such as programing a VCR, may prevent them from embracing today's automation (Bird 86). Other professors are afraid to admit their ignorance of new technology. Most rely on graduate assistants or the libraries' on-line searcher to do their computer searching, so they are unaccustomed to automation in the library.

There is another important reason that faculty prefer traditional print sources. The Chronicle of Higher Education reported that technology does not always mesh with the needs of the humanities researcher. Their interview with Douglas Greenberg, vice president of the American Council of Learned Societies, revealed that "...humanists do not use a linear research style." Mr. Greenberg explained:

> When we use a book, we tend to browse through it. We make quick comparisons of text in the front and the middle and the back...The way humanists tend to work is not suited to the current availability of text in electronic form. (A21)

CD-ROMS AND STAFF

As the CD maintenance person in my library, I can confirm that compact disc technology produces a variety of headaches for the reference staff. Dealing with students' anxiety is only one of the added problems with new technology. Library schools offer little if any training in disc maintenance, software installation, hardware assembly, networking issues, vendor licensing agreements, tape loading, and general systems "troubleshooting." Librarians are not trained as computer technicians, but are regularly faced with these kinds of issues and dilemmas. The jargon and hands-on experience must be picked up independently from reading incomprehensible manuals, phoning for customer support, and attending training sessions. Librarian Steven Zink, in his article "Planning for the Perils of CD-ROM" states:

> CD-ROM's uniqueness and warm user reception have often overshadowed an underestimation of the human resources required for its use. Extensive planning might have exposed the ongoing costs for increased staff time, reassignment of public services area space, maintenance, workstation furniture, supplies, and the differences in software access to the same databases from various vendors. (51)

Extra staff time is not only needed for CD training, but also for increased demand in reference service, interlibrary loan, and bibliographic instruction. In one study, reference assistance increased 20 percent after CD-ROM installation (Silver 62).

We are, perhaps, like the students in that we want one simple answer. Unfortunately, we are faced with more choices now than ever before. Purchasing choices are multifaceted, involving vendor customer support, trial options, software ease, hardware requirements, licensing agreements (including LAN and WAN options), and long-term budgeting. Poor planning at the time of purchase means that money must be "borrowed" from another area of the budget to maintain the equipment. As an example, Dacus Library purchased the majority of its CD-ROM workstations within a two year period. The equipment is now all in need of phased replacement, but budget constraints prohibit updating the hardware. Replacements are only ordered when a piece of equipment completely stops working. Since our CDs are not networked, we have to manually switch discs on the remaining workstations until the new component arrives. This situation frustrates both librarians and students to no end. As new technology develops and budgets get tighter, we find ourselves wanting more and affording less. There is a legitimate fear that what we buy today will be obsolete tomorrow.

The number of CD products has exploded in the past few years. The journal On-line reports over 3,500 CD-ROM titles are available on the commercial market, and 52% of those titles came out within the last year (67). The sheer number of products appearing each year makes the job of selecting and maintaining CD-ROM that much more difficult.

Network options and problems are a source of anxiety in themselves. In a 1990 Association of Research Libraries survey, 26 percent of the libraries surveyed had networks and all had CD-ROMs (LaGuardia). Oregon State University had problems with Meridian CD-Net. After an initial period of success they wrote, "Later we encountered hardware difficulties and software incompatibility problems, with resultant increasing downtime...The vendor's response to service assistance was poor, and we essentially became a beta test site for their software" (Scott). Reports like these are frightening to those of us examining network options. Brian Nielson sums up the librarian's attitude well:

> Whereas a decade ago there was widespread optimism among librarians who embraced technological advances, today there is considerable dis-ease. The technology over which we were to be the masters has turned into an avalanche of technological change that threatens to leave many of us in the dust...Were we misled to think that advancing automation would offer us a special place in the 'information society?' (92)

ADDRESSING CD-ROM ANXIETY

Perhaps the most difficult part of dealing with anxiety is finding a means of relief. The first step is realizing that you are not alone in your discomfort. Anxiety is real, and is a response to external stimuli. For the incoming freshman, anxiety can be partially relieved by letting them know they are not alone in their confusion. Instruction sessions must stress that we are there to help them, and that no question is stupid. We must work to foster positive responses to the technology as it replaces traditional paper sources. Hands-on computer time in small groups will help the student take that first step toward automation. Immediate connection to a class assignment helps motivate the student to learn to search the CD-ROMs efficiently.

CD-ROM anxiety in faculty members is perhaps the most difficult problem to address. Every institution has faculty who fight the advent of new technology. Because of their stature, they have more to "lose" by asking for help. Those people with computer anxiety will avoid computers and promote negative attitudes toward automation. However, some faculty may be reached by special focus sessions held only for professors and graduate students. Others may find that by being present in the classroom during their student's instruction sessions they can help themselves as well as their students.

Staff anxiety can also be addressed with hands-on training and demonstrations. Simply reading about products cannot compare with actual search experience. Visiting other academic institutions and talking with peers can help us understand the pitfalls associated with certain vendor products and can also alleviate the feeling of facing the technology alone. Furthermore, we all must try to use CD-ROMs when conducting our own research. This will refresh our memories on the specialized search functions that are infrequently used. Finally, those faced with a great number of CDs may want to consider a "product manager" program, in which each staff member is assigned to one CD-ROM product. Product managers are responsible for the user manuals, help sheets, and general expertise on their assigned system. This discourages the "jack of all trades, master of none" mentality that can easily develop with a large collection of CDs.

The advent of new technology is at once exhilarating and baffling. As information professionals we must recognize psychological barriers to change and work to overcome these barriers. It is wonderful to watch people "discover" the power of computer searching. Being sensitive to user needs and fostering positive attitudes toward new technology can promote CD-ROM use and help establish the computer as a viable gateway to knowledge.

WORKS CITED

Ballance, Collin and Sydney Rogers. "Psychology of Computer Use: XXIV. Computer-Related Stress Among Technical College Students," Psychological Reports 69 (1991): 539-542.

Berry, John. "CD-ROM: The Medium of the Moment," Library Journal 117.2 (1992): 45-47.

Bird, Jane. "Overcoming Technofear," Management Today Feb. 1991: 86-87.

Brod, Craig. Technostress: The Human Cost of the Computer Revolution. Reading, MA: Addison-Wesley, 1984.

"CD-ROM Concerns," Wilson Library Bulletin 66.6 (1992): 78.

Faerstein, Paul H. "Fighting Computer Anxiety," Personnel 63.1 (1986): 12-17.

Fine, Sara. "Human Factors and Human Consequences: Opening Commentary," in Information Technology: Critical Choices for Library Decision Makers, ed. Allen Kent and Thomas J. Galvin (New York: Marcel Dekker, 1982): 209-224.

Fliotsos, Anne. "Anxiety Layering: The Effects of Library and Computer Anxiety on CD-ROM Use," The Southeastern Librarian 42.2 (1992): 47-49.

Griswold, Phillip A. "Difference between Education and Business Majors in Their Attitudes about Computers," AEDS Journal 18 (Spring 1985): 131-138.

"High Tech and High Touch," Datamation 38.19 (1992): 128.

Hudiburg, Richard. "Relationship of Computer Hassels, Somatic Complaints, and Daily Hassels," Psychological Reports 69 (1991): 1119-1122.

Hudiburg, Richard and T. Morris Jones. "Psychology of Computer Use: XXIII. Validating a Measure of Computer-Related Stress," Psychological Reports 69 (1991): 179-182.

Kupersmith, John. "Technostress and the Reference Librarian," Reference Services Review 20.2 (1992): 7-14, 50.

LaGuardia, Cheryl et al. "CD-ROM Networking in ARL Academic Libraries: A Survey," Online/CD-ROM '90, Washington D.C., 6 Nov. 1990.

LaGuardia, Cheryl and Chuck Huber. "Digital Dreams: A CD-ROM User's Wish List," Library Journal 117.2 (1992): 40-43.

Nielson, Brian. "CompuServ Information Manager: A Mediator on a Floppy." Online 15.4 (1991): 92-94.

"The Printout," Online 17.2 (1993): 67.

Scott, Shirley R. et al. "Networking CD-ROMs in a University Library." working paper, 1990.

Sievert, Mary Ellen et al. "Investigating Computer Anxiety in an Academic Library." Information Technology and Libraries 7 (1988): 243-252.

Silver, Howard. "Managing a CD-ROM Installation...A Case Study at Hahnemann University." Online 12.2 (1988): 61-66.

Tenopir, Carol. "Electronic Reference Options: How They Stack Up in Research Libraries." Online 16.2 (1992): 22-28.

Watkins, Beverly T. "Scholars, Librarians, and Technologists Urged to Join in Using Electronic Information." The Chronicle of Higher Education 27 Nov. 1991: A21.

Zink, Stephen D. "Planning for the Perils of CD-ROM." Library Journal 115.2 (1990): 51-55.

"CONNECTing" with Campus Staff

Thomas L. Hallman
Associate Chancellor for External Programs and Administration
University of South Carolina-Aiken
171 University Parkway Aiken, South Carolina 29801
803-648-6851
HALLMAN@UNIVSCVM.CSD.SCAROLINA.EDU
HALLMAN@UNIVSCVM.BITNET

CAMPUS OVERVIEW

The University of South Carolina - Aiken (USCA) is one of three four-year campuses, apart from USC-Columbia, in the nine-campus University of South Carolina System. Slightly more than 3,200 students are enrolled in baccalaureate programs in nineteen areas, with nearly 1,200 graduate students attending classes on the campus offered through USC-Columbia. There are approximately 135 full-time faculty members and academic administrative and nearly 115 non academic employees spread across the institution.

BACKGROUND

Like many institutions in recent years, USCA has faced midyear budget cuts, no pay raises, unfilled vacancies, and a growing sense that there was nothing positive that could be done. For several years the Administrative Planning Retreat, a gathering of campus leaders from all segments of the University, had expressed an interest in implementing some widespread effort to improve morale among our colleagues. With no additional funds, this was a major challenge. The primary outcomes of this interest in years prior to 1992 were "casual Fridays," when everyone was encouraged to wear less formal attire in an attempt to lighten up the interactions, and several campus-wide luncheons after critical events such as registration and commencement.

A NEW IDEA SURFACES

In the summer of 1992, however, information was received and reviewed by a group of administrators on a program developed for the Noel/Levitz National Center for Staff Selection and Development called <u>CONNECTIONS: Practice for Excellence, Path to Success</u>. The program consists of three video tape sessions, interspersed with small group activities, and a text. The program was geared to aiding non-academic employees on college campuses in their efforts to better appreciate their many contributions to the success of the institution. Since it was exactly this group on the USCA campus that had not been satisfactorily served in the morale-boosting efforts to date, the program got a serious review.

In an effort to make an assessment of the potential value for our campus, a number of interested persons reviewed the materials. It was unanimously agreed that the scenarios portrayed in the video segments were appropriate (while not always Oscar-quality acting!), the exercises in the text were stimulating, and that the overall impact of the activity would be positive for the campus. The group made a presentation to the Chancellor and received approval to purchase the package. While it would be inappropriate to discuss the terms of that sale here, suffice it to say that in a difficult budget year this purchase was not a burden, and the cost per person served was nominal.

IMPLEMENTATION

When the full materials arrived, the campus group that had been involved to this point undertook to participate in the full course. This was an effort to familiarize ourselves with the program and to train the facilitators for the sessions to begin. Finally, in the fall of 1992, all non-academic employees and all administrators on the campus were divided up into groups of 1012, with care given to limit the number of persons from the same department in any one group. The participants received a letter from the Chancellor outlining the program and a summary of the potential times they might participate in the three sessions of 150 minutes each. There were groups of sessions scheduled into every day-part, in order to meet as many different schedules as possible. While participation was officially voluntary, each unit head received direct notice from the Chancellor that these meetings were to be treated with the highest possible respect, with nothing other than a major emergency taking precedence over scheduled attendance.

The training program began with fairly typical "icebreakers" and introductory exercises, with which many professionals are very familiar. It was an unanticipated finding that many of our participants had not had these experiences, and they were all the more effective in this setting as a result. The video and text materials quickly moved the groups into areas such as treating students as customers, respecting our colleagues in the workplace, heading off trouble, and resolving problems. It quickly became apparent that this group on our campus held the same values about service and commitment to the institution that some may have thought were limited to more responsible or more highly educated members of the campus community. Their participation in the sessions was, without exception, well above any expectation the planners may have held.

OUTCOMES

As a direct result of the "Connections" sessions, there is a greater unity among our campus staff members now than has been experienced before. In the main, this is measured subjectively, since no campus-wide assessment activities of morale have been undertaken. But, based upon the level and quantity of greetings on the campus, the number of persons who now know each other better than six months ago, the ability of groups to solve problems that may have been considerably more difficult last year, I am convinced that this effort has been worthwhile. We have held follow-up meetings with all participants, meeting in their original groups, to assess their impressions of the program and to seek information on the next steps. The ratings are overwhelmingly positive. The opportunity to speak candidly with peers about problems and to share common solutions, to rehearse new behaviors and to learn new techniques were all valued highly by the participants. Shortcomings were identified in the effort, including: sessions too long in the work day, faculty did not participate, some video presentations too basic, and more role/playing is needed.

The followup sessions were also utilized to generate lists of the activities that would be helpful in the near future to continue the level of enthusiasm that is now present. These lists included item such as dealing with difficult students/supervisors/colleagues, stress management, and communications issues. Several of these have already been addressed in sessions available to the entire campus community. Others from the list are under consideration for program development in the near future.

CONCLUSIONS

The "CONNECTIONS" program has been a great asset to our campus in that it was the focal point for beginning a conversation among some of our most important people. The people who manage the University outside the classroom have a pivotal role in making things go smoothly and in assisting the all-important efforts at student retention. The events of the past year have demonstrated that the level of caring is just as high in those administrative areas as in the academic halls, and the opportunity to provide some intangible acknowledgment of that commitment has been rewarding to all involved. The specific materials used were suitable in our setting, although the 'top-down' commitment to the effort was the most important aspect of this venture. A campus that does not consider some similar outreach program to the non-academic employees in this period of declining resources may be risking considerably more than the apparent savings of doing nothing.

<u>For More Information on the "CONNECTIONS" Program, Contact:</u>

Noel/Levitz Centers, Inc.
Noel/Levitz Office Park
2101 ACT Circle
Iowa City, Iowa 52245
319-337-4700

Technologies Planning and the Small College

David C. Bonner
John Hammer
Jane C. Neale
Kevin A. Synnott.
Russell Sage College
Troy NY 12180
518-270-2328
hammer@albnyvms
dbonner@albnyvms

INTRODUCTION

Small colleges, particularly private institutions, have not always considered long-term, comprehensive, strategic planning as an essential part of their growth and maintenance. By long-term planning we mean plans that reach beyond two years, by comprehensive we mean planning for all institutional constituencies (and being particularly mindful of the dynamics and interconnectedness of planning goals and strategies between the different institutional planning units), and by strategic we mean planning that goes beyond the reactive response to emergencies that often passes for planning; strategic planning is truly forward looking, not something that simply legitimizes the status quo and deals with unanticipated problems.

There are several reasons why many small colleges do not plan appropriately, they include:

1. We are so small we don't need it. Everybody knows each other, everybody knows our deficiencies (they do not), there is no need for a formal planning process.

2. We can't afford it. Planning costs money (true) and its financial return is negligible (false).

3. Our present one or two year budget cycle is an adequate planning effort (It is not; it only addresses the fiscal side of the planning process).

4. Small colleges often assign more than one responsibility to a single person since each particular job may be seen as not demanding a full-time position. That means no one takes a dedicated leadership role in planning.

5. There is a critical mass issue, the pool of faculty is small, hence the number of visionary thinkers and risk takers is also small, perhaps so small that it may end up being a single person who probably is working in isolation and is not supported or encouraged by their colleagues.

6. Faculty tend not to travel to conferences and generally communicate with their off-campus peers to keep up with changes in technology.

In the discussion that follows we intend to show that planning is an essential part of enlightened management of an institution and that the three excuses given above are at best specious, at worst harmful. Our stress is on technologies planning because of its currency and because it is the one area where many mistakes are made. Finally we will present a technology model for the small college setting.

THE ROLE OF PLANNING IN THE MAINTENANCE AND GROWTH OF AN INSTITUTION

Proper planning must be seen as an essential component in the proper operation of an institution since it plays a most important role in that institution's growth. One obvious purpose of good planning is to help managers make good decisions about how the institution should allocate its funds. This is of course particularly true for small private colleges which seem to be always short of what we might refer to as discretionary funds. The basic bills (i. e. salaries, utilities, insurance, maintenance) get paid, but how will the institution profitably and effectively spend any additional money? Strategic long-range planning will answer that question.

Even more important is the fact that proper planning makes us articulate the long-range vision of the institution. A college needs to think about changing demographics, student expectations, changes in the skills students bring with them as freshmen, and to consider new managerial strategies. If no such vision exists, appropriate planning will force the creation of such a vision. The creation and articulation of vision is of particular importance because it insists that we think about the future in a practical and realistic manner. In addition proper long range planning allows for the evaluation of the institution's efficiency and will generate suggesting corrective action for streamlining.

Thinking about an institution's future in this manner generates a whole group of interesting questions about goals and strategies that might not have been brought up or thought about otherwise. This is especially true of technologies planning. An interesting example is a question being asked by faculty once they had been shown a technological blueprint of their institution. The question was: "What do I do with this stuff? How do I use it?". This led immediately into the much larger issues of technology supported pedagogy and curricula. Ultimately we have to face the issue of how does the teaching/learning process become transformed by the newly available technology; this sort of thinking leads naturally to the consideration of a new pedagogic model, and example of which is described below. What are the consequences if not doing proper planning?

1. Uncontrolled technological growth, individuals start adding little pieces to the institutions technological resource base that don't hang together
2. Serious oversights, there will be large and important pieces that will not be addressed by anyone;
3. Duplication of software, equipment and capabilities;
4. No one guarantees the currency of equipment and software that the institution buys;

5. There is no centralized point of responsibility for assessing the institution's technological needs;

6. There are no performance or compatibility standards for anything the institution buys;

7. Planning requires research and examination of what others are doing and of what the current technology is capable of; lack of planning leads to ignorance in all these areas.

The commitment to appropriate planning is only the first step towards maintaining institutional viability and growth. The planning process itself needs to be effective and inclusive, and may confront managers with a tricky balancing act. The delicate balance stems from the competing goals of inclusion and competence. Those who are ignorant or under-informed about the technology clearly will have little effective role in its planning, yet everyone' needs must be taken into account, and all deserve a voice in the articulation of the institution's technological vision. A classic example of too much stress on inclusion and not enough thought given to the technological literacy of the planning community is from an institution that had just purchased equipment for a computer lab based on the Intel 80286 technology. Within a year of the installation of that lab it became clear that the workstations did not have sufficient power to meet user's needs. When the planners where questioned as to why they recommended such anemic computing power, their reply was "That's what everybody wanted". The lesson is this: Appropriate planning is not a popularity contest.

This example has clear implications for the planning process. One possibility for involving all constituencies in planning, and at the same time not letting their technological illiteracy pervert the process, is to perform a needs assessment that asks about desired \ services instead of desired hardware and software. A different strategy might be to let a small cadre of technological literate and sophisticated users develop a basic plan that can then be presented to institutional members at large for their comments and suggestions. However realigning the consciousness of the institutional community at large can be a time consuming and frustrating task. Accusations of elitism and empire building will be very common.

However if the planning process is too inclusive, the result is often little more than the legitimization of the status quo. Particularly those who are not technologically literate tend to be fearful not only of technology but the impression that their skills and abilities are outmoded and no longer adequate. Those who are fearful of technology and its implications will not do a good job of planning for it. In these situations goal setters tend to be too timid, aspiring to reach up one level above their present capability instead of moving up to the current technology. In addition the complex dynamics between the many constituencies tend to be blurred.

Financial constraints, although obviously needing consideration, often enter the planning process too early. Initially one must plan with both an unlimited vision and pocketbook. Once the major plan is roughed out then finances and budgets help determine which of the things being asked for are necessities and which are luxuries. After these cuts the whole project needs to be reevaluated to make sure that what is left hangs together and makes sense. It makes no sense to "unfund" the fiber optic cable if one is trying to establish a campus-wide communications network.

One other *caveat*: Don't let the planning process become too complicated. It will lead to confusion of the planning process with the planning goals. The process needs to be just sophisticated enough to get the job done. it should not become an end unto itself.

A TECHNOLOGY BASED PEDAGOGIC MODEL

New computer based applications and the integration of multimedia techniques will provide for a new way to think about the teaching and learning process generally, and about pedagogy specifically. There is an excellent opportunity for a new pedagogic paradigm to emerge. Although Thomas Kuhn (1970) wrote specifically about natural and physical science, one can make the argument that a similar fundamental change (a paradigmatic shift) is coming to teaching and learning. This change in teaching has two related aspects, delivery and elucidation.

DELIVERY: The blackboard and chalk, the transparency, the slide projector, the film and the video tape are being replaced by delivery methods based on computing technology. The reason we need to replace these delivery methods as the primary delivery technology is that they usually are not interactive, that they are linear in the sense that one goes through information being presented in discrete steps from beginning to end, only seeing one aspect of the phenomenon being observed at any one time, watching passively as information is reeled out in front of us.

This lack of interaction and passivity prevents the kind of involvement that the student, and the whole class, should be having with the material presented. Students get the impression that knowledge is a linearly linked chain of facts to be absorbed one item at a time. Intuition, the deduction of general principles and lawlike statements about the information shown, are seen as being secondary. Faculty of course are aware of this and try to supplement their teaching with discussion, the most obvious and effective form of interaction and involvement available to them.

ELUCIDATION: Traditional explanation is also linear and unidimensional. Problems arise when complex phenomena become involved; there is only so much one can do with text and diagrams on the blackboard or transparency. This makes it very difficult to clearly demonstrate complex, dynamic, time dependent, multidimensional concepts and relationships.

The new pedagogic model is based in the integration of computing technology into teaching and the curriculum. The goals of the model are interactivity and involvement, self exploration and open-ended problem solving, collaborative learning, and visualization of complex relationships, all to be discussed below.

With proper facilities, students will be able to examine primary data electronically and in real time. Instead of listening to a lecture and looking at transparencies, students will be able to directly explore primary data sources, drawing conclusions, and forming hypotheses that can be tested by them directly. Not only will students interact with the data and concepts, but also with each other. Such an arrangement encourages collaborative problem solving, brainstorming, exploration and experimentation, things that the traditional teaching situation makes very difficult or does not allow. The students will in effect be helping to teach themselves, learning from their own efforts, and taking an active role in the learning process. This not only will make their results and conclusions come to life and have a reality that "taught" results lack, it also raises their self esteem as learners. They will have an enhanced sense of accomplishment about their self discovery.

Visualization of complex relationships can be greatly enhanced by computing technology. Imagine the calculus instructor trying to demonstrate the relationship between the volume and surface area of a cylinder as its dimensions change. This is virtually impossible to explain in text or illustrate on the blackboard, yet with the use of computing technology this concept can be easily visualized. Particularly the passage of time, for example in a demographic simulation, is very difficult to perceive with traditional methods, yet becomes very clear when presented using computing technology.

Reducing the tedium of mechanical computation is another important advantage of computing technology. Although students will, of course, have to learn to do their computation initially "by hand," once having demonstrated that they can do it there is no need to have them spend significant amounts of class and lab time doing it over and over. Calculus, linear algebra and differential equations are classic examples in which the burden of algebraic manipulation takes on more importance and more time than the contemplation and elucidation of concepts that are central to these subjects.

Although one is immediately drawn to examples from mathematics and the sciences the potential of computing technology in the teaching of the social sciences and humanities is just as great; complex relationships are found everywhere.

- Imagine a cultural anthropology class being able to directly access the Human Relations Area Files (HRAF), forming and testing hypotheses in real time, working in small groups and brainstorming, experimenting with the data and the questions being asked of it.
- Imagine a class on Elizabethan literature. The instructor might show a section of dialogue from Romeo and Juliet, on a second window a sonnet, and while scrolling through them, showing structural similarities (or differences). How can one possibly do that with board and chalk?
- Imagine a music history class studying baroque vocal music. How does one illustrate the then common practice of alliteration, or having the words mimic the music? In the Handel's *Messiah*, the basses aria "Every valley shall be exalted" contains the words

> "Every valley shall be exalted.
> And every mountain and hill shall be made low..."

in which the melody runs up and down the scale, while with the words

> "Make straight in the desert a highway.."

the melody is held on a single note. Would it not be wonderful to show the words, the score, and listen to the music at the same time, scrolling through the display, highlighting the similarities?

A classic but different non-numeric example is the freedom that word processing has brought to writing. The new technology allows, and in fact encourages, revision and editing, something that traditional methods did not encourage and made very cumbersome indeed.

The relevant concepts of the new pedagogic model include collaborative learning, open ended questions and exploratory learning, real-time manipulation of primary data sources, and an highly interactive classroom environment, the use of technology to illustrate complex relationships.

A TECHNOLOGY BASED INFORMATION RETRIEVAL MODEL

For the library to lie truly at the heart of the academic enterprise, it must support academic excellence by mediating access to the scholarly research needs of both students and faculty. In the current climate of technology-based information retrieval, it is more important now than ever for the library to offer a comprehensive package of information services that transform the library into an information gateway that opens for the user the doors of their library as well as the doors to collections of many other libraries. Such a gateway, which should be accessible across a campus-wide network, (Moberg, 1992) encourages the library user to become the finder of knowledge as well as its user.

This paradigm of mediated information retrieval, in which the librarian facilitates student and faculty acquisition of information retrieval skills, has taken deeper root in the increasingly technology-based library climate (Carpenter & Edmonds, 1992). In the pursuit of knowledge, the student gains the experience necessary to build expertise in locating, retrieving, assessing and utilizing information however and wherever stored. By developing such critical skills and the habits of thought and inquiry with which they go hand in hand, college graduates will be better prepared for their roles as students, as citizens and as professionals in a world increasingly traveled on "information highways".

In keeping with this paradigm, automated library systems must be selected with the user in mind. A single user interface that provides access to library holdings, as well as other local and remote databases, will enhance student success at information retrieval, reduce the technological barrier to research and contribute enormously to the academic experience. Such an environment will also facilitate increased technological literacy for those individuals who are not yet computer literate. The addition of an integrated online public access catalog, with linkages to other databases, both internal and external, will eliminate some existing barriers to technology mediated information retrieval.

A robust research environment also requires the capacity to work and communicate with students and faculty both at the local institution and in the greater global campus. Improved communications via a campus-wide network will provide that important local linkage. As we network within our campuses, we must also look beyond, to the external academic community, via the Internet. The worldwide academic community has adopted the Internet as a primary method of communication, from which the small college should not want to remain apart. The Internet provides researchers, scholars, and students the means of informal information exchange and virtually immediate access to information and databases in a wide variety of formats (Lane & Summerhill, 1993). Internet functions, such as file transfer protocols, discussion groups, and remote log in via Telnet, round out the suite of Internet capabilities (Lane & Summerhill, 1993). As academia moves towards an increasingly electronic mode of communication, higher education is being transformed into a virtual global campus. To provide our students and faculty with the appropriate tools of education, small colleges must become citizens of this virtual campus.

Small college libraries must keep up with these evolving information technologies if our students are to graduate as information and computer literate individuals. To move toward such a climate, a series of computer-based tools and services can be implemented over time. To optimize these additions, it is useful to have a clear picture of what the emerging technology-based library should be.

An Online Public Access Catalog (OPAC) that establishes a climate for library research in keeping with the future of libraries as electronic information gateways should be selected. Such a system would include the OPAC, automated circulation, serials control, acquisitions and cataloging. Each of these modules will improve access to, and the quality of, the library database. This in itself will enhance students' academic experience and improve the quality of research by both faculty and students. In addition, the system should include an interface to locally mounted bibliographic databases, such as those currently available on CDROM., and access to the Internet. Features such as user interfaces, hardware platforms, functionality, technical support, product adaptability and the potential for product growth into the future should be given careful consideration. It is wise to select a catalog that will meet institutional needs both now and in the future and one that will continue to develop in keeping with telecommunications and information exchange protocols and standards, such as the Z39.50 protocols. The integrated OPAC should permit improved procedures for the acquisition and control of institutional library holdings and improve efficiency while providing better and more detailed information to patrons. Administrative features should include acquisitions, circulation, serials management, cataloging, and authority control. The capacity for connectivity to a campus network is extremely important.

A small college library may find a less sophisticated and less costly OPAC to be an attractive solution. We have found that it is better to aim for a more costly long term solution than a less expensive product that may not evolve well into the future. A utilitarian OPAC, while less costly, may have neither the stamina nor the functionality to serve as a information retrieval tool in contemporary libraries that serve as information gateways, rather than information repositories. Initial thrift may lead to expensive replacement costs if product inadequacy or extinction occurs.

Current and evolving information technologies are transforming libraries into information gateways that swing both ways. Patrons should be able to search their local library as well as the catalogs of other libraries without leaving their desks, using a sophisticated yet simple user interface that does not require mastering multiple command structures. Such a climate at a small college library will enhance student success at information retrieval, reduce the technological barrier to research and contribute enormously to the academic experience, while creating an environment that increases students' technological literacy.

AN EXAMPLE OF THE EVOLUTION OF AN IDEAL

At Sage, our technology model started as a seed for a project that would have demonstrated the intertwining nature of computer and information science. The project, designed to bring together the strengths of several departments using technology, started as an effort to create a technology based campus information system, on a very small-scale. While the idea received strong support from many individuals, getting any formal, institutional support was a slow, sometimes torturous process.

As the original proposal started on its way through the institutional grant process, it met with friction. Well-intended people, presented with an unknown quantity, were unsure of what to do with the idea. At points, the proposal appeared to disappear beneath the bureaucratic surface, resurfacing at unexpected locations and times. Even within the original departments, the concept met with some skepticism, related not to the idea but more to its chances of success at the institution. It became apparent to the writers that a shepherding process might be necessary to see that the original idea survived the process.

When it finally reached the President, she recognized immediately that this was indeed an excellent demonstration project. Having the endorsement of the president was an important step. She had the vision to see the small demonstration project as a possible full-scale implementation and welcomed discussion from the originators, both junior faculty members. Excited by that discussion, she legitimized the project by giving it her approval. Once that occurred, support from other individuals flourished, perhaps detrimental to the proposal. Many persons had various thoughts on the best way to proceed. Institutional endorsement of the idea also changed the scope of the project. In terms of dollars the project started at $50K, jumped to $250K, and finally went to $2.5 million. The full project was now escalated to its final dimensions. Where previously limited domains and stakes were involved the process now opened up to the entire community. At this point, whereas previously the writers had involved only a small group of individuals, representatives of various factions and technological literacy levels became involved. Although broader input was welcomed, many times, it led to confusion as the mix of conflicting priorities, talents, and procedural ideas often created difficulties. To ensure the integrity of the original proposal, it became necessary to enlist the support and guidance of a knowledgeable, respected more senior faculty member. Despite our efforts, much time was spent allaying fears, and controlling rumors, misinformation, and territoriality.

Having identified the proper scope of the evolved project and realizing the level of involvement from various constituencies, logistical questions surfaced. One ironic but serious problem was the lack of technology that we had to endure while seeking a technology-based academic model. Outdated equipment, multiple personal computing platforms, mixed disk formats, software packages and absence of connectivity created chaos. Flawed communication structures within the institution resulted in predictable accusations of empire-building, in consideration and incompetence. Some of these problems could have been alleviated if the institution had provided adequate equipment and other resources. In defense of the institution, the development of an institutional technology plan by definition will meet with technological limitations. As the grant evolved, we invited vendors to meet to discuss our tentative plans. They provided invaluable advice and gave credibility to the underlying concepts of the model. From these meetings, it became clear to the senior officers that a broad technological plan for the institution was evolving from the grant writing process. As the five year technological plan emerged, Kevin Synnott, the vice president in charge of computing technology, assumed a primary role in its genesis by making it a high institutional priority.

At this juncture, it was decided that the institution would endeavor to implement the plan whether or not the grant was successful. The fundamental changes that have occurred at Sage as a result of this project will have far reaching ramifications whether or not the grant is funded. Although the funding is important, the ultimate value of the enormous time and energy spent will be the changed philosophical outlook on technology at Sage. On an institutional time frame, Sage has changed over a relatively brief period of time, demonstrating a strong dynamic vision of the future.

A TECHNOLOGY MODEL FOR SMALL COLLEGES

What services and capabilities should be provided by an institution's technological resources?

1. Connectivity.
   A network should electronically connect:
   * Each academic office.
   * Each administrative office.
   * All students.
   * All campuses.
   * Libraries.
   * Access to Internet.
   * Modem/dialup for use from off-campus.
   * Integrative access to the institution's administrative computer system.
   * Email and public bulletin board services.

2. Computing Facilities.
   A college should provide:
   * Student access to computing in the Libraries, dorms, several other kinds of facilities in a variety of settings.
   * Workstations in each academic department.
   * Workstations in each administrative office.

3. Technological Support of the Academic Mission.
   A colleges should provide:
   * Electronic classrooms.
   * Portable instructor's computer/multimedia consoles.
   * A networkable On Line Public Access Catalog (OPAC).
   * Upgraded media labs including faculty authoring stations.
   * Faculty training and release time.
   * Technological support for faculty.
   * Student Academic Support Services.

4. Additional Considerations.
   * The system should be able to grow both in size (number of users) and in capability (e. g. interactive video transmission).
   * The system should be as "virtual" and as invisible as possible.
   * The system should allow any kind of user platform.
   * Users with various skills and technological literacy should be able to be effective users of the system.

How does one achieve these goals? At the heart of the enterprise should be an institution wide fiber optic network. This network should be based on the ethernet TCP/IP and FDDI communications protocols. These protocols allow for all of the flexibility called for above.

One network architecture model would be to have a central office that includes the main network hardware and software which supports a series of "supernodes". These supernodes in turn directly support and drive the individual workstations. Supernodes can de dedicated along traditional disciplinary/administrative function lines or can be assigned by building or other spatial considerations. This two-tiered hierarchical architecture will allow the sophisticated user to have as a complex and an independent local environment as they can afford, while at the same time providing the novice with as much support and simplicity as they need.

THE ADMINISTRATIVE PERSPECTIVE ON INSTITUTIONAL TECHNOLOGY PLANNING

From the administrative perspective, planning for technology needs to be consistent with planning done for all the activities of the institution. It needs to finds its genesis in the same thoughtful and aggressive posture that keeps an institution vital and responsive and it needs to be consistent with the mission and the vision for the academic enterprise.

Most would quickly recognize that technology planning brings with it a special challenge for most small colleges in the form of an enormous financial burden. The extensive initial outlay for both hardware and software, as well as the long-term commitment to both which comes within a networked environment, is daunting for many. The relatively rapid pace of technological improvement makes the entire idea seem to be a continuous drain on already tightened budgets. However, every institution recognizes that decisions about resource allocation must be made carefully, without ever losing sight of a college's top priority: the quality and strength of the programs and activities that make up the learning/teaching environment. Because technology has caused both reformation and transformation in education, it prompts, and even demands, a similar reconsideration of how the business of the college can be brought to bear on the necessary burden of moving forward with the needed and expected technologies.

It is important, therefore, to tie such planning to the long-range, strategic thinking about the institution and to its sense of planned for, intentional change. A college well determined in its directions for the future will be able to be confident about the deployment of its resources to meet the level of its own imagination about what it is and what it might become. Aligning technologies planning with institutional welfare liberates everyone to think about allocation and reallocation with a refreshed point of view, and should allow creative responses to emerge. Traditional models might well need rethinking; a new paradigm might emerge through the institution for planning, for administrating and for financing a project of such transforming scope. Such a process is at once daunting and exciting, causing transformation of "business as usual"; planning to keep everything the same is not planning at all.

It becomes a part of the administrative thinking to prepare for virtually all aspects of the college's work to be touched by technology and it becomes its obligation, with all partners to the enterprise, to assure wise selections, comprehensive implementation, collegial resource sharing and efficient and consistent application. If planning is truly institutional, then it follows that the best thinking throughout the institution will emerge and converge in the process. With a determined and shared awareness of the critical steps to be taken and a commitment to support the values inherent in the decisions reached, a plan can be successful, and the transformation we see everywhere in information management will have no less an impact on how we do business that it has had on the business we do.

CONCLUSION

Some closing thoughts and suggestions:

- The process of planning itself will strengthen an institution.
- Learning how not to plan, and what to avoid, is as important as learning how to plan.
- Identify which evaluations of the plan are valuable (e. g. come from reliable and knowledgeable sources that are not tainted by territorial and turf considerations) and which are not.
- Prepare a plan for planning that includes "winning over" by educating and enlightening those individuals who will ultimately play a major role in the adoption and development of a plan, and to minimize the influence of those who are afraid of change and tend to be obstructionists.
- It is important to get the administration to provide adequate technological and logistical support the planning process including computing equipment, clerical staff, and release time for grant writers (if they have other duties).

REFERENCES

Carpenter, Gail & Ann C. Edmonds. "The Library Intersection: Roles and Models for Information Services" In: Ringle, Martin, D (Ed.), Computing Strategies in Liberal Arts Colleges, pp. 373-397. EDUCOM Strategies series on information technology. New York: EDUCOM/AddisonWesley. 1992.

Kuhn. Thomas. The Structure of Scientific Revolutions, 1970.

Lane, Elizabeth & Craig Summerhill. Internet Primer for Information Professionals. Westport. Conn: Meckler Publishing, 1993.

Moberg, Thomas F. "Network Technology: Designing and Building the Infrastructure" In: Ringle, Martin, D (Ed.), Computing Strategies in Liberal Arts Colleges, pp. 233-262. EDUCOM Strategies series on information technology: New York, EDUCOM/ Addison-Wesley, 1992.

Multimedia in the Classroom

Carl W. Helms and Doris R. Helms
Educational Information Technology Laboratory
Clemson University
Clemson, SC 29634-1910
803-656-0623
cwhelms@clemson.clemson.edu

With the availability of powerful and reasonably-priced personal (small, single-user) computers, educational institutions have added distributed computing resources for staff and students as budgets and administrative commitment have allowed. These computers are often placed in large laboratories (or classrooms) where students may schedule their use, often tied to a specific course. Commercial, pre-authored software is provided and may include drill-and-practice exercises, simulations, educational games, and productivity applications (such as word processing software, spreadsheets, graphics packages, and, perhaps scheduling or electronic-mail where individual computers are tied into a local area network). Teachers may also be able to use software for administrative activities (maintaining attendance, grades, and other records, etc.).

As older, character-based computers are replaced in schools with faster, modern machines with color monitors capable of displaying graphics, more memory, and larger hard disks, which include or support peripherals such as a mouse, sound card, or CD-ROM, many new computers in schools are now capable of handling an advanced, computer-based technology known as *multimedia*.

WHAT IS MULTIMEDIA?

Multimedia is (are?) the integration of two or more media--graphics (including animation and full-motion video), text, and sound -- in an interactive environment. These "images" (which can also be referred to as data, or elements) are media that can be used to build a multimedia product or application that can be used in education (or that may be a product of the educational process).

WHY DO WE WANT TO USE MULTIMEDIA IN EDUCATION?

Various reviews indicate the effectiveness of multimedia...decreased time-in-training, and increased achievement, content retention, consistency, flexibility, and access, plus increased motivation and satisfaction -- multimedia is fun! (Wright, E. E., 1993, *J. Instr. Delivery Sys.* 7(6): 15-22). In addition, studies of interactive video show that the more interaction that is included in the application, the more effective it is in promoting learning (Fletcher, J. D., 1990, *IDA Paper* P-2372, Institute for Defense Analysis, Alexandria). Within the home, passive video often plays second fiddle to consumer electronics providing interactive games (such as Sego or Nintendo, with Compact Disk-Interactive and other CD-ROM technologies coming). Soon interactive applications will be accessed directly from cable or telecommunication companies' connections into the home, office, and school. The potential for commercial delivery of multimedia information- and entertainment-on-demand is just around the corner! Commercialization of interactive learning resources will increase access and decrease costs.

HOW DO WE "DO" MULTIMEDIA?

Multimedia is a software application or product developed (authored) to run on a particular type of computer configured with the added hardware required to handle multimedia "images." What is needed?

HARDWARE. Realistically, we need an IBM or compatible computer with an Intel or compatible 80386 processor running at 20 Mhz (megaherz), memory beyond 1 MB (megabyte), and an 80 MB (or larger) hard disk. With today's prices, a reasonable entry-level IBM or compatible computer would contain an 80486SX (or DX) processor running at 33 Mhz, 8 MB of memory (RAM), and a 200 MB hard drive. The color display should be VGA or better—this capability may be part of the computer or may be supplied by an added graphics card or adapter. In the Macintosh (Apple) line, an LC III would be a minimal platform. A mouse, touch screen, or other pointing device is necessary. A sound card (or adapter that fits inside the computer) may need to be added to capture and play back aural images (sounds). A video card (adapter) is necessary to capture or show video (full-motion) images. Other peripherals may include a scanner, videodisc player, or a CD-ROM drive.

VIDEO. We need to distinguish between the ability of a computer to display still graphics and full-motion video. Most personal computers purchased today come with an analog monitor driven by a graphics adapter to display the digital information generated by the computer (VGA, 24-bit color, etc.). However, video incorporated from TV broadcasts, tape, disc, etc. is an analog medium and requires special hardware for the digital computer to be able to display it. A single frame of video can be captured and digitized for use as a still image. However, the information stream found in full-motion (30 frames per second) and full-screen (525 lines) video contains more information than most computers systems can capture and save. In most cases, a video adapter's role is to play analog video in real time by converting its scan rate to that of the computer monitor or to digitize (but not save) the data stream so it can be resized, windowed, etc., within the multimedia environment. However, the analog video source (tape, disc, etc.) is necessary each time the video segment is used by a multimedia system. This digital/analog hybrid provides an integrated system using a single screen for display in the ideal multimedia environment today.

It is also possible to display video information directly on a second screen -- a television or video monitor -- while the computer displays text or other data and provides user controls (this is a typical configuration for many schools using Hypercard or Linkway). Video, including single-images from a video disc, can also be displayed by a teacher directly on a TV or monitor using a disc player and a controller or bar-code reader without incorporating a computer at all. Videodisc, thus, provides a "scalable" technology. In summary, multimedia elements located on a videodisc can be used (1) directly by the teacher, (2) with a simple computer that controls access only, or (3) with a full-fledged hybrid system providing sophisticated integration of media. Many educational resources are currently available on videodisc, recommending it as a major player in educational multimedia at present.

However, one other video "flavor" should be mentioned. Digital video (in which video information is stored as 0s or 1s rather than signal amplitudes) is coming and will eventually be available on the desktop. The advantages of digital video include (1) the ability to store digital video images directly on the hard disk of a computer (or other digital media) and (2) the ability to supply the information over digital networks. The lack of standards in this arena, the enormous storage requirements for video data, the necessity to move vast amounts of information quickly, and the need to (compress and) decompress video information in real time make digital video a high-end commercial and desktop solution at the present time. Digital "video" supplied by QuickTime (Apple) and Video for Windows (Microsoft), similar technologies now available without added hardware support, are a less than full-screen, full-motion solution and may not be appropriate in many multimedia applications. Intel's Digital Video Interactive (DVI) is a full-screen solution but it requires an expensive hardware card or adapter for acceptable playback and does not (yet) provide video approaching the quality obtainable from an analog videodisc.

Stay tuned! Desktop digital solutions are coming, but educators are advised to wait for standards to emerge, costs to drop, and for better compression algorithms and authoring software to emerge.

SOFTWARE. Speaking of software, there are at least four types used in the multimedia arena:

1. Operating system software (DOS, OS/2, System 7, Unix) is necessary to run the computer. Programs such as Windows and QuickTime enhance the capabilities of operating systems but are not operating systems as such (Windows NT *will* be an operating system). All computers require an operating system.

2. Image- (datum-, element-, medium-) preparation software is used to capture or create and edit graphics, prepare text, and capture and edit aural images (sounds) (Corel Draw, Rio, AutoCAD, etc.). Let's call this media preparation software. Capturing (digitizing media) can also be referred to as *production*.

3. Authoring software is used to arrange media elements, add links, ties, or navigation aids among the elements, and prepare the final product actually used in or resulting from education or training activities (Hypercard, Linkway, Toolbook, Authorware, for example). Note that some authoring software may also provide tools that assist in media preparation. Authoring is *post-production*.

4. The final product is the end result of gathering media elements and authoring an application. This product is harder to name because it is harder to define -- it may be a linear presentation used to support a teacher in class, a one-on-one exploration of a given subject with hyperlinks connecting related topics, an application that provides directions or information to the user, a simulation or educational game, etc. It might even be a package designed primarily for entertainment. Lacking a better term for this fourth category, let's call it product presentation software.

There are two different types of product presentation software that can be used in education and training:

1. Product presentation software may form the basis of a multimedia lecture or lesson in which teachers "illustrate" information or data for students and demonstrate facts, principles, and concepts (lecture support). This software can also be a stand-alone, one-on-one application used directly by students on their own machine, in their own way, and at their convenience-computer-based education or training (CBE or CBT). The use of multimedia in lectures and computer-based education is an example of using **formal media** -- media that are developed and arranged ("authored") by an "expert," someone other than the student or learner.

2. Students may use graphics-capable computers with various productivity applications (media-preparation software) to create reports, essays, tables, and graphics or a combination of media. These products constitute **informal media** -- media that are developed or authored by the learner himself or herself. With the incorporation of any two media elements, these creations are multimedia products. Enhancements possible today include the addition of sound clips, digital video clips in a window, full-screen, full-motion analog video, and connections ("hyperties") linking media elements ("hypermedia") in flexible ways.

MULTIMEDIA AT CLEMSON UNIVERSITY

The College of Sciences at Clemson University established the Educational Information Technology Laboratory (EITL) in 1989 to (1) assist faculty in developing educational applications using advanced technology, (2) develop media for use in education, and (3) maintain and enhance an interactive, multimedia software environment called *InteractiVision*™. This software has provided our primary multimedia environment.

EXPERT-AUTHORED PRODUCT PRESENTATION SOFTWARE. Development, initially focused on formal media for use in lectures in introductory biology (lecture support), is now expanding to other subject areas. EITL assists faculty in *planning* their use of multimedia (**pre-production**), in *preparing images and capturing* (**digitizing**) *elements* (**production**), and in *authoring* (**post-production**). As faculty become involved, they increasingly assume a more active role in all aspects of development and, for convenience, come to perform much of their development on authoring stations located in their own department. EITL continues to assist with complex graphics or animations that require additional equipment or expertise.

Media developed for lectures are also available for constructing one-on-one applications (CBE). However, this requires efforts that are several orders of magnitude higher. Stand-alone programs require careful design, production, authoring into a planned interactive environment, and assessment to determine if educational objectives are being met. User testing and assessment instruments can be included in the final product presentation software to suggest where changes are needed. If full-motion video is needed, the production of a video disc with images and full-motion video segments may also be necessary to provide video images.

At Clemson, an intermediate format between the product used to support lectures and a full CBE application. The visual images used in a lecture can be recorded sequentially to video tape with a "voice over" done by the instructor to provide context -- but not the full lecture. This 5-10 minute tape can be reviewed by students in a learning laboratory to reinforce images presented in the lecture.

TIME AND EFFORT? Variations in the effort expended in creating multimedia products relate to the intended uses of the final multimedia product software and to the time and physical and financial resources available. Many teachers, once they have mastered the rudiments of computer operation and media-preparation and authoring software, find it as easy, and often more satisfying, to prepare multimedia lectures -- the main difference is that they do require preparation; instructors cannot "wing it." As image resource libraries are built in a given subject area by teachers working together, authoring also requires less media preparation and becomes easier -- teachers can pool resources. These resources can also be incorporated in one-on-one CBE applications as time permits.

PROBLEMS. If, however, the goal is to produce salable, commercial software items, more effort is required. Clear instructional goals must be set and assessed. Appropriate instructional design must be incorporated. Copyright issues must be carefully observed (publishers may be lenient if images are used locally and are taken from materials purchased for the course). Facts and concepts must be correctly presented. Assessment must show that the product, in some way, facilitates "learning." Commercial efforts may well require an media center with appropriate hardware and software and staff with various talents (instructional designer, subject-matter expert, electronic artist, assessment specialist, etc.) included in the development team.

Other problems with commercial expert-authored software include:

- the design and images chosen may not suite the particular teacher, student group, or level;
- teachers and students do not feel any pride of ownership;
- the product software may not be edited or modified in any way to incorporate new information;
- appropriate assessment tools may be lacking; and
- commercial packages tend to be expensive.

On the other hand, the quality of materials and the instructional design included in commercial multimedia applications such as *Columbus* or the *Illustrated Great Books and Manuscripts* (IBM) require resources of information, expertise, and talent that could not be assembled in most schools.

However, media and product presentation software developed by a subject area group or within a given department or school may result from a cooperative and coordinated effort that will include "ownership." Thus, creative teachers are more likely to use product software they have had a role in developing or to help students become the authors of product presentation software themselves!

LEARNER-AUTHORED PRODUCT PRESENTATION SOFTWARE. As an alternative to expert-authored product presentation software, learner's may be placed in an environment in which they have access to multimedia computers and media production and authoring software, not pre-authored product software. Given a question or concept, individuals or small groups use the computer workstation and its software to capture (digitize) information (prepare media) and author their own product— an "electronic" essay or report that represents both a synthesis of media and a creative activity. In this environment, the teacher becomes a guide, facilitator, or mentor and the author is the active learner. The product software can be assessed on its merits which including creativity as well as content. These products may stand alone and have no further function, but in most cases they should be saved. Both the media elements and the final product software could be used as a basis for further refinements by other groups of students or for augmentation to broaden the "essay" into related areas. This mode of progressive and cooperative learning is being actively explored in a several schools (re., the Cumulative Curriculum Project of the Dalton School in New York City).

MEDIA RESOURCES, NETWORKING, AND EDUCATION. Data (information) used for developing media can be varied. They can include relevant books, journals, magazines, slides and photographs, news broadcasts and other live video productions or scenes, video tapes and discs, compact audio disks and tapes, personal experiences or interviews, three-dimensional objects, flat art, computer-generated graphics, animations, CAD drawings, spreadsheet tables, equations, etc. They can be gathered and saved locally or can be placed on a multimedia server on a local area network (LAN) where all learners/authors in an institution have access to them.

Given networks with multimedia database servers containing text, graphics, sound files, digitized video, and connections to analog video sources, sound inputs, and CD-ROMs, among other devices, students will increasingly be able to (1) communicate with peers and mentors and to (2) access global information resources. Local and campus networks are now connected to those on other campuses through a variety of inter-networks -- the Internet being a world-wide collection of networks available to educators. Over 8,000 networks, 1.3 million computers, and 8 million users now constitute and access the Internet. These numbers have been doubling since 1988.

The groundwork has also been laid for the National Research and Education Network (NREN), a planned national network that will connect to the Internet, providing high speed (gigabit/sec) sharing of computer resources and transmission of information. By the end of the century *all* schools, K-16, and many business and industry groups will be interconnected. Students will be able to communicate with each other and with teachers, at home and abroad. They can communicate with scholars in universities or industry, engineers in the field, researchers in their laboratories, medical practitioners worldwide, or friends next door. Located on the Internet today are the card catalogs of many libraries. An increasing number of documents are also available in electronic format in their entirety (including text and figures). Text-based databases are available on virtually any topic. Several repositories of multimedia information including sound and video are also being developed.

With free access to global information resources, students of the future will become "lifelong" learners and "just-in-time" learners. As technologies and jobs skills shift, information will be instantly available to assist all of us in growing and changing. Problems? At least two. (1) With over a million nodes on the Internet by the end of the decade, how does one locate information? Directories and navigation aids -- user tools and interfaces that make finding information transparent to the user -- become necessary. And, (2) once information is located, how reliable is it? Anyone can supply information to the open databases of the network so that ways to select and validate data will be essential.

Another problem? How can we afford advanced technology for education? First, personal computers are cheaper today and prices continue to decline as functionality increases. Hardware to capture and display sound and video is also coming down in price -- in fact, just as graphics capabilities have been integrated onto the mother board and chips supplied with all computers, sound and video capabilities will soon become integrated at the chip level and their costs will be incremental at best. Second, consumer economics are at work. Computers have become commodities -- like TVs, VCRs, camcorders, and interactive digital games -- and we will see them merge with other commodity items, especially the TV and telephone. Third, as these developments are unfolding, the telecommunication giants (AT&T, MCI, Sprint) and regional Bell operating companies (RBOCs) are competing with cable companies to be the first to your home (or office or classroom) with interactive information resources providing hundreds, even thousands, of channels of "information-on-demand." These connections will feed our computer-video phones of the future. With voice recognition technology, keyboards will vanish and our personal "electronic assistant" will shrink in size. With cellular technology, the only intermittent connection our portable *"Compu-Vu-Phone* R" may need in the future is to power (or sunlight?)! Education will no longer be limited to a physical space or location or time or social context or? The potentials are limitless and exciting and portend a new way of communicating and learning. Could this pending digital revolution in information technology equal that engendered by the printing press?

AN APOLOGY AND A VISION

Looking at the vista of advanced technology in education that we have outlined here, Clemson and EITL have not come very far. We, like many others, develop media and incorporate it into expert-authored product presentation software, but, like many others, lack the resources to develop great products for computer-based education. Perhaps, however, the future of education does not lie with expert-authored applications but with learner-authored applications. In this case, providing media resources and an environment that fosters the development of informal media may actually be our best choice. Will perhaps the day come when the world's information resources are available over transparent, world-wide, internetworks to life-long learners wherever they may be and whatever their social, ethnic, and economic backgrounds? Move over printing press! We may have found a real place for the world of computers!

Machine Models of Human Pattern Recognition - The Case of Melody Recognition

William J. House
Department of Psychology
University of South Carolina-Aiken
171 University Parkway, Aiken, South Carolina 29801
803-648-6851
wjhouse@univscvm.csd.scarolina.edu
wjhouse@univscvm.bitnet

After the seminal theoretical work of Whitehead and Russell (1910 1913), Godel's (1931) "incompleteness theorem", Turing's (1936) "effective procedure" the forerunner of the algorithm, and neural calculation demonstrated by McCulloch (1949), it remained for Allen Newell and Herbert Simon (1959) to create the "General Problem Solver" (GPS) before it could be said that a machine demonstrated anything like human ability and performance. Most scientists who, in one way or another, focus their energy on computing machinery are aware that the foregoing outline represents the beginnings of "Artificial Intelligence" (AI). Many fewer, however, know that as AI came into being in the early sixties, so also did "Information Processing" or "Cognitive" psychology have its modern beginnings. Indeed, some have considered the contemporary advent of computing machinery to be for psychology analogous to the invention of the microscope for biology.

Newell and Simon were not working to establish AI; rather, their principal concern was psychological and they established what has come to be the representative approach for experimental cognitive psychology in the last half of the twentieth century.

Today, I will sketch for you the significant and what I believe to be instructive differences between IP and AI theory and present you with an example of what I think may be helpful IP approaches to computer programming in general.

DIFFERENCES BETWEEN AI AND IP

The goal of Artificial Intelligence (AI) research and practice is to create programs and machinery which efficiently solve problems and accomplish difficult work. An axiom of this goal is often assumed to be that machinery should do jobs that people find onerous, difficult, or which impedes human creativity. Thus, the results of such efforts are tireless robotic workers, unperturbable information searchers, and "expert systems" which purport to overcome the failings of even the best human problem solvers. Thus, AI strives to maximize the abilities and efficiency of programmable machinery.

On the other hand, IP psychology's goals are to model, for better or worse, all aspects of human functioning. Ultimately the IP scientist wishes to understand actual human behavior by attempting to create analogous machine behavior. These programs, consequently, must perform the same stupid mistakes, the merely adequate actions, as well as the glorious accomplishments that people regularly produce as a rule.

The method of IP research, as Newell and Simon initiated in the GPS research, consists of six steps. For example, in the case of human chess playing behavior research on the particulars of what people normally do is first accomplished with real human subjects. As a subject goes through thinking about a set of chess moves, he/she is trained to produce a "protocol": that is, an oral monologue of ongoing thought is produced and recorded. Second, the protocols from several subjects are distilled into a set of common heuristics or behavioral rules. ("Heuristic problem solving" is conceivably the most important contribution of Newell and Simon's GPS to IP psychology and, perhaps, also to computer science.) These rules are, third, crystallized into decision steps in the form of program algorithms. Fourth, the program is presented with a set of stimulus input conditions and its performance is recorded. Fifth, human subjects are presented with the same stimulus conditions that the program confronted: the human performance is, likewise, recorded. Sixth, the computer performance is used as the predictor of the human performance. With regression statistics, the amount of variance accounted for by the hypothesis is the test of the efficacy of the machine model.

"NAME THAT TUNE"

One of the most interesting abilities that people can exhibit is that of recognizing musical melodies. The old quiz show "Name that Tune" placed this ability into the form of an interesting contest in which people gambled on being able to guess a tune in fewer and fewer notes: "I can name that tune in four tones." This is a scientifically significant behavior because people make their identification of a temporal pattern (the tune) on the basis of only a part of that series; this is a fundamental example of using inferential processes, long, and short-term memory to identify an ongoing temporal pattern.

Based upon a large amount of human experimentation with "serial order" behavior, I have constructed a number of computer models of melody recognition. For the purpose of this paper, I will describe two of these: CON1 and INT1.

MEMORY AND STIMULUS REPRESENTATION

In both of the models considered here, short and long-term memory (STM and LTM) work in the same way: a long list of melodies is represented in long term memory as a sequence of intervals (rather than tones) associated with a melody name. People appear to readily recognize such intervals rather than frequencies of tones unless they are blessed (or some would say, cursed) with the very unusual ability of "perfect pitch". When the tones of a melody which is to be guessed is presented from the keyboard, short-term or "working memory" takes pairs of tones (frequencies) and calculates their musical interval (e.g., minor third, perfect fifth, augmented sixth) for scanning of LTM. This is apparently what happens in the inner ear and sub cortical levels of the human brain. It is at this step that CON1 and INT1 differ. INT1 directly compares the derived intervals with those stored in LTM. Some research suggests that the intervals of melodies are the critical element by which recognition occurs. CON1, on the other hand, further calculates the contour of the melody; that is, yet another melody representation is created which is merely the sequence of tonal ups and downs that make up the melody. This temporally derived melodic contour, which many human experiments have implicated as the critical stimulus for melody recognition, is then matched with LTM information.

PARTIAL MATCHING PROCEDURE

As each contour (CON1) or interval (INT1) is briefly held in STM, a scanning of LTM immediately commences. All LTM melodies which are identified as having the same interval (INT1) or contour (CON1) as that which is stored in STM are given a higher weight than those LTM melodies which do not match, which are given a negative weight. Thus, after a few input tones are processed, each LTM melody gains a different weight some quickly become highly weighted but most have a negative weight. An arbitrary trigger value is set such that when that value is reached by any LTM melody, that melody is "guessed".

LEARNING

After the guess has occurred, the keyboard operator inputs whether the guess was correct or incorrect. If the program's response was correct, then the trigger value associated with the correctly guessed melody is diminished by an arbitrary value; if the response is incorrect, then that trigger value is increased. The result of several guesses, then, is that some LTM melodies come to have high and others low trigger values essentially the program becomes biased in favor of some and against other melodies.

RESULTS

When the responses of CON1 and INT1 over several trials is used to predict similar human behavior, CON1 accounts for considerably more variance than does INT1. That is, even though INT1 performs better makes fewer mistakes and improves in its ability over time CON1 performs more like real people.

CONCLUSION AND IMPLICATIONS

The difference between AI and IP research is a matter of capitalizing on the power of computing machinery (AI) versus seeking understanding of human behavior (IP). Information processing researchers have gained much from their unquestionable dependence on AI findings and procedures. However, computer programmers in general will discover the findings of IP research useful for many purposes other than clarifying the details of human functioning.

For example, machine memory is unquestionably superior to human memory: a $20 tape recorder will remember far more detail for a greater period of time than the very best human mnemonist. By most people's criteria, the recording machine, however, is extremely low in intelligence.

Understanding intelligence is more than a matter of understanding extreme degrees of capability. Our best model of intelligence continues to be that which is regularly observed in humans and that model shows us that global intelligence is a delicate balance of efficiency and inefficiency. Plainly, a simple computer program could be construed that would directly match tonal frequencies in memory with input frequencies and that program would produce recognition which would outperform either of the programs presented in this paper as well as any human listener. As Newell and Simon and others learned early on, intelligent performance requires making creative guesses which might be wrong, learning from mistakes, and flexibly pursuing ends with rule based action.

REFERENCES

Godel, Kurt, Uber Formal Unentscheidbare Satze der Prencipia Mathematica and verwandter Systeme, Part I, Monatshefte fur Mathematik und Physik, Vol XXXVIII (1931), 173189.

McCulloch, W.S. The Brain as a Computing Machine, Electrical Engineering, Vol. LXVIII (1949), 368376.

Newell, A. and Simon, H.A. The Simulation of Human Thought, Report No. P1734. Santa Monica, Cal.: The RAND Corporation, 1959.

Turing, A.M. On Computable Numbers, with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, Vol. XLII (1937), 230265.

Networking Made Easy: The Bulletin Board to the Rescue!

Louis M. Janelle, Jr.
Director of Computing Services
The University of Akron - Wayne College
Orrville, OH 44667
216-683-2010
LouJanelle@UAkron.Edu

Wayne College, a branch campus of The University of Akron located in Orrville, Ohio, installed a Banyan Vines local area network in June of 1991 to support student, faculty, staff, and administrative computing. The network provides a link to The University of Akron campus located some 30 miles northeast of Orrville. This connection allows anyone on the network to connect, via telnet or ftp, to the mainframe computers and minicomputers located on the Akron campus. The Akron campus connection also serves as Wayne College's gateway to Bitnet and to the Internet.

When the network was originally set up, software to support a local area network bulletin board system, called TackBoard, was purchased from Trellis, Inc. (a Banyan Vines third party developer). The bulletin board runs on a dedicated PC workstation and serves as a campus wide information system for Wayne College.

One of the nice features of the bulletin board is that it supports "executable" bulletins — short DOS command sequences substituted for bulletin text. When the bulletin is read, the sequences of this "executable" bulletin are performed.

In the Trellis bulletin board implementation, each "board" and each "bulletin" is stored as a separate, coded, ASCII file. Once can start up TackBoard with a specific board, e.g. TB - ABC12BB will start the bulletin board at the "board" whose DOS filename is ABC12BB.

The system's security can limit one's access according to the following settings:

a) Read-only access — users can read bulletins but cannot add bulletins of their own
b) Public access — users can add, modify, and delete their own bulletins on a particular board
c) Manager access — users can modify or delete ANY bulletin on a specific board
d) Supervisor access — allows a user to add and delete sub-boards and assign access rights to other users

TackBoard's executable bulletin support makes it easy to provide network users with a front-end to the Internet. Instead of needing to know how to issue telnet or ftp commands, users merely access the BBS, choose an application or site listing, and execute it. To date, I've included a number of gopher sites on the Wayne College BBS. I've also included local access to Bitnet's "List of Lists".

To automate searching the "List of Lists", I downloaded the entire file from an ftp site. I then created an executable bulletin that runs the "viewer.exe" program from PC Tools. This program has a built-in search utility. The program is excellent for viewing or searching very large documents.

In fact, in searching this list in preparation for today's presentation, I discovered a discussion list called CWIS-L, a mailing list for discussing the creation and implementation of campus-wide information systems. Bitnet users may subscribe by sending the command, SUB CWIS-L FirstName LastName, to LISTSERV@WUVMD. The Internet address is: LISTSERV%WUVMD.BITNET@vm1.NODAK.EDU.

The screen snapshots, which will be shown and distributed in printed form at the conference, should give the reader an insight into the bulletin board's organization. Some of the slides depict the creation of an executable bulletin. The panels also include views of user and supervisor options available for creating files, editing files, and managing boards.

Managing Computer Labs with "For Credit" Lab Technicians

Dan W. Krupp
Indiana Vocational Technical College
3501 First Avenue, Evansville IN 47710
812-429-1448
dkrupp@ivy

BACKGROUND INFORMATION

Indiana Vocational Technical College (Ivy Tech) is a two year technical school serving students in the state of Indiana. In fact, the College is the third largest state supported school in Indiana, with Purdue University and Indiana University being the largest.

Degrees granted include the Associate in Applied Science degree and an Associate Science degree. The latter degree allows our students to take two years of school at Ivy Tech and finish with a Bachelor Degree at the University of Southern Indiana.

Ivy Tech has thirteen regions with campuses throughout the state. The Evansville campus is the third largest serving 3500 students. We have many night classes, and many our students can attend only the night sessions. Our average student age is 31 and includes many people who are coming back into the job market after an absence of several years.

The Business Division is the largest on the Evansville campus, and all the programs in this division require at least one microcomputer course as part of the curriculum. The Computer Information System (CIS) program has two specialties available: Programming Specialty and Microcomputer Specialty.

With the large number of students in the division taking computer related courses, computer availability is most important.

The following computers are available for student usage:

IBM 9370 — This machine is used to communicate with the College mainframe located in The central office in Indianapolis. Student use includes using the communication facilities available on the computer for Data Communications types of course work.

IBM AS/400 — This computer is used extensively by computer programming students for programming, database and operating system classes. One computer lab is used for the classes using this computer. In addition, access to this computer is also available for those students who choose to use the microcomputers hooked to the AS/400 via Token Ring connection.

Maintenance for this computer is handled by a CIS instructor who also teaches computer programming courses.

IBM Microcomputers

There are six microcomputer labs with 150 computers that are used to teach microcomputer-based courses. Two of these labs are "open" labs, and the other four are used for course work.

All the microcomputers are linked to two file servers via an IBM Token Ring network. This network also connects all faculty offices on the campus and many staff member computers.

Maintenance on the computers in the labs, faculty offices, and staff offices is done by a Computer Technician working under the guidance of a CIS instructor who also teaches microcomputer based courses.

OPEN COMPUTER LAB

The number of hours available to use the open labs varies from term to term, but 71 hours per week has been the average for the past several terms. These hours generally are from 8 AM to 9 PM Monday through Friday with Saturday availability being 8 hours.

The open lab contains 48 IBM computers; 36 are 486 machines while the remaining 12 are 386 class machines. Four of these machines also have laser disk capability. All software taught on the campus is also available in the open lab. This software also includes those applications taught by the Applied Science Division. The open lab also contains four dot matrix printers and two laser printers with Postscript capability.

Students using this lab include those taking General Education courses that involve computer usage, Nursing students using specialized health care software, Applied Science students using drawing programs and specialized tutorials, as well as the Business Division students including CIS majors.

The lab is supervised by a paid lab technician who is available during each hour that the lab is open. These technicians are CIS majors who show an interest is performing this type of work. The lab technicians report to the Computer Technician.

Application Manuals as well as current computer magazines are also available for review in the lab.

COMPUTER LAB TECHNICIANS

Education

Each lab technician must complete the Ivy Tech course "Introduction to Microcomputers". Optionally, the technician must have completed a Microcomputer Operating System class. Lab technicians must be current Ivy Tech students, and a preference is given to CIS majors.

Hiring Criteria

Before any student is hired to be a lab technician, the attendance record is checked with each instructor having the student in class. Any attendance or punctuality problems disqualify a student from being a lab tech.

Turnover

We have encountered few turnover problems as most of our lab technicians stay with us until they graduate.

Duties:

- Clean keyboards, monitors, printers on a weekly basis.
- Keep the lab free of litter on the floor and each workstation.
- Complete a supply inventory weekly for all computer labs. This inventory includes computer paper, printer ribbons, laser printer paper, etc.
- Perform virus checking periodically for all computers.
- Make minor computer repairs when possible.
- Assist student with hardware situations such as printer problems, application exiting problems, etc.
- Reconnect stations to the network in cases of disconnect problems.
- Load software on demand for selected applications.
- Install new or changed hardware such as additional cards, printers, memory.

Training

Training has become an easy task because of the type of students hired for lab technician duties. The students who make the best technicians tend to be those who use the lab frequently and spend time in the lab doing classroom assignments.

These students use the computer for class work for non computer related courses also, so there is a tendency to recognize the value of the computer and how to use it. Before they are hired, the students have spent a lot of time in the lab and have learned what activities are performed by the lab technician.

Formal training activities include group sessions, one on one meetings, written materials on how to perform tasks, etc. The education process is an on-going activity for the lab technicians, and they have been able to suggest and provide training for themselves on hardware and software usage.

Other Items

- Homework, game playing, and other similar activities are not allowed during work time for the lab technicians.
- Cooperation between the lab technicians has been excellent in that they will "cover" for each other during times of scheduling problems. If special circumstances arise where a lab technician needs to be absent from work, others will "swap" times.
- If any hardware problems occur, the lab technicians will first try to fix the problem before calling in the Computer Technician for additional help.

LAB TECHNICIANS FOR CREDIT

The Computer Information Systems curriculum calls for six hours of electives in the computer field. These electives are higher level courses in the IBM PC type of computers or other courses that use the Apple / MacIntosh type of computers. Such courses have consisted of Desktop Publishing, Computer Graphics, or additional computer programming courses.

To provide more "work experience" while attending classes, I developed a special course called Special Topics in Networking. This course is taken as an elective to satisfy the requirements of the curriculum. Enrollment in the course is by invitation only and permission of the instructor is definitely required. Enrollment is limited to a maximum of two students, and the course has been offered only during the summer term so far.

The course has involved all laboratory work using existing computer facilities as well as new computer equipment to be installed. Reading assignments involve vendor software manuals and DOS reference materials from a prerequisite course.

Prerequisites

Prerequisite course work consists of completing the Microcomputer Operating Systems course with a grade of A. Other criteria include an excellent attendance and tardiness record. An excellent demonstration of work habits in the classroom is also included in determining who will be allowed to take the course.

Duties

This course mainly uses a hands on approach to solving various networking problems and situations. Some of the time devoted to the course is spent in the open lab while other time is spent working on the various computers throughout the campus.

Installation

Computer, printer and network access is provided for labs, faculty, and staff computing equipment.

New Computers

As new computers are purchased, hardware placement and electrical wiring considerations are reviewed. Network connections must be installed and hooked up to the new computer.

Additional hardware needs such as token ring cards, memory expansion modules, laser disk cards, etc. are also installed in new computers.

Software

Network software is installed on each machine and connections are tested to verify that the user has access to both the Regional network as well as access to the Central Office system for student registration, financial information, etc.

Application software such as word processing, spreadsheets, and graphics is installed on each machine based on the needs of the user. Access is also provided to applications that reside on the network based again on the needs of the user.

Menu

A complete menu is set up so that the user has access to both those applications on the computer's hard disk and applications on the network.

Procedures

The menu system on the network is the same for all lab computers. What the student sees is always the same in each classroom lab. Application access via the network for faculty and staff also uses the same menu screen.

Each faculty or staff member also has a customized menu depending on the software available on the individual computer.

Loading of software applications has been standardized to provide ease of loading time.

Summer 1993

The wiring scheme throughout the campus has grown rapidly and not always in a controlled manner because of time restraints. While the existing scheme works well, there have been two occasions of hardware failure on the network where finding the problem has been very difficult and time consuming.

A complete rewiring is not necessary, but some connections need to be changed to fit a more orderly schematic of the system. Also, there is a need to change the booting procedures on many of the computers to provide for updated software and hardware. With both tasks needed, the course for this summer will consist of both a wiring change and computer software changes.

Duties for Summer, 1993 include the following:

- Introduction to Ivy Tech network procedures
  This task is done by the instructor and makes the student familiar with the existing system at Ivy Tech. The students are familiar with how the system is used at this point, but are now learning what is "behind the scenes".

- Documentation of existing token ring wiring scheme
  The existing wiring schematic for the network is re visited at this point and all connections are checked and verified against a master schematic for the system. This step must be performed before any changes can be contemplated.

- Planning for new wiring scheme / changes to existing scheme
  This step includes redrawing the schematic to reflect the changes that are to be made. A new wiring schematic is developed and checked for reasonableness.

- Planning for new software booting procedures
  During this step, booting instructions for all computers are reviewed, possible changes determined, and written procedures are written.

- Rewiring portions of the token ring
  This step involves the actual stringing of cable, unplugging and re plugging token ring connections and testing each connection at the time of the change.

- Changing software booting procedures
  During this step, actual changes are made to the hard disks of each machine for boot information. Changes are also made to the boot diskettes for those machines that do not have a hard disk.

- Testing both the new wiring scheme and booting software
  Although each wiring change and each boot change was tested on an individual basis, the entire network needs to be checked under a heavy load to verify that the changes will work.

Weekly tasks are assigned to each student. These tasks include what is to be done and how long it is estimated each task will take. The student is then evaluated at the end of the week as to what was accomplished, time required, and the satisfactory completion of the project. A written report is due each week from the student stating the above and what was learned from the week's efforts.

Cooperation between the students is very important during the course to make sure that the network "survives" and that a catastrophe does not occur. All the changes being made will occur on a "live" system, so care must be taken during every step and phase.

I have been known for my "no guts, no glory" style of teaching and classroom assignments and this style has carried over into projects such as the above. With careful planning and review, disasters should not occur.

Lander's Touch Screen Building Directory

Terry W. Lancaster
Lander University
Computer Repair
Greenwood SC 29646
803-229-8256

Lander University presently has two touch screen based building directories in place and others scheduled to be implemented. Each contains a listing of the faculty and staff that work in the building by name, room number, and telephone number as well as a listing of campus events, a screen saver of campus scenes, and a tour of the buildings at Lander.

These were conceived as a replacement for the movable letter type of directory that is commonly in use. Lander maintained these through our Telecommunications Office using student workers. This proved to be so labor intensive that complete updates were made only once a term with deletions from the list and additions made to the bottom of the board only once or twice a semester.

Mr. George Franke, Assistant Vice President for Technical Services, saw a system using a touch screen that was under development while at a conference at Ball State University and began the investigation of what such a unit would require to set up at Lander.

The first step was researching the touch screen mechanism. At that time there were two different types available to use on an IBM compatible platform. The first used an infrared transmitter/receiver frame that fit around a regular monitor and detected when an object interrupted the beams. The other used a screen overlay that sensed pressure by using a membrane and needed to be physically touched to register. Each was capable of reporting an area by returning an X-Y coordinate. Both types used a standard serial port to interface with the CPU.

After considering the merits of each, the infrared grid type was selected. That was because it had no mechanical parts, did not require pressure to be applied, and was less expensive. Carroll Touch Screen was chosen as the manufacturer as they would deal directly with the University.

For the display, a Zenith VGA monitor was used. It uses a flat screen so could have the light beams a uniformly close distance from the CRT face. If a curved screen had been used, the beam distance from the face would vary over the display area and may have presented a problem in selecting a small target. In addition, the CRT glass is textured so glare and fingerprint smudges are less apparent. A color VGA was not an absolute requirement in the beginning, but came to be very important as graphics became part of the package.

While the display and touch screen mechanism had to be purchased, the other components that were already on-hand were incorporated into the project whenever possible. An original five slot IBM PC was used as the CPU base. These were acquired from administrative and faculty offices where they were not being fully utilized and were transferred to Technical Services. The parts not needed were placed on the spares shelf and what was kept in the machines was the diskette drive and controller. one serial port adapter. a VGA monitor adapter. and a real-time clock card.

Before purchasing the Carroll Touch Screen frame. it was found that they specifically recommended two software packages from other companies to develop applications. The first package was 'VPIC'. It is used to interpret grid interruptions and provide these locations to the program in a form it can use. The second is "SayWhat' and is used to create a target screen that can then be referenced with X-Y coordinates.

At this point all the initial hardware and software that was needed was at Lander. Both 'VPIC' and 'SayWhat' provided the entry code for their packages in BASIC. C. and Pascal. Mr.

machine was to be in a common mall area to be shared by two buildings and a central location was selected by a support pillar. Then a very short person and a very tall person were sent to determine the optimum height for the display. Custom enclosures were then constructed by Lander's Physical Plant carpenters.

The time frame from the delivery of the equipment to the installation of the working system was about five months. The approximate unit cost of the two systems now in place is less than $2,500 total cost. The unit price should drop as the other systems come on line as the will help absorb the development costs.

Maintenance of these systems is carried out by Lander Work-Study students although Mr. Pombo has since graduated and no longer is involved with the equipment. Every morning a walk-by is performed by one of these students. This is necessary because the Lander Campus experiences power fluctuations that often affect the machines. A bootable diskette with the clock utility programs on it are kept in the enclosures to reset the systems if needed. Shortly after the installation, the temperature inside the enclosures were believed to be excessive and fans were installed to insure a constant airflow through them.

The information in the directory files can now be readily updated and is entered as it arrives to the Telecommunications Office. Also Campus Events can be added and edited as needed.

As might be expected, aspects of the touch screen systems have been identified for modification during the ongoing maintenance. The original code has the data file names hard coded into it. A separate file should be kept that contains the names of the files unique to that location in that we now have different files that must have the same name. Also in that name file should be the names of the screen saver images that are called. This would allow current information, such as sport scores, to be produced quickly and kept for short periods of time. Additions have been proposed to add floor plans for each building to be combined with directions from the touch screen location to a specific room. Also, suggestions have been made to make the screens easier to use by adding highlighting bars that can be moved through the lists and the option of scrolling through the lists instead of using screen changes.

These suggestions for improvement could be implemented in the touch screen system, however it is a useful tool and meets its goal as it stands. It has replaced an information vehicle that was difficult to keep up current and did so by using resources that had a low impact on the budget and provided a useful experience to the student workers.

Computer Integration in Professional Education: A Case Study of A Regional Institution of Higher Education

Charles D. Manges
Larry Harris
Division of Education
Wayne State College
124 Brandenburg, Wayne NE 68787
402-375-7382
rmanges@nde.unl.edu

HISTORY

Three years ago, the Division of Education at Wayne State College had two Apple IIe computers available for student and faculty use. During the spring semester of 1991, a sixteen station Local Area Network (LAN) lab of IBM compatible computers was installed for student usage. At the same time faculty offices were being equipped with similar equipment. As we approach 1994, the Division of Education now has upgraded the sixteen station lab which is networked, providing software access for students, with plans for expansion. Each faculty member in the Division of Education now has a computer in their office which is networked providing software access and access to students and other faculty.

FOCUS

This presentation will be concerned with exploring changing directions in the preparation of teachers and administrators. It will also address the incorporation of computer networking capabilities related to scholarship, research and decisionmaking at both the undergraduate and graduate levels of professional education.

PRESENTERS

The symposium will be conducted by two faculty members who have had extensive experience with various aspects of the computer network. Dr. Larry Harris, Head of the Division of Education, will present information about acquisition of computer technology, the growth of the computer network in the Division of Education, an overview of the educational usage of the LAN, and administrative functions of the computer in a teacher education program. Dr. Charles Manges, Assistant Professor of School Administration and Secondary Education, will present information concerning the use of computers in the school administration program.

The presentations will encompass only a portion of the symposium. Computer demonstrations will be provided by to enhance presentations. The audience will be encouraged to generate additional ideas and suggestions related to the use of computer technology in teacher education programs. Because others are deeply involved in this area, it is hoped that divergent points of view can be generated. The audience will have the opportunity to experience working with the LAN through an interactive means.

NEW TEACHING METHODOLOGIES

Recent research (Playko and Daresh, 1990) has suggested that administrator preparation programs, based exclusively on traditional instruction in university courses, are not very effective. Individuals progressing through innovative programs providing simulations, case studies and greater opportunities for reflection and collaborative activities should be better prepared to deal with the complex issues facing future educational leaders. Computer networking capabilities allow alternative delivery systems for instruction and assessment. They also allow access to current educational software which is available in the local school districts.

Students are assigned collaborative projects (preparing case studies, group simulations, and group presentations) and are challenged to create simulations (school district funding alternatives, enrollment projections and salary calculations) which are applied to assist local districts. With the aid of these projects the instructor is free to extend learning beyond the regular classroom into the "real world" environment. With the incorporation of internet, the instructor also may serve groups in a mentoring capacity both during and outside of regularly scheduled class time.

One particular example of using the network to create new teaching methodologies is evident through a course in statistics. In the past, statistics courses have largely centered around computational skills. With the inclusion of spreadsheet applications (Excell 4.0) and statistical packages (SPSSPC) on the network, the instruction can concentrate on the interpretation of statistical functions for decision making purposes. This shift in emphasis has, and will greatly enhance the importance of statistics for school administrators.

CD ROM AND LIBRARY AUTOMATION APPLICATIONS

The acquisition of the Local Area Network has provided faculty and students with access to a variety of databases and other resources for research. A CD Rom tower arrangement, which includes ERIC, Academic Abstracts, Governmental Periodicals, CARL, and Business Periodicals can be accessed from each computer station. Thus, students and faculty have the opportunity to search these databases for both current and classical information on a myriad of topics. In addition to the CD Rom access, students and faculty can search the holdings of the Wayne State College Library, other libraries in the state college system, and through Internet, and libraries in other states and around the world.

The ability to access information in a timely fashion has greatly enhanced the capabilities of both faculty and students to conduct meaningful research. The access of the CD Rom material is identical to stand alone systems. With this service available we have effectively moved from one workstation per CD Rom to multiple workstations.

ADMINISTRATIVE USAGE

While the Local Area Network was originally installed to provide the opportunity for students to develop computer usage skills and literacy, it has also provided needed services for faculty development. The completion of a variety of administrative tasks has been enhanced greatly through the usage of the network. The ability to maintain divisional budgets through the use of spreadsheet applications is but one component. The scheduling of courses has been greatly streamlined through the use of spreadsheets and databases. With the acquisition of on campus Email communication needs will be enhanced. Faculty are currently sending written material to students through the network. The Email will allow communication between faculty to occur more efficiently.

SIMULATIONS

The simulations will encompass only a portion of the presentation. The audience will be encouraged to discuss additional ideas and suggestions related to the use of computer technology in teacher and administrator education.

This symposium should acquaint those currently using computer technology in teacher education with additional methods for incorporating this. For those who have not begun using computer technology in their programs, this symposium will provide an opportunity to observe first hand the types of learning which can be achieved through this. Administrators (Deans and others) will have the opportunity to discuss the LAN usage by administrators and methods for encouraging faculty to become involved in the use of computers for instruction.

REFERENCES

Playko, Marsha, & Daresh, John. (1990). The Journey toward Educational Leadership: Reflective Voyages of Four Teachers. Paper presented at the Annual Meeting of the American Educational Research Association Boston, MA

Computer Usage in an Academic Enrichment Center

Mabel Jones Matthews
Division of Business
Saint Augustine's College
Raleigh NC 27610-2298
919-516-4061

INTRODUCTION

The purpose of this paper is to present a rationale for implementing a Computer Academic Enrichment Program in a Division of Business at a small, four year, liberal arts college. The model for this Center is designed with a three-folded purpose: (1) to provide business majors enhanced computer-based instruction, (2) to respond to the need for supplemental instruction, and (3) to enhance academic enrichment.

The Program design was developed following a ten year developmental period where funding was received to increase the computer literacy of the Division's student and faculty, to create two computer related curriculum (Computer Information Systems and Office Administration), and to establish a computer laboratory. During this period considerable curriculum development and program revision occurred along with the expansion and enhancement of computer hardware and software that yield increased computer literacy in the Division of Business.

However, the growth of the Division in terms of enrollment in computer-related courses; a need to respond to student performance and the attrition rate in "high risk" courses; the need to respond to an increased number of under-prepared students; the need for the use of more creative and innovative instructional strategies; the need to provide opportunities for business students to develop competence through practical experience, mentor/mentee relationships, and appropriate assimilation of workforce role models, each suggest a need for expansion of computing academic support. For this reason, the Computing Academic Enrichment Program is designed with the following components:

1. Computing Academic Enrichment Center
2. Computer Instruction
3. Peer Teaching
4. Computer Assisted Instruction
5. Learning Resources Center
6. Business Alumni Network (BAN) and Alumni Speakers Bureau.

An illustration of the Computing Academic Enrichment Program will be presented, keeping in mind that the design of the Program is based on an evolutionary approach to the Academic Enrichment Cycle. That is, it is not a fixed model or finished product. Planned change is to be expected through the five year developmental cycle. This model serves as an example of how a small higher education institution, with limited funding might utilize computer resources in "academic instruction" and "academic support services" to enrich the learning and teaching experience in the Division of Business.

CONCEPTUAL FRAMEWORK

Students enter college today highly effected by modern technology. Ruby (1983) states that computers touch our lives daily. In the home, microwave ovens, electronic heating systems, electronic security systems, and video games are commonplace. In factories, robots are performing many routine jobs for workers. In the office, word processing applications, can edit, delete, format, save, and transfer files from one disk to another, from one Office to another, from one coast to the other, and even around the world. From the teller in the bank to the ticket seller at a basketball game to the checkout counters at the local supermarket computer based products and services permeate our daily lives. Small higher education institutions cannot afford to ignore the revolutionary impact of the computer; rather we must explore opportunities for integration of computer training and use across disciplines.

Students receive training in the use of computers as early as the primary grades therefore, they come to college accustomed to and expecting the drill-practice, tutorial, and problem-solving capabilities of the computer. It is then only natural that a center designed to enrich teaching and learning should use computers to enhance traditional techniques.

This paper draws from the concept of computer-assisted instruction and peer teaching. These well researched tenets provide a foundation for the development of a computing academic enrichment program.

COMPUTER-ASSISTED INSTRUCTION

Computer-Assisted Instruction (CAI) draws from three themes in learning theory: individualization, behavioral objectives, and educational technology. Liao (1992) provides a comprehensive look, through a meta-analysis, of CAI. Many CAI programs share the following characteristics:

(a) they store a series of experiences, often providing alternative learning paths for individuals
(b) they offer independent pacing for individuals
(c) they give the users controlled, contingent reinforcement
(d) they evaluate performance quickly and accurately to provide data on the degree of mastery.

Liao concludes that CAI is more effective than conventional instruction for increasing students' achievement. Since fostering students' cognitive abilities (e.g., planning skills, reasoning skills, logical-thinking skills, critical-thinking skills, skills for transferring concepts, and problem-solving skills) is usually part of our educational goals, the result indicates that CAI is a mildly effective approach for teaching cognitive skills in the classroom setting.

With the increasing use of microcomputers in business, there has been a high demand for college graduates who have hands-on experience in business computers applications. In addition, the American Assembly of Collegiate Schools of Business (AACSB) recognized the need for CAI in business programs. Since the early 1980's schools of business have started to experiment with CAI in course instruction and tutorial software, at the same time encouraging faculty to revise many business courses in order to enhance computer literacy (Ahadiat, 1992). Because of its drill-practice application CAI is strongly encouraged for "high risk" business courses — courses in which a high percentage of students have failed, withdrawn or experienced difficulty— such as quantitative analysis and statistics (Bishop, Beilby, & Bowman, 1992), accounting (Kaye & Nicholson, 1992), and economics (Hobbs and Judge, 1992).

PEER TUTORING

According to The New Merriam-Webster Dictionary, tutoring is defined as teaching, guiding, coaching— to provide instruction especially privately. Zaritsky (1989) based upon the research of Moore, Cross, Booher, and Roueche, acknowledges that tutoring is not intended to be the primary means of instruction for students. Instead, it is intended to be supportive or supplementary and is usually aimed at students who are experiencing difficulty with their learning. The United States' most acceptable teaching strategy, the lecture method, is impersonal. Students often have difficulty learning and succeeding in class that use a lecture format. Tutoring, an extremely personalized method of teaching in which instruction occurs either in a one-on-one setting or in small groups, is a critical alternative.

Tutoring, as it is most commonly employed in the United States, uses students to help students. This type of tutoring is called peer tutoring and is generally accepted to be a cost-effective and practical method for reducing attrition and improving academic performance. It appears that because peer tutors are closer in age and situation, they are better able to understand the problems students are experiencing than older professionals. In addition, the students who serve as peer tutors also experience academic enrichment and affective gains, such as improved self confidence (Ross, 1972).

Jeger and Slotnick's 1985 study on computer-based education strengthens the argument for CAI and peer tutoring. This study looked at computer usage in Math and English and found that the experience resulted in student and faculty role changes.

Student Role Changes. Faculty observed the following changes in student role as a consequence of the experience:

* CAI students demonstrated a better conceptual grasp of subject matter.
* CAI induced a more active and participatory role in learning.
* CAI students asked more questions.
* Greater perseverance at math problem-solving and at revising and editing English writing assignments was reflected in many students' working after the lab sessions had ended.
* Student difficulties were often identified sooner than usual and addressed more quickly.
* CAI encouraged more independent work. At the same time, CAI students often worked in pairs, providing peer-based mutual support.
* Peer-teaching led to greater social cohesiveness and created a positive social climate among students.

Faculty Role Changes. The experience led to changes in faculty role as well:

* A greater number of individual interactions with students resulted.
* New ways of explaining important concepts emerged as a consequence of the computer's requirement that problems be broken down into logical segments.
* In the lab setting, the professor or peer tutor tended to become more of a facilitator of learning than an authoritative lecturer.
* Faculty learned a new teaching strategy, CAI.

* Faculty felt a sense of rejuvenation about teaching mathematics and writing.
* Faculty acquired some programming skills.
* Faculty achieved visibility and gained recognition on campus for computer-related activities.
* Facility collaborated with colleagues from other disciplines.

Jerger and Slotnick concluded that because faculty are the primary mediators of student learning, their new professional development will, in the long run, contribute to improved student learning. Such faculty development might serve as a catalyzing institutional change for larger-scale adoption of computer-based educational delivery systems. In turn, these new delivery systems should continue to facilitate student learning.

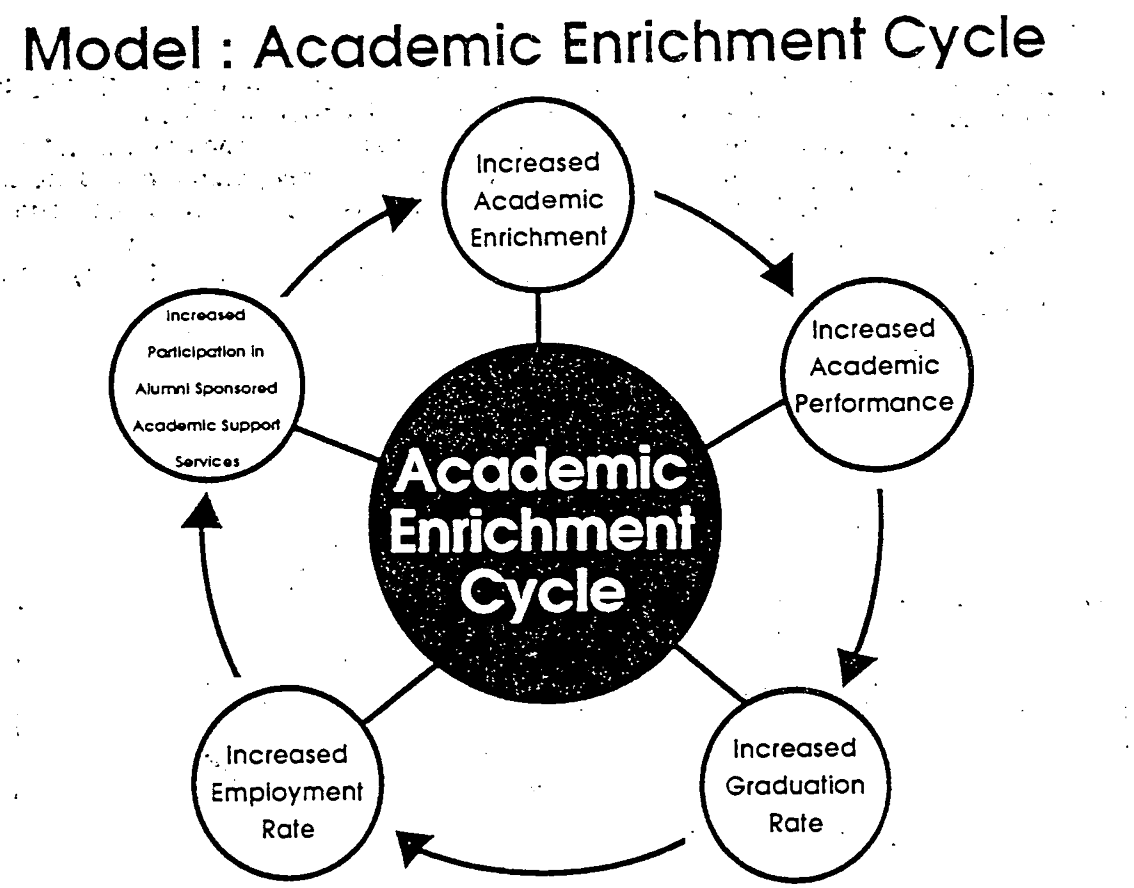
COMPUTING ACADEMIC ENRICHMENT PROGRAM

The Computing Academic Enrichment Program design was developed following a ten year developmental period where funding was received to increase the computer literacy of the Division's student and faculty, create two computer related curriculum (Computer Information Systems and Office Administration), and establish a computer laboratory. Spiraling student and faculty computer literacy was evident in the increased use of the computer laboratory and computer related assignments.

With an acceptable level of realization of the previous "academic computing" goals, new approaches were sought for expanding "academic computing" in the Division of Business beyond the computer courses and word processing applications. The guiding force was to expand the "academic computing" concept throughout the Division and across majors.

The growth of enrollment in computer related courses; a need to respond to student performance and the attrition rate in "high risk"" courses; the need to respond to an increased number of under prepared students; the need for the use of more creative and innovative instructional strategies; the need to provide opportunities for business students to develop competence through practical experience, mentor/mentee relationships, and appropriate assimilation of workforce role models, each suggest a need for expansion of computing academic support. For this reason, the Computing Academic Enrichment Program evolved, based upon the Academic Enrichment Cycle presented in Figure.1.

FIGURE 1.



The first step in developing the CAEC was to create a model of the Academic Enrichment Cycle. It is believed that "increased academic enrichment" yields "increased academic performance", yields "increased graduation" rates, yields "increased employability", yields "increased participation in the Business Alumni Network" and the Network in turn offers services to students that will increase academic enrichment, which initiates a repeat of the cycle.

The next step was to determine how computers might enhance each stage of the cycle. A semester pilot program was implemented. Data was collected from students, peer tutors, faculty, and the Lab Coordinator. Evaluation and analysis of the pilot program data, historical data, and observations created the bases for the Division of Business Computing Academic Enrichment Program.

The components and the key personnel of the Computing Academic Enrichment Program (described in Figure 2) are:

1. Computer Academic Enrichment Center (CAEC). The previously existing computer laboratories, used primarily for computer course instruction activities, were expanded to include software and hardware that is specific to the enrichment and tutorial needs of each discipline. When fully staffed the key personnel will include a CAEC Coordinator, Network Administrator, two part-time Lab Assistants, and 14 Student Lab Assistants (Peer Tutors). The Center's hours of operation were expanded to 11:00 p.m., Monday-Thursday, 10:00 a.m.-3:00 p.m. Saturday, and 1:00 p.m.-5:00 pm. Sunday.

FIGURE 2.

COMPUTING ACADEMIC ENRICHMENT PROGRAM

PROGRAM COMPONENTS

- Computing Academic Enrichment Center (CAEC)
- Computer Instruction
- Peer Tutoring
- Computer Assisted Instruction
- Learning Resource Center (LRC)
- Business Alumni Network (BAN)

KEY PERSONNEL

- CAEC Coordinator
- Network Administrator
- Lab Assistants
- Peer Tutors
- LRC Coordinator

Mabel Jones Matthews (1993)

The Center's computer content is as follows:

Network Architecture:
- 386/25 MHz EISA w/64K CACHE System
- 300 MB Hard Drive
- 1.2 MB Floppy Drive
- Intel Ether LAN Adapter Card
- Novell Netware 2.2/100 Users
- DOS 5.0

Hardware:
- <u>54 IBM/IBM Compatible Computers</u>
- 386-Monitor/VGA (quantity 37)
- 286-Monitor/Monochrome (Quantity 17)
- <u>3 Printers</u>
- HP4 Laserjet
- Epson LQ 2550
- IBM Proprinter 24P
- <u>Scanner</u>
- Panasonic Image Scanner

Software:
- <u>Instructional</u>
  - a. Programming Languages (Compilers)
    - FORTRAN V5.0
    - COBOL V3.0
    - PASCAL V4.0
    - Assembly
    - C++ V3.0
    - SAS
  - b. Application
    - Windows V3.1
    - Word for Windows V3.1
    - Lotus for Windows V1.0
    - Lotus 1-2-3 V2.2
    - EXCEL V3.0
    - Wordperfect V5.1
    - PageMaker V4.0
    - Ventura V3.0
    - Dbase Plus V3.0
    - Dbase V4.0
    - Briefcase Tools
    - Grammatik V4.0
  - c. Utilities
    - Norton V4.0
    - XTREE PRO
    - ViruScan V8.4

Computer Assisted Instruction
- Lotus Tutorial
- Ventura Tutorial
- Typing Tutor
- CPA Review
- LAMBER
- Studyware for GMAT
- Studyware for GRE
- DOS Tutorial
- Dbase Tutorial
- Introduction to Algebra
- Introduction to Calculus
- Accounting Tutorial
- Studyware for Calculus
- Studyware for Statistics
- Studyware for Economics

2. Computer Instruction. Four faculty members provide instruction in academic computing courses. Faculty members are encouraged to have formal meetings, that include the CAEC's Coordinator and Network Administrator, to evaluate academic computing in the Division. The faculty also conducts in-service computer training for faculty and staff and assist non-computer faculty with implementing computer application in their instruction. Instruction is offered in the following computer courses:

   Computer Concepts
   - MicroComputer Software Application I
   - MicroComputer Software Application II
   - COBOL
   - FORTRAN
   - ASSEMBLY
   - Programming in C
   - Operating Systems & Computer Architecture
   - Data Structures
   - Database Management
   - Data Communications & Networks
   - Management Information Systems
   - Systems Analysis & Design
   - Computer Information Systems

3. Peer Tutoring. The peer tutoring component was designed to offer assistance to students experiencing difficulty in what is termed "high risk" courses, those courses in which a high percentage of students have failed; withdrawn or experienced difficulty. Student Lab Assistance (SLA) are selected to perform peer tutoring and are assigned to provide supplemental instruction for these courses. Peer instruction may take the form of a scheduled study or review session, drop-in individual or small group tutoring, or one-to-one scheduled tutoring,. The SLA works closely with the faculty member providing the course instruction and serves as a liaison between student and teacher and is able to alert the faculty member of instructional problems.

While SLAs must have successfully completed the course they peer teach. they are encouraged to audit the class of the faculty member to whom they have been assigned to better understand instruction offered and course expectations. SLAs generally have strong computer backgrounds. but if they do not. they are required to complete computer workshops. SLAs must become familiar with varied software and how they are applied in courses. Students are also, required to complete the "How to Tutor Workshop" sponsored by the English Department. These paid SLAs work 12-15 hours weekly and provide services in the CAEC or the Learning Resource Center (LRC). At the end of the academic year special recognition is given to SLAs during the Division's awards banquet.

4. Computer Assisted Instruction (CAI). CAI programs. especially those with tutorial components are selected by faculty and the Center Coordinator. Software that accompanies textbooks is also assigned by faculty and used by the students. Although drill and practice CAI programs may strengthen a student's learning experiences in courses where difficulty is experienced. a greater academic enrichment is realized when CAI enhances communication and discovery-oriented learning opportunities. Because learners have different learning styles. CAI is not dependent upon as a sole source of instructional support. The use of CAI in the CAEC is viewed as a facilitator of learning. CAI in advanced courses (i.e., Intermediate Accounting. Statistics. CPA Review. Technical Writing. Entrepreneurship. Investment and Quantitative Methods) is used as a tool to achieve some high-order thinking skills. such as problem solving through the use of simulation activities.

5. Learning Resource Center (LRC). The LRC provides a library oriented setting where students can study. use learning resources. and obtain peer tutoring. The Center also provides support services to the Division's faculty. The Center is staffed with a full-time Coordinator who operates the Center. schedule SLA assignments. and coordinates Business Alumni Network (BAN) activities. Computerized listings of resources and equipment are available. Students may use the resource on site or check resources. The resources of the Center include newspapers. business journals and magazines. instructional videotapes. equipment and supplies.

6. Business Alumni Network (BAN) and Alumni Speakers Bureau. BAN was developed to enable the Division's alumni to provide academic enrichment opportunities to students by sharing their "expert opinions" with students who are following a professional path similar to their own. An alumni database has been developed and updated regularly to provide a listing of graduates and their professional profiles. A biannual newsletter facilitates communication with alumni. Alumni are invited to serve as speakers at assemblies. during classes. and other special occasions. to participate in career fairs and job recruitment. and to provide tutorial assistance in their areas of expertise.

   A two-way benefit can be derived from the establishment of such a network. While perfecting their craft alumni lend support to their alma mater. In addition. our students gain first hand knowledge from those with whom they share a greater similarity. However. one of the most beneficial aspects of development of such a network is the use of our graduates as mentors and experts from the seven disciplines provided through the Division of Business.

The Speakers Bureau portion of this component provides a forum for dialogue between students and graduates about the workforce in a manner often deemed more realistic by the student than when shared by experts unknown to the student. BAN serves as an alumni professional liaison between our students and the workforce, while providing our students with what is felt to be one of the most appropriate types of role models and an exceptional opportunities for academic enrichment.

Computing Academic Enrichment Program
Five-Year Plan (1992-1997)

This program is currently operating under a five-year plan. The goals in bold face type have been either accomplished or are in progress.

Marketing

Goal 1. To market Computing Academic Enrichment Program to students, faculty, and administrators in a manner that will remove the stigma attached to tutorial services.

Method: Through the employment of the Academic Enrichment Cycle users will see "enrichment" as a tool for enhancement and growth (i.e. "D" student can become "C", "B" can become "A").

Facilitator: Division Chairperson and Computing Academic Enrichment Program personnel

Goal 2. To increase the use of CAI across majors.

Method: Encourage faulty to implement teaching strategies that use computer assisted instruction and computer assignments and projects.

Facilitator: Division Chairperson and Computing Academic Enrichment Program personnel.

Staff Development

Goal 1. Key Computing Academic Enrichment Program personnel (see Figure 2) will obtain on-going formal computer training in areas related to their responsibilities. Personnel will be encouraged to cross train so that they will have needed skills in times of emergencies and increased workloads.

Methods: Complete computer courses, in-service workshops, and professional conferences.

Facilitator: Higher education institutions, computer companies and professional organizations.

Goal 2. All faculty and staff, across disciplines (Accounting, Business Administration, Business Education, Economics Computer Information Systems, Management, and Office Administration) will be knowledgeable in teaching strategies involving computer applications.

Methods: Provide in-service workshops during staff development.

Facilitator: CAEC Coordinator, CIS faculty, software sales staff, textbook sales staff, and other training sources.

Goal 3. Establish open communication between Computer Academic Enrichment Program personnel and the Division faculty and staff.

Methods: Provide program updates at Divisional meetings, establish formal procedures for requesting services and distribute periodic newsletter.

Facilitator(s): CAEC Coordinator and LRC Coordinator.

Goal 4. Identify "high-risk" courses, those courses in which a high percentage of students have failed, withdrawn or experienced difficulty, and increased faculty use of CAI in those courses.

Methods: Review and design a comparative analysis of students' performance in Division's course offerings and implement teaching strategies in "high risk" courses that employ computer assisted instruction.

Facilitator: Faculty, CAEC Coordinator, Network Administrator

Goal 5. Provide adequate tutoring assistance for students.

Methods: Provide in-service training for tutors; Encourage tutors to enroll in Methods and Materials of Tutoring (ENG 241); Encourage Division of Business Scholars to serve as tutors; Encourage participants in Business Alumni Network to serve as tutors.

Facilitator: Computer Academic Enrichment Program personnel.

Computer Hardware

Goal 1. Upgrade existing (CAEC) hardware and software, including an internal tape backup system, and replace obsolete equipment.

Methods: Examining and evaluating academic enrichment centers at other institutions, inventory existing software access faculty for software needs, access College sources for existing software licenses, extend and upgrade software licenses, purchase new and enhance existing hardware and software, including back-up system, develop software in advanced CIS courses.

Facilitator: CAEC Coordinator, Network Administrator, CIS Faculty, and computer vendors.

Goal 2. Establish a CAEC Administration System that will record students' activities and teacher/student interaction.

Method: Purchase and install a Classroom Administration System.

Facilitator: CAEC Coordinator, Network Administrator, and computer vendors.

Goal 3. Expand existing network to include faculty and staff offices.

Method: Connect faculty and staff to CAEC network.

Facilitator: CAEC Coordinator, Network Administrator, and computer vendors.

Goal 4. Install UNIX hardware and software system to existing network.

Method: Purchase UNIX hardware and software.

Facilitator: CEAC Coordinator, Network Administrator and computer vendor.

Scheduling

Goal 1. To provide accessible hours for CAEC usage.

Method: Expand hours of operation to reach 60 hours per week.

Facilitator: CAEC Coordinator, Network Administrator, and Student Lab Assistants.

Goal 2. To provide accessible hours for LRC usage.

Method: Expand hours of operation to reach 50 hours per week.

Facilitator: CAEC Coordinator, Network Administrator, and Student Lab Assistants.

Procedures

Goal 1. Establish formal standards and procedures for CAEC.

Method: Develop a handbook for faculty. staff and student use.

Facilitator: CEAC Coordinator.

Goal 2. Establish formal standards and procedures for LRC.

Method: Develop a handbook for faculty, staff and student use.

Facilitator: LRC Coordinator.

Student Involvement and Evaluation

Goal 1. Establish formal method for student feedback.

Method: A program-developed survey form.

Facilitator: CEAC Coordinator and LRC Coordinator.

Goal 2. Establish a database of students using Center.

Method: A student data sheet.

Facilitator: CEAC Coordinator and LRC Coordinator.

Faculty Involvement and Evaluation

Goal 1. Establish formal method for faculty feedback.

Method: A program-developed survey form.

Facilitator: CEAC Coordinator and LRC Coordinator.

Business Alumni Network (BAN) Involvement and Evaluation

Goal 1. Establish formal method for Alumni feedback.

Method: A program-developed survey form, periodic Newsletter, and Alumni database.

Facilitator: LRC Coordinator.

Goal 2. Establish topic and schedules for BAN Speakers Bureau.

Method: Select alumni database.

Facilitator: LRC Coordinator.

Goal 3. Establish BAN mentoring program.

Method: Match students with participating alumni.

Facilitator: LRC Coordinator.

Goal 4. Establish BAN tutoring program.

Method: Access local participating alumni to provide tutorial assistance.

Facilitator: LRC Coordinator.

Summary

The cry for academic enrichment was heard from those who were experiencing difficulty in "high risk" courses and those who were seeking more challenging academic experiences— from those who appeared to enter college under-prepared and those who entered with strong academic preparation. The modern technology of computer was viewed as a possible bridge between the two types of academic enrichment experiences, supplemental and enhancement, needed. Business faculty wiliness to explore computer-related instructional strategies, high achieving business students wiliness to serve as peer tutors in the Computing Academic Enrichment Center, and business alumni wiliness to be mentors and to share their expertise, all lead to opportunities for effective academic enrichment in the Division of Business.

The design of the Computing Academic Enrichment Program is evolutionary and serves as an example of how a small higher education institution, with limited resources, might utilize computer resources in "academic instruction" and "academic support services" to enrich the learning and teaching experience. Further research and development will continue throughout the five year developmental period. The researcher proposes to present findings at the end of years three and five.

References

Ahadiat, N. 1992. A Study of Work Values of Computer Education in Accounting. Computer Education. 19:131-143.

Bishop, P., M. Beilby, and A. Bowman. 1992. Computer-Based Learning in Mathematics and Statistics. Computer Education. 19:131-143.

Hobbs, P. and G. Judge. 1992. Computers As A Tool For Teaching Economics. Computer Education. 19:67-72.

Jeger, A. M. and R.S. Slotnick. 1985. Toward A Multi-Paradigmatic Approach To Evaluation of CAI: Experiences From the NYIT Computer-Based Education Project. Paper prepared for presentation to the NYIT Invitational Seminar on Computer-Based Education, deSeversky Conference Center, Old Westburg, NY.

Kaye, G. R. and H. S. Nicholson. 1992. An Educational Framework For Information Technology in Accounting and Management Education. Computer Education. 19:105-112.

Liao, Y. K. 1992. Effects of Computer-Assisted Instruction on Cognitive Outcomes: A Meta-Analysis. Journal of Research on Computing in Education. 24:367-379.

Ross, S. F. 1972. A Study to Determine the effect of Peer Tutoring on the Reading Efficiency and Self Concept of Disadvantaged Community College Freshmen: Final Report. Washington, DC: National Center for Educational Research and Development. (ERIC Document Reproduction Service No. ED 081415).

Ruby, R. 1983. Increasing Use of Microcomputers in Business Education. Paper presented at the National Association of Teacher Educators in Business Education, American Vocational Association Convention, Anaheim, CA.

Zaritsky, J.S. 1989. Peer Tutoring: Issues and Concerns, Results of a Survey. (ERIC Document Reproduction Service No. ED 315134).

Moving to Client/Server Computing: How Do You Get There?

David K. Moldoff
President, Applied Business Technologies, Inc.
4631 West Chester Pike
Newtown Square PA 19073
215-359-0700

The subject of this paper is client/server technology and how to move to client/server networks. Topics covered include a background on client/server, how client/server works, the advantages of client/server technology, the issues and, most importantly, the steps to building an effective client/server environment.

Since nearly every college and university is computerized already, this paper assumes that most institutions considering a move to client/server will do so as a rightsizing (moving up from stand-alone systems, or moving down in size from mainframes or minis) or updating measure. They will be eliminating a minicomputer system and replacing it with the components of client/server technology. In the corporate world, companies, organizations, entire industries have scrapped old mainframes and minis and replaced them with client/server-based solutions. First, a definition; what is client/server? The "client" is the user; in hardware terms, it's the workstation, the place where data entry occurs, reports run, and hardware interfaces with human beings. The "server" is the file server(s), which can be a very powerful PC, or a mini converted to file server use. The file server contains the database, the network software, and the application software. It is what allows a user or client in one office to access the same information as a user in a different office. The client uses and the server provides services to the client.

BACKGROUND-COMPUTING IN THE 90's & BEYOND

Why are so many corporations, businesses, and colleges downsizing or rightsizing? There are several clear reasons, (all revolving around that ever-important "bottom line").

- Reduced maintenance costs
- Lower upgrade costs
- Greater application software selection
- Simpler new application development
- Reduced MIS backlog for servers
- "Enabled" end-users (greater efficiency)
- "Enhanced" analysis capabilities

It is already clear to many Fortune 1000 companies that computing in the 90's is radically different from computing in past decades. Pressures of three kinds have driven an evolutionary movement toward downsizing and to implementing open systems:

- Business
- EndUser
- Technical

Because of the advances in technology in recent years, vast amounts of data have become readily available to users. This information can help us make better, quicker decisions. The challenge we face in the '90s is how to manage information, and specifically, how to make it available to the people who need it when they need it, keep it secure, and keep it current. As we work to consider the computing needs prevalent today, and to anticipate the needs of tomorrow, we must consider several concerns:

ACCESSIBILITY TO DATA AND SYSTEM RESOURCES

Users are concerned with doing their jobs. They want to access data when they need it. They want access to applications that will help them do their jobs more efficiently. They don't care about the technology as long as they can do their jobs without technological barriers, i.e., system failures or inability to access data.

ADMINISTRATION AND CONTROL

Systems also must provide top-level administrators with the information they need in a timely manner. However, as a campus' computing needs grow, so too will the network. The system chosen must be easy to administer and adaptable or able to incorporate new technology. The system must also provide the flexibility to meet the varying needs of different departments.

SECURITY & RELIABILITY

In client/server systems, departments across campus share the same information base -- which has its many benefits. However, valuable data must be secure enough (from loss, theft and damage) to allow network administrators and management to sleep at night. Today's networks must provide administrators the ability to control security and access, and have enough built-in safeguards to minimize downtime and loss of data in case of system failure. System administrators of client/server networks can limit access down to the field level to ensure the security of sensitive data.

AVAILABILITY OF APPLICATIONS

One of the motivating factors for not downsizing involves the availability of applications commonly used in mainframe environments. This valid concern has been minimized in recent years as software developers have recognized the trend toward downsizing and rewritten applications specifically for network environments. Because of the client/server architecture's easy user interface, security, reliability, and administrate ability, many developers have migrated to PC networks.

BUSINESS PRESSURES

- Cost Control and Reduction
- Productivity
- Adaptability to Change
- Competitive Advantage
- Information/Data
- Better Analytical Tools

COST CONTROL AND REDUCTION

More than any factor in the downsizing trend, business pressures drive the evolution to smaller, more flexible client/server systems. Competition for enrollment among colleges and universities is more fierce than ever, now that the population bulge of the baby boom years has passed. There are fewer students to go around and less money coming in from tuition. University management now requires a better return on their computing investment and a general reduction in costs from information systems. While deploying a LAN-based client/server system on a campus-wide basis involves start-up costs, these costs are offset by the elimination of excessive processing expenditures including maintenance that can run in the tens of thousands of dollars annually, and the expense of system-time. By migrating to client/server, capital dedicated to inefficient technology is freed.

> Typically, existing mainframe-based production systems (commonly called "legacy" systems) are not replaced by the new LAN-based subsystems. They are simply integrated and extended at the LAN server and workstation. Quite often this results in transactions with substantially fewer errors, making host processing more efficient. All of which means higher return on investment (ROI).1

Effective systems also enable an organization to maximize the return on more of their activities and other investments (i.e., not only can you cut or control expenses, you can be more effective with your income producing activities).

PRODUCTIVITY

The people who use the computer system cannot be productive if it regularly fails, is difficult to use and understand, or is slow and unresponsive. Because client/server networks have such easy-to-use interfaces as the Microsoft Windows Graphical User Interface (GUI), the impact on user productivity can be profound.

> It is the GUI that makes computer usage truly easy. It is the GUI that sharply reduces user training in applications. It is the GUI that increases productivity and satisfaction. It is the GUI that can be used to mask application and system changes from the user, which can be critical as applications are in various
>
> stages of downsizing. It is the GUI that reduces overall support and training

costs. It is the GUI that can provide a consistent, virtually seamless interface across multiple applications and services.

In client/server computing, the GUI affects productivity by making computing less complex, easier to understand, and more attractive to the eye. The GUI also serves to impose easy-to-learn standards on the end user. Different applications, even those with very dissimilar purposes or functions, work the same way, thus reducing the user learning curve.

User productivity, in a client/server architecture with a GUI can greatly improve.

ADAPTABILITY TO CHANGE

How well a system adapts to changing technology tells volumes about how well that architecture will serve an institution in the coming years. In the mainframe and mini platforms, upgrading is no simple matter. In the past, to take advantage of new technology, businesses and colleges had to scrap their existing system and purchase an entirely new one. The pressure business offices feel today, and also exert on data processing centers, is to get more mileage out of their investment in computer systems. A system unadaptable to change, by definition, will have a lower return on the initial investment. In contrast, client/server systems take advantage of PC and Network architecture, whose developers in turn have recognized the need for adaptability to future technology and incorporated the ability to change within the system architecture. For example, come with open sockets for upgrading to the next generation of processors. As new advances in network hardware hit the market, these changes are more easily and inexpensively incorporated into your existing base of PCs and file servers.

COMPETITIVE ADVANTAGE

Colleges and universities compete the same as corporations compete. Educational institutions compete for market share in the form of enrollment, for research grants from government agencies, and for endowments and grants from corporations. A cost-effective, efficient computer system that enhances user productivity can help a college be more competitive by reducing costs and making the college more attractive to its students (who wants to stand in line for two hours to register for classes because of an ineffective system?). In addition to cost savings, the access to data offered by client/server systems as well as their security, reliability, and flexibility, can help give colleges a competitive edge.

INFORMATION/DATA

Need for information by administrators should closely parallel, if not match, the computer systems ability to provide access to information. Lack of timely, reliable information puts pressure on all levels within a department. For example, suppose the Vice President of Finance needs to see all revenues and expenditures for a particular department, but cannot access the data he needs or cannot get it in the form he needs it. The domino effect will occur as he applies pressure to the assistant to "Get me that report!" By the time he gets the report, scores of data entry, computer center, and departmental personnel could be affected.

BETTER ANALYTICAL TOOLS

Formless data doesn't help anyone make informed and more qualified decisions. Data becomes information by appearing in a useful format. Client/server architecture with its ability to utilize multiple databases and operating systems allows administrators and management to use their system as a key analytical tool, helping in the decisionmaking process.

END-USER PRESSURES

End-users, despite being an important link in the data processing cycle, will make themselves heard when unhappy with the computer system. How many times have computer center directors heard users scream: "The computer lost my data!" "I can't get into my directory." "The computer screwed up!"? "Where's the 'Any Key?'" Computers can provide a convenient scapegoat for End-users, often taking the blame for any loss in productivity, tardiness of reports, long lines, or lost data.

ACCESS TO INFORMATION

Client/server systems, with its shared databases, give users the access to information they need to perform their job. Give users access to the information they need to do their specialized jobs and you can help keep them happy and productive.

RESPONSIVENESS TO NEEDS

End-users have requirements. As the "client" in the client/server network, users demand that the computer system do everything short of data entry. They need a system capable of handling their word processing, data entry, and student processing needs. Client/server systems provide the flexibility required to meet the wide variety of user needs by providing a development environment in which the vast majority of software developers can write their products.

PERFORMANCE

As technology marches on, user perceptions of performance change dramatically. They go from "Oh, wow! That's fast!" To, "This report takes longer and longer to process." To, "This system is awful." Their perceptions follow technological advances pretty closely, especially when the person next to them gets a faster PC or a better network card. Client/server architecture, fortunately, allows for the easy, relatively inexpensive replacement of components to keep current with user performance requirements. You can upgrade hardware in the most critical areas and redistribute from there.

EASE-OF-USE

Systems have to be easy to use or users won't use them. It sounds simple enough, but it's true. The benefits of an easy-to-use computer with easy-to-use applications are diverse.

These include a redundant, shorter training period, greater productivity, and happy users who actually enjoy using the computer.

TECHNICAL PRESSURES

The computing industry changes so often and so quickly that institutions, like corporations, feel pressure to somehow keep current with technology. This urgency can hamper the decisionmaking process when institutions finally are ready to rightsize. Which technology will be the easiest to maintain and keep current? On which platform and in which environment will the best applications be written? What are the other guys doing? The primary technical issues concern:

- New Advances in Computing
- Increasing Pace of Developments
- Need to Stay Current
- Creeping Obsolescence
- Electronic Society Worldwide Network

DOWNSIZING

Downsizing is a natural evolution in computing and has been happening since the beginning of computing. The first working computer, Eniac, at the University of Pennsylvania in Philadelphia occupied several rooms and was capable of only the most rudimentary mathematical processes. As computers gained a foothold, they eventually occupied a more manageable size. When vacuum tubes were replaced by transistorized components, the size of the computer shrank even further. Today, a desktop personal computer has more processing power than many minicomputers and mainframes of the '70s and '80s. Therefore, the computing industry has moved from its inception to downsizing.

Today downsizing has new meaning. It describes the shift from centralized mainframes and minicomputers to enterprise-wide client/server PC Networks. Several technological forces and advances drive the crossover. These are:

- Local area networks (LAN)
- Client/server architecture
- High-performance file server platforms
- GUI's
- Local area network/wide area network (LAN/WAN) gateways
- Widespread application software

These technologies have been proven and their potential advantages have been demonstrated through the use of prototypes and pilot projects, as well as through the success of end users who have exploited these technologies in full blown production systems. In particular, the advances in server technology and LAN operating systems have been extraordinary....

The advantages and benefits realized in the corporate world have been well documented by the business media.

> In August 1991, the Turner Corp. said goodbye to its IBM 4341 mainframe computer. One of the nation's largest construction companies, Turner was then moving to a new headquarters...and began running its 30city, 3,000 employee business entirely on a network of personal computers. ...Turner's chief information officer, Richard A. Schell, looked forward to cutting the company's annual computer budget in half to $2.5 million.

Turner Corp. did achieve its goals, but not without a little bit of turbulence along the way.

> "There was a tremendous resistance to the PC technology," Schell said. The mainframe jocks had good reason: By the time the mainframe was phased out, 60% of the staff was let go -- a big factor in achieving Schell's goal of cutting his $5 million budget in half.

Initially, corporations and institutions were reluctant to downsize because of the lack of the type of powerful application software available on mainframes. However, the software industry has worked hard to catch up, and the abundance of comprehensive application software for client/server systems has fueled the move to downsize. The most sought applications include accounting, budgeting, and project management, all processing and transaction-heavy programs, which PC networks now handle. The savings in software and software maintenance costs alone have helped spur the switch.

Networks and High-performance hardware, at a fraction of the cost of a mainframe, have made it difficult for many IS departments to ignore client/server architecture. The price of a new network server and workstations is so much less than that of a mainframe with similar capabilities that organizations find they can afford to make the switch. And, because upgrading PCs involves replacing individual components instead of entire workstations or servers, institutions can keep their hardware up to date with a reasonable and manageable budget.

The abundance of end-user environments (i.e., Windows) and tools also make client/server attractive to more and more institutions. With GUI present on a majority of client PCs, user-friendly applications and tools help to boost productivity as well as user enthusiasm for computing.

CLIENT/SERVER FUNCTIONS

ANOTHER DEFINITION OF THE CLIENT/SERVER RELATIONSHIP

The primary purpose of any network is to provide certain services to the users of the network. A computer that provides a service is called a server, and a computer that uses a service is called a client. The term client/server refers to the interaction between two computers, in which one computer provides services to another. Some networks employ a single dedicated server, while other networks use multiple dedicated servers. These multiple-server networks are often referred to as distributed networks because the server load is distributed among several servers.

CLIENT/WORKSTATION FUNCTIONS

A client/workstation works much the in same way as a stand-alone PC does. For the most part, the interaction between the client and the server is transparent to the user. "Client programs represent the portion of the total application that allows the user to interact with the computer. This typically consists of a user interface that can start the application, provide input, and display results."[7] A GUI is usually selected as the client interface for these reasons:

- GUI's consistent appearance
- The ability to hide the complexity of the application from the user; ease-of-use
- Users can usually continue to use applications with which they are already familiar and do not need to learn an entirely new set of programs. The user interface, in short, can help the user's productivity as well as his adaptability to new applications.

The client/workstation is the place where all the manual work is performed. Users enter data, run reports, format documents, print letters and labels, and edit spreadsheets. The system administrator controls the network from a workstation, and a high-level administrator uses the workstation to analyze information. The workstation interfaces directly with its human users and provides the access to applications and data necessary for the user to be productive. The workstation sends and receives data in the form of transactions to and from the server, where that data is stored for later use. To the user, however, data retrieved from the server appears the same as data retrieved from a local hard drive; the entire process, again, is transparent to the users.

SERVER FUNCTIONS

Applications shared by multiple users use the server as a kind of home base, both for the application's operating software and for the database. The most process-or-intensive work is performed by the file server. The client/workstation handles the user interface and some other application-specific processing. Because of the dramatic difference in the price of processing power on a PC vs. a mini or mainframe, this arrangement makes a very attractive price/performance ratio.

The server allows users at client/workstations access to applications and data campus-wide. With valuable information shared so universally, data integrity and security are key issues. The system administrator has control over security rights for individuals using the system. The user log-in and password determine which data and applications the user can access (no matter where he logs in). If the user does not have rights to a database, a directory or a network drive, he will not even see the directory or drive when searching the network. The network also allows for easy data backup, and redundant systems and unlimited power supplies can protect even work in progress.

While the client application interface is familiar to PC users, the server half to a client/server application has a different purpose. On the back end, server applications perform process-or-intensive operations, greatly improving networking processing power and performance compared to the older resource-sharing model. By exploiting server intelligence, LANs also enjoy more controlled data management, better security, and easier administration. Typical server applications include database or communications services. Both of these services often have considerable processing that occurs at the server.

KEY BENEFITS/ADVANTAGES OF CLIENT/SERVER NETWORKS

REDUCED SYSTEM COSTS

As discussed earlier, wherever client/server networks have replaced centralized mini and mainframe systems, organizations have realized significant savings in the following expense categories:

- Hardware
- Applications
- Development of new applications
- Maintenance, both hardware and software
- Personnel
- Organizational
- Overhead

MORE EFFICIENT OPERATIONS & INCREASED PRODUCTIVITY

The switch to client/server technology has also led to more efficient operations on a system-wide basis. Efficiency is a product of the GUI, the improved responsiveness of networks, and access to information at all levels. As more and more users use their workstations to perform more of their job duties, their efficiency and effectiveness also can improve.

TIMELY INFORMATION

With the shared databases of client/server applications, users at all levels can more easily have access to the information they need when they need it. Good timely information, in a useful format, with flexible analytical and reporting tools, are the keys to effective decisionmaking.

FLEXIBILITY

A system's flexibility affects its ability to meet not only current needs, but future needs as well. Client/server networks offer flexibility to run applications as simple as word processing and spreadsheets to those as complex as database management, budgeting, and operational/administrative software. PC networks have garnered enough attention that the lion's share of applications currently under development are being written specifically to take advantage of client/server architecture.

SCALABILITY

In a scalable system, you don't have to change the application to accommodate more users or larger processing volumes. They can be handled by adding stations to the network, increasing the capacity of the server, or by adding more servers. Client/server, with its foundation of open standards, ensures that you will have a wide range of options for dealing with any future problems. Applications and systems which are based on client/server technology are inherently scalable. As newer faster technologies come along, they can more easily be swapped for existing components. This applies to everything from the workstations to the data base manager.

STEPS TO BUILDING AN EFFECTIVE CLIENT/SERVER ENVIRONMENT

What follows is a fairly complete high-level set of steps which, if followed, will help you avoid most of the common mistakes in moving to a new strategic architecture. Those steps are:

- Inventory your Existing Components
- Perform an Operational Audit and Needs Analysis
- Identify your Critical Applications
- Develop a Long-term Strategic Plan
- Select the Component Technologies
- Develop an Implementation Plan

Below we examine each of these steps in more detail.

INVENTORY EXISTING COMPONENTS

Identify what components you have, their current importance and their effectiveness. These components include not only hardware but the applications (big and small) as well as current and historical data. The success of your move to a client/server environment will depend greatly on how well you integrate or migrate current applications and data bases. Those applications which perform truly valuable functions must be accommodated by any new environment. Some critical functions will usually be driving your organization to move to client/server. Those functions will form the basis of the new environment. They may only need a more cost-effective platform or those critical functions may be inadequately handled by the current systems and require replacement.

As you get a handle on all of the existing components you will also be gathering other information about your organization which will prove invaluable as you proceed. This is usually an eye-opener, and tends to uncover more systems and functions than any one person was aware of. Be sure not to overlook the informal systems like manually-kept spreadsheets or even paper based systems.

As you proceed, you will determine which pieces to replace, which to integrate, which to migrate, and which to discard. These decisions will drive the rest of the process as you move to a client/server environment.

OPERATIONAL AUDIT AND NEEDS ANALYSIS

As discussed previously, no transition in technology is entirely simple or painless. Organizations should carefully plan every step of the downsizing operation. Applied Business Technologies, Inc., identifies the issues affecting six important and different aspects of each department campus-wide:

- People
- Policies
- Procedures
- Program
- Process
- Purpose

Each of these areas must be addressed in any system implementation on a variety of levels:

- Institutional
- Management
- Operational
- Support
- Technical
- User

When performing a needs analysis, start with the people who use the system: what do they require? What do the users want? Are their requests reasonable and feasible within the framework of your IS budget? Compare users' needs with the applications currently available and make sure you have a match, or at least a compromise that everyone comfortably can live with.

Next, your institution will need to examine its overall purpose in acquiring and implementing client/server architecture. What are the goals? What does Management expect of a new system? What are the business pressures driving the decision?

After determining user needs and institution-wide goals, current policies, procedures and processes must be reexamined in the context of a new system. Do the current policies, procedures and processes currently meet the institution's needs? How will these change with the new system, and how will people at all levels be affected?

Finally, look at the administrative program as a whole, again within the context of a computing system. Does the system work? Can the system be improved? Is the system hardware-dependent, or can it be migrated to work within new client/server environment?

The ramifications of a client/server implementation should not be underestimated. Change of this kind will impact everyone from data entry clerks to top level administration. Before planning a system change, perform a needs analysis, or have a competent outside vendor perform the analysis for you. Usually a third-party can help you develop a more objective picture of your situation.

IDENTIFY CRITICAL APPLICATIONS

Examine your application needs from the varying departments. Determine which requirements will benefit the greatest number of users, which will have the greatest financial impact, and/or which will have the best chance of succeeding (either due to support from the users or due to interest from senior management). You should also perform a cost-benefit analysis on those needs to determine which will provide the best return on investment (ROI).

DEVELOP A LONG-TERM STRATEGIC PLAN

Taking all of the organizational and administrative issues into consideration, you're now ready to develop a strategic plan. A solid long-term strategic plan is extremely important for success. Short-term plans are quickly obsolete and lack sufficient depth to provide lasting impact on an organization. A good strategic plan will integrate the myriad needs and long-term goals of your organization, the projected business conditions of the next 5 to 10 years. These will be balanced with the many limitations or constraints such as available capital, personnel, etc. The end result will be a comprehensive plan which you will come back to again and again as you plow through the large volume of specific decisions yet to be made.

Decisions made in light of the strategic plan will mesh effectively over a prolonged period. Decisions made without a long-term plan will be haphazard and will only yield more and more problems as time goes on and the inconsistencies begin to clash and cause friction. More than likely, you are struggling with the fruits of just such an uncoordinated approach as a result of past actions or inaction by your predecessors.

A long-term plan takes vision to conceive, courage to propose, and persistence to implement, however the benefits are well worth the investment. Once the plan is complete it will require only periodic maintenance every year or so, to keep it current.

SELECT COMPONENT TECHNOLOGIES

In this stage of planning, you evaluate the myriad of platforms and hardware options available. You meet with vendors, go to different sites to see the various configurations, and eventually select and purchase the components that will comprise the client/server network. Fortunately, it is a buyer's market with scores of vendors, developers, and manufacturers offering systems in almost any imaginable configuration. Unfortunately, there are no easy answers to this selection. Institutions must carefully balance the needs and talents of users, departments, managers, and system administrators with the reality of the information systems budget.

The network administrator will be evaluating the several different platforms available to function as the system server. The server can take on several different forms, from one or more powerful PCs to a converted mini or mainframe to a superserver. When selecting the server, again, the impact of future needs should not be minimized. The server should be able to accommodate reasonable growth, which will accompany the network implementation.

The next step involves selecting the workstations, monitors, and workstation operating systems and GUIs. You'll have a good look at the Microsoft Windows GUI, as well as IBM's OS/2. You will need to connect everything so you will need to choose which form the cabling will take, as well as network cards that allow the workstations to communicate with the server.

In general, look for the one that offers you the best balance of price and performance, offers the simplest upgrade path, and best takes advantage of your current resources.

The operating system you choose can make or break your client/server network. If you are starting small, beware of purchasing LAN software designed solely for small systems, unless you're willing to scrap and start over when you expand. Look for the highest standards of security, reliability, flexibility, performance, and interoperability.

You should also pay careful attention to choosing some key vendors who will need to be your partners. This applies to both hardware and software. A good vendor can significantly ease the many burdens of managing a network installation. Which areas you choose to work with vendors on will depend on your organization's strengths and needs. For instance your staff may do your wiring while another organization may choose to go to a vendor for that service.

DEVELOP AN IMPLEMENTATION PLAN

The implementation of the full vision will probably take many years and will constantly be evolving. It should be broken down into manageable stages or phases to help maximize your success. The longer each phase is and the more work to be accomplished, the higher the likelyhood of delays and losing site of the goal. Clear goals or objectives, and shorter moderately ambitious phases will be easier to manage and will provide more focus to the whole plan. Among other things your plan will need to cover the following (in no particular order):

- Budget Allocations for Acquisitions, Maintenance and Ongoing support (Multiyear if necessary)
- Hardware Acquisition Plan (Phased if necessary)
- Network Cabling Plan
- Staffing/Personnel Impact Plan
- Application Migrations
- Acquisition of New Applications
- Conversions to New Applications
- Training
- Internal Support and Handholding
- Backup and Disaster Contingency Plan

OPTIMIZING THE PRICE/PERFORMANCE TRADEOFF

By choosing a client/server network, you have already taken a dramatic step in optimizing the price/performance tradeoff. Your annual maintenance costs will be significantly lower. The number of internal staff required to maintain the system will go down, so those resources can be reallocated as part of the new investment in hardware. You have eliminated your dependence on proprietary mini or mainframe architecture, operating systems, and applications. Now that you have decided to move to client/server, the only decision remaining is how to best take advantage of the new technologies, and which of the new technologies will best serve your needs today and into the future.

Plan for future replacements -- each component has a reasonable life and this should be taken into account.

MAKING A SMOOTH TRANSITION TO CLIENT/SERVER

A major concern facing many colleges when considering a transition to client/server systems involves the fate of "legacy" applications, those which have been in place for an extended period. What happens to those programs after the switch?

> Reengineering is not a complicated process in a client/server environment, however, and managers, application developers and users alike will appreciate the shorter development cycles. In contrast, "mainframe application development lead time can be two years or more.... A direct result of shorter application development cycles is a more streamlined and responsive organization that can deliver information to customers quickly and effectively.

In making a smooth transition, the management team and the system administrator have to work hard to address all the major concerns, particularly those regarding access to data, cost savings, and overall price/performance benefits. Users may require a certain amount of handholding, especially early on when resistance to change can run very high.

SUMMARY

Client/server has many benefits as we have seen, and the pieces that exist are sufficiently proven. The question is not whether to go to client/server technology, but when and how. The sooner you begin to plan and implement, the sooner you will reap the benefits.

BIBLIOGRAPHY AND SOURCES CITED

Boyce, Jim, "LANs: Making the Right Connections," Windows Magazine, January 1993: 113-118.

Catchings, Bill, and Van Name, Mark L., "What's the Right Size?," Corporate Computing, June/July 1992: 86

Client/Server Computing, Microsoft Corporation, 1991.

Downsizing Corporate Information Systems, Microsoft Corporation, 1991.

Downsizing Computer Operations, Digital Equipment Corporation, 1992.

Preston, Alyson, "Make a Smooth Client/Server Transition," PC Week, March 1993: 106

Schwartz, Evan I., "Finally, Software that Slays the Giants," Business Week, March 15, 1993: 96-98

Teaching CS2 With Ada

Dr. Patricia T. Montague
Department of Mathematical Sciences
Campus Box 038
Metropolitan State College of Denver
P.O. Box 173362, Denver, CO 80217-3362
303-556-4518
montague@zeno.mscd.edu

ABSTRACT

This paper describes the author's experiences teaching CS2 with VAX Ada including a class where half of the students had taken CS1 in Pascal and half in Ada. It includes discussion and examples of CS2 level Ada topics such as exceptions, access types, the use of packages to implement abstract data types, and generic packages. The information about Ada included in this paper extends that in the author's earlier paper: "Teaching CS1 With Ada" [ASCUE 1991].

In Spring Semester 1993, I taught a CS2 class where half of the students had taken CS1 in Pascal and half in Ada. The students in the class who had taken CS1 in Ada had used a preliminary version of Smith and Frank [10]. I looked unsuccessfully for a suitable text for CS2. I ended up recommending to students that they buy a reference book on Ada and handed out examples for use in class discussion. Some of these examples were from my paper "Teaching CS1 With Ada" [7]. Others were specifically created for this course and are included in the discussion here.

This paper begins with a discussion of some of the more advanced topics in Ada. The intent is to include material that would be appropriate for a CS2 course. Section 1 is a review of one dimensional arrays in Ada. Linear search for one-dimensional arrays is in Section 2. Multidimensional arrays are discussed in Section 3. Records including discriminated and variant records are discussed in Section 4. Pointers and allocation of storage are discussed in Section 5 and deallocation in Section 6. Exceptions are in Section 7. Section 8 discusses generic units -- packages and procedures. A generic stack package and a generic sort procedure are included. Section 9 discusses the content and programming assignments for the course.

1. REVIEW OF One-dimensional ARRAYS IN ADA

A one-dimensional array type in Ada can be declared with specific subscripts or can be an unconstrained array.

```
type Vec5 is array(1..5) of Integer;
type Intarray is array(Integer range <>) of Integer;
subtype Intarray5 is Intarray(1..5);
```

It should be noted that subtype has been used in defining Intarray5. It is not possible to use type in place of subtype.

The unconstrained array type allows procedures and functions to be written using the type as a formal parameter. At the time of the call, the actual parameter must be a constrained array. For example,

B: Intarray(1..5); or C: Intarray5;

It should be noted that B and a vector of type Vec5 are of different types while B and C are of compatible types.

An aggregate can be used in assigning values to the array B. For example, using positional association

B := (9, 7, 7, 6, 6) or B := (9, 7, 7, others =>6);

will produce the following assignments:

B(1) := 9; B(2) := 7; B(3) := 7; B(4) := 6; B(5) := 8;

Named association can also be used. The same assignments are produced by

B := (1 => 9, 2|3 => 7, 4 => 6, 5 => 8);

or

B := (1 => 9, 2..3 =>7, 4 => 6, 5 => 8);

Note: B:= (1 => 9, 5 => 8; 2|3 => 7, others => 6);

is declared to be illegal by the VAX Ada compiler.

Array slices are permitted. In an assignment such as

B(2..5) := B(1..4);

the right-hand side is evaluated before the left-hand side. This assignment is the same as

B(5) := B(4); B(4) := B(3); B(3) := B(2); B(2) := B(1);

B(5..2) is a null slice since the range 5..2 is null. It should be noted that a null range can have values out of the specified range for subscripts.

The following operators can be used on two arrays of the same type: assignment (:=), equality (=), inequality (/=). Actually array assignment requires that the number of components be the same, not that the bounds be the same. In addition, boolean operators may be applied to onedimensional arrays of boolean values. Furthermore, comparison testing (done lexicographically) is permitted for one dimensional arrays with discrete components.

The type string is predefined in Ada by

```
subtype Positive is Integer range 1..Integer'Last;
type String is array (Positive range <>) of Character;
```

Ada uses single quotes for a character and double quotes for a string. If S1 and S2 are declared by

```
S1 : String(1..3);
S2 : String(1..1);
```

then

```
S1 := "cat";    is equivalent to   S1 := ('c', 'a', 't');
S1 := "b";      is illegal
S1(1) := 'b';   changes  S1 = "cat"  to be  "bat"

S2 := "b";      is legal          S2 := 'b';  is illegal
S2(1) := 'b';   is legal
```

In the writing of procedures/functions using an unconstrained array type for a formal parameter, it is possible to use attributes to access various information about the actual parameter. These are discussed in detail in Section 3. If A is of type Intarray, then A'First is the first subscript of the array and A'Last is the last subscript of the array.

The following procedures and functions illustrate the use of an unconstrained array formal parameter.

```
procedure Fillarray(A: in out Intarray; ACtr: out Integer) is
      I: Integer := A'First;
   begin
      while not End_of_File(FileIn) and I <= A'Last loop
         Get(FileIn, A(I));
         I := I + 1;
         Skip_Line(FileIn);
         end loop;
      ACtr := I  1;
   end Fillarray;


procedure Outputarray(A: in out Intarray; ACtr: in Integer) is
   begin
      For I in Integer range A'First .. ACtr loop
         Put(A(I));
         New_Line;
         end loop;
   end Outputarray;
```

```
function Findlarge(A: Intarray; Actr: Integer) return Integer is p73
      Largest : Integer := A(A'First);
   begin
      For I in Integer range A'First+1 .. ACtr loop
         If A(I) > Largest
            then Largest := A(I);
            end if;
         end loop;
      return Largest;
   end Findlarge;


procedure Searcharray(A: in out Intarray; ACtr: in Integer;
         Pos: out Integer; Val: in Integer; Found: in out Boolean) is
      I: Integer := A'First;
   begin
      Found := False;
      while I <= ACtr and not Found loop
         If A(I) = Val
            then Pos := I;
                 Found := True;
            else I := I + 1;
            end if;
         end loop;
   end Searcharray;
```

2. LINEAR SEARCH IN ADA

The Searcharray procedure above is written in a style usually used for Pascal. Ada provides several other ways of doing the search. The first uses "and then".

```
I := A'First;
While I <= ACtr and then A(I) /= Val loop
   I := I + 1;
   end loop;
If I <= ACtr then Found := True;
         else Found := False;
         end if;
```

The next version uses a for loop and exit. The variable I is needed since J is a temporary variable that doesn't exist after the for loop is executed.

```
Found := False;
For J in Integer range A'First .. ACtr loop
   If A(J) = Val then Found := True;
            I := J;
            exit;
         end if;
   end loop;
```

The for loop above contains the range specification "Integer range". The range specification is not needed if the type of the loopparameter is implicit in the bounds of the range. However, a nonnegative integer literal is of type universal_integer and a negative integer literal is of type integer. One can convert the universal_integer 10 to the integer 10 by "Integer(10)".

3. MULTIDIMENSIONAL ARRAYS IN ADA

The general form for the type of a multidimensional array is

```
type identifier is array (range_1, range_2, ..., range_n) of
          component_type;
```

If the component of an array is an array, then the component has to be a constrained array.

```
type Vector is array (Integer range <>) of Float;
type V_List1 is array (1..10) of Vector(1..4);

subtype Four_Vector is Vector(1..4);
type V_List2 is array (1..10) of Four_Vector;
```

V_List1 and V_List2 are different types. The use of subtype in defining Four_Vector should be noted. It is not possible to use type in place of subtype.

Ada distinguishes between a multidimensional array and an array with array components.

```
type Matrix is array (1..3, 1..2) of Integer;
type Row is array (1..2) of Integer;
type Matrixbyrow is array (1..3) of Row;
A : Matrix;
B : Matrixbyrow;

A := B;  and  B := A;  are both illegal
A(2,1) := 6;  assigns 6 to the component of A in row 2, column 1
B(2)(1) := 6; assigns 6 to the first component in B(2)
A(2)(1) := 6;  and  B(2,1) := 6;  are illegal
```

Array attributes include First, Last, Length, and Range. Each can be applied to the type or to a variable of that type -- for example, B'First is the same as Matrixbyrow'First.

```
B'First      is  1          B'Last       is  3
B'Length     is  3          B'Range      is  1..3
A'First(1)   is  1          A'Last(1)    is  3
A'First(2)   is  1          A'Last(2)    is  2
A'Length(1)  is  3          A'Length(2)  is  2
A'Range(1)   is  1..3       A'Range(2)   is  1..2
```

It should be noted that A'First is the same as A'First(1), etc..
One use of the Range attribute is in a for loop such as

```
for I in B'Range loop ...
    ...
```

Aggregates can be used to assign values to a multidimensional array. An aggregate for a two-dimensional array will consist of one dimensional subaggregates. For example in

```
A := ((1, 2), (3, 4), (5, 6));
```

the subaggregate (1, 2) specifies the first row of A, (3, 4) the second row, and (5,6) the third. Named or positional association can be used for such aggregates. For example:

```
A := (1 => (1, 2), 2 => (3, 4), 3 => (5, 6));
```

or

```
A := (1 => (1 => 1, 2 => 2), 2 => (1 => 3, 2 => 4),
        3 => (1 => 5, 2 => 6));
```

Aggregates are useful in setting all components to the same value:

```
A := (1..3 => (1..2) => 0);
```

For n > 1, an ndimensional array aggregate consists of (n-1) -- dimensional subaggregates. Slices do not exist for multidimensional arrays.

Multidimensional unconstrained array types exist in Ada. All of the dimensions must be unconstrained.

4. RECORDS, DISCRIMINATED RECORDS, AND VARIANT RECORDS

The syntax for records in Ada is similar to that in Pascal.

```
type rec is record
        code:  Character;
        value : Integer;
        end record;
R : rec;
```

The fields of R are R.code and R.value. It should be noted that a field can not have an anonymous type. An aggregate can be used to specify the values for the fields in R.

```
R := ('A', 25);    or    R := (code => 'A', value => 25);
```

is equivalent to

```
R.code := 'A';   R.value := 25;
```

A record aggregate can contain a mixture of named and positional association -- all of the positional notation must occur before any named notation.

Ada allows a record to have one or more discriminants. A discriminant acts as a parameter for the record. Such records are called discriminated records or parametrized records. A discriminant can be any discrete type.

An example of a record with one discriminant is

```
type Char_String(MaxLen: Natural) is record
        Char : String(1..MaxLen);
        end record;
```

A variable of type Char_String must be constrained. This is done when the variable is declared, for example:

```
S : Char_String(5);    or    S : Char_String(MaxLen => 5);
```

S is then a constrained record variable. The value of the discriminant of S can't be changed during execution.

A default value for all of the discriminants can be specified. (It is not possible to have a default value for only some of the discriminants.)

```
type Char_Str(MaxLen: Natural := 0) is record
        Char : String(1..MaxLen);
        end record;
```

A constrained record variable of type Char_Str would be created by

```
S2 : Char_Str(2);
```

The value of the discriminant of S2 can't be changed during execution of the program.

An unconstrained record variable of type Char_Str would be created by

```
S3 : Char_Str;
```

The discriminant value for S3 is the default value which is zero. The discriminant value for S3 can be changed only by assigning another record of type Char_Str to S3 or by using a record aggregate such as

```
S3 := (3,"cat");
```

Changing a discriminant value requires changing the entire record and is possible only if a default value was declared for the discriminant and the default was used when the variable was declared.

An example of a discriminated record with two discriminants is

```
type Mtype is array(Integer range <>, Integer range <>) of Float;
type Matrix(Rows, Cols : Positive) is record p73
      A : Mtype(1..Rows,1..Cols);
      end record;
B : Matrix(3,5);
```

B then represents a matrix with 3 rows and 5 columns. The i,j element of B would be given by B.A(i,j)

Variant records are also available in Ada. Ada requires that the contents of a variant record are always consistent. This means that a variant record variable can be changed only by assigning to it another record of the same type or by assigning to it a record aggre gate. A record can contain only one variant part and it must be the last component. Variants can be nested. However a variant part can contain only one variant part and that must be the last component.

The following is an example of a variant record.

```
type pointtype is record
        x : Float;
        y : Float;
        end record;
type figure is (point, circle, square);
type figurerec(f : figure := point) is record
        base : pointtype;
        case figure is
            when circle => radius : Float;
            when square => side : Float;
                        pt : pointtype;
            when point => null;
            end case;
        end record;
F, P : figurerec;
...
F := (circle, (0,0), 4);
P := (point,(0,0));
```

The fields F.figure, F.base, and F.radius exist. The fields F.side and F.pt do not exist. Note that P does not have a field from the variant part.

5. POINTERS

Pointers in Ada are called access types. Ada provides the literal null which is used to give a value to any access variable which does not point to a dynamic object. Ada variables of type access are automatically initialized to null.

The declaration "type Node;" is called an incomplete type.

```
type Node;
type PtrToNode is access Node;
type Node is record p73
        value : Integer;
        link  : PtrToNode;
        end record;
P : PtrToNode;
```

Storage is allocated for the node pointed to by P by

```
P := new Node;
```

The fields of the Node are then P.value and P.link. P.all refers to the entire dynamic variable. The dynamic variable is initialized by

```
P.value := 10; P.link := null;
```

or an aggregate can be used

```
P.all := (10, null);   or   P.all := (value => 10, link => null);
```

It is also possible to assign values when the storage for P is allocated

```
P := new Node'(link => null, value => 10);
```

6. DEALLOCATING STORAGE

The specification of Ada's generic procedure for the deallocation of storage is

```
generic
    type Object is limited private;
    type Name is access Object;
procedure Unchecked_Deallocation(X: in out Name);
```

To use this with the Node defined in Section 5, include

```
with Unchecked_Deallocation;
...
procedure Free is new Unchecked_Deallocation(Node,PtrToNode);
...
P := new Node;
...
Q := P;
Free(P);      --  results in the value null for P
...           --  storage pointed to by P is released for reuse
              --  value of Q is not changed
Q := (5,null)     --  not a runtime error
                  --  storage pointed to by Q may have or may not
                  --  have been reallocated
```

7. ADA EXCEPTIONS

Various errors that occur in running an Ada program raise exceptions. Some of these are

Constraint_Error	--	value not in range
Name_Error	--	invalid file name used in open or create
Program_Error	--	no return executed in a function or entity used before elaborated
Data_Error	--	data not of the expected type
End_Error	--	attempt to read at end of file
Numeric_Error	--	predefined arithmetic operation cannot return a valid result (example: division by zero)
Storage_Error	--	computer runs out of memory

An Ada program can contain an exception handler for each such error. If the exception occurs, control is transferred to the appropriate error handler -- if such does not exist, the program terminates. Note: "when others" can be used if appropriate. In addition Ada allows programmers to name and declare their own exceptions. These are known as programmer declared exceptions. Such an exception is raised only by the statement "raise <exception name>". It should be noted that "raise" can appear in an exception handler and it reraises the exception being handled.

A frame has one of the following two forms and can be constructed anywhere within the statement part of a procedure or function. A frame can be nested inside another frame.

```
begin
    <statement sequence>
end;
```

```
begin
    <statement sequence>
exception
    <exception handlers>
end;
```

If an exception is raised inside a frame, then

a. If the frame contains an exception handler which deals with that exception, then the exception is lowered (turned off) and the statements in that exception handler are executed. The control then passes to the statement that would have been executed next had the frame completed execution normally.

b. If the frame does not contain an exception handler for that exception, then the exception remains raised and control passes to the statement that would have been executed next had the frame completed execution normally. The exception remains raised so control passes to the exception handler, if it exists, for the outer frame. The exception is said to be propagated from the inner frame to the outer frame.

It should be noted that a frame can also be preceded by a declaration of variables -- for example:

```
declare A: array(1..N) of integer;
begin
...
end;
```

This allows the size of the array A to be determined at execution time.

The exception mechanism can be used to guarantee that data entered is in a certain range. For example, assume that the integer input is to be in the range from 0 to 12.

```
with Text_IO; use Text_IO;
...
subtype Valid_Range is Integer range 0..12;
package Valid_IO is new Integer_IO(Valid_Range);
Number: Valid_Range;
...
begin
...
    loop
        begin
            Put("Enter an integer in the range 0 to 12");
            Valid_IO.Get(Number);
            exit;
        exception
            when Data_Error => Put_Line("Number entered not in range");
                               Skip_Line;
        end;
end loop;
...
```

It should be noted that the style above is not friendly if the user can not enter an integer in the range 0 to 12. This problem is corrected in the next example where the programmer raises an exception.

Code to request the name of an input file from the user and then open that file was included in [7]. That code used an infinite loop to handle the Name_Error exception. Below is a revised version of this code which after the third incorrect name is entered, raises a programmer declared exception. The procedure is contained in an external package.

```
with Text_IO; use Text_IO;
Package OpenFile is
    Procedure Open_Input_File (FileIn: in out File_type);
    No_Input_File : exception;
end OpenFile;
```

```
Package body OpenFile is
  Procedure Open_Input_File (FileIn: in out File_type) is
        InFileName : String(1..40); InNameLength : Natural;
        Count : Integer := 0;
     begin
     Put_Line ("Enter name of input file");
     loop
        begin
           Get_Line (InFileName, InNameLength);
           Open(FileIn, In_File, InFileName(1..InNameLength));
           return;
        exception
           when Name_Error =>
              Count := Count + 1;
              If Count = 3
                 then raise No_Input_File;
                 else Put("Bad file name, try again");
                 end if;
        end;
     end loop;
  end Open_Input_File;
end OpenFile;
```

The file copy program included in [7] is revised here to use the package above.

```
  with Text_IO;  use Text_IO;
  with OpenFile;  use OpenFile;
  procedure FileCopy is
        FileIn, FileOut : File_type;
        Ch : Character;
     begin
        Open_Input_File(FileIn);
        Create(FileOut, Out_File, "copy.out");
        while not End_Of_File(FileIn) loop
           while not End_Of_Line(FileIn) loop
              Get(FileIn, ch);  Put(FileOut, ch);
              end loop;
           Skip_Line(FileIn);  New_Line(FileOut);
           end loop;
        Close(FileIn);  Close(FileOut);
     exception
        when No_Input_File =>  Put("Terminating program");
     end;
  end FileCopy;
```

An excellent step by step development of exceptions is included in the text by Smith and Frank [10].

8. GENERIC UNITS IN ADA

A generic unit in Ada is either a generic package or a generic procedure. It can be considered as a template. A generic unit starts with the word "generic" followed by a list of generic formal parameters. These parameters can be types or subprograms or variables and can have default values. A variable parameter can be of mode "in" or "in out", but not "out".

A package or procedure is instantiated from a generic unit and is called an instance or instantiation of that unit. This process is p73 basic to Ada. A program that includes input and/or output of a variable that is not of type character or string uses an instantiation of a generic unit. For example. Integer_IO is a generic package inside Text_IO. It begins

```
generic
    type Num is range <>;     -- formal generic parameter
package Integer_IO is         -- actual parameter must be integer
...                           --        or a subtype of integer
```

The instantiation is done by a statement such as

```
package MyIntIO is new Integer_IO(Integer);
```

Named association may be used for the actual parameters.

Generic units can be nested inside a generic or nongeneric unit. The specification of the generic unit must precede the code for the unit if both are placed in the same file. If they are in separate files, then the specification must be compiled before the body.

The type formal parameters can be as follows:

```
type T is private;           -- T is any type that allows
                                assignment and comparison ( =, /= )

type T is limited private;   -- T is any type  assignment
                                and comparison need not exist

type T is (<>);              -- T is a discrete type  i.e.,
                                enumeration type or integer type

type T is range <>;          -- T is an integer type
type T is digits <>;         -- T is a floating point type
type T is delta <>;          -- T is a fixed point type

type T is array(index_type) of element_type;    --    T is an array
                                                      type

type T is access any_type;      --  T is an access type
```

The following example of a generic stack has generic parameters for the type of element in the stack and the maximum length of the stack. The storage for the stack itself is declared in the package body and is not accessible to a user of the stack. One advantage of this approach is that the code in the package body does not have the stack as a parameter. This simplifies the interface to the type.

```
generic                                  -- Length, Element are generic
    type Element is private;             -- parameters
    Length : Natural := 100;             -- default value of 100 given
package Stack is                            -- mode of length is "in"
    procedure Push (e : in Element);
    procedure Pop  (e : out Element);
    function Empty return Boolean;
    function Full return Boolean;
    Stack_Overflow : exception;
    Stack_Underflow : exception;
end Stack;
                                       -- body can only include the operations
package body Stack is                           -- of assignment and
    S: array(1..Length) of Element;             -- comparison (=, /=)
    Top : Integer range 0 .. Length := 0;           --    on objects of
    procedure Push (e : in Element) is              --    type Element
            begin
        If Full then raise Stack_Overflow
            else
                Top := Top + 1;  S(Top) := e;  end if;
        end Push;
    procedure Pop (e: out Element) is
        begin
        If Empty then raise Stack_Underflow
            else
                e := S(Top);  Top := Top  1;  end if;
        end Pop;
    function Empty return Boolean is
        begin
        return Top = 0;
        end Empty;
    function Full return Boolean is
        begin
        return  Top = Length;
        end Full;
    end Stack;
```

The following illustrates the use of the generic package Stack in a program.

```
with Stack;
...
package Stack1 is new Stack(Integer, 250);
--  yields a stack of integers, initialized as empty, length 250
package Stack2 is new Stack(Character);
--  yields a stack of characters, initialized as empty, length 100
package Stack3 is new Stack(Element => Integer, Length => 50);
--  using named parameter association
...
Stack1.Push(8);          --  8 is pushed onto Stack1

--  if one also has  use Stack1;  in the code, 8 is pushed onto
--      Stack1 by     Push(8);

--  a program (or subprogram) using the package Stack can contain
--  an exception handler including
--                         when Stack_Overflow
--                         when Stack_Underflow
```

If a generic package is being used to represent a data structure where an operation is applied to two or more instances of that data structure, then the representation of the data structure must be placed in the package specification. The form originally presented for stack is appropriate for that data structure. A revision of it is included below to illustrate the form of a package if the representation of stack is placed in the package specification.

The specification of the package below contains a private part. The part of the package specification before the private part is called the visible part. The visible part contains all the information that another program unit is able to know about the package. The information in the private part of the package specification is for use by the compiler and the package body. Another program unit can not access it.

```
generic
    type Element is private;
    Length : Natural := 100;
package Stack_Package is
    type StackType is limited private;
    procedure Push (Stack : in out StackType; e : in Element);
    procedure Pop  (Stack : in out StackType; e : out Element);
    function Empty (Stack : StackType) return Boolean;
    function Full (Stack : StackType) return Boolean;
    Stack_Overflow : exception;
    Stack_Underflow : exception;
```

```
private
    type Stackarray is array(1..Length) of Element;
    type StackType is record
    S:   Stackarray;
    Top : Integer range 0 .. Length := 0;
end record;
end Stack_Package;

package body Stack_Package is
    procedure Push (Stack : in out StackType; e : in Element) is
        begin
        If Full(Stack) then raise Stack_Overflow
            else Stack.Top := Stack.Top + 1;
                Stack.S(Wtack.Top) := e;
            end if;
        end Push;
    procedure Pop (Stack : in out StackType; e: out Element) is
        begin
        If Empty(Stack) then raise Stack_Underflow
            else e := Stack.S(Stack.Top);
                Stack.Top := Stack.Top  1;
            end if;
        end Pop;
    function Empty (Stack : StackType) return Boolean is
        begin
        return Stack.Top = 0;
        end Empty;
    function Full (Stack : StackType) return Boolean is
        begin
        return  Stack.Top = Length;
        end Full;
    end Stack_Package;
```

The following illustrates the use of the generic package Stack_Package in a program.

```
with Stack_Package;
    ...
    package Stack1 is new Stack_Package(Integer, 250);
    use Stack1;                -- Stack1.S and Stack1.Top can not
    ...                        -- be accessed
    Push(Stack1,8);            -- 8 is pushed onto Stack1
```

The example above declared StackType to be limited private. The details of a type declared as private or limited private are included in the private part of the specification. The private part can include other types and variables. The information placed in the private part is for the use of the compiler and the package body. The compiler needs this information in order to allocate storage for variables of the types declared in the specification. Any information not needed can be deferred to the package body. This permits a type which is private or limited private to be specified in the private part as an access type to an incomplete declaration. Also, a private or limited private declaration can serve as an incomplete declaration. For example,

```
package ...
    type EgType is limited private;        --    or private
    type Ptr is access EgType;             --    legal
    ...
private
    type Data:          --     Data would be declared in the package body
    type EgType is access Data;
    ...
```

If a type is declared to be private, then only ":=", "=", and "/=" for variables of that type automatically exist outside the package in which the type is declared. Other operators such as "+" and "<" can be defined by the package. In the visible part a deferred constant can be defined for a private type. Its value is specified in the private part after the private type is declared in full.

If the type is declared to be limited private, then ":=", "=", and "/=" do not automatically exist for variables of that type outside the package in which the type is declared. This keeps the implementation abstract. If appropriate, "=" can be defined for the type. It is never possible to overload "/=" (it is automatically defined by the equality operator). Assignment is possible inside the package that declares the type, but not outside of the package. A copy procedure can be supplied by the package. It is recommended that limited private be used when ":=" and "=" do not exist for the objects being modeled or do not correspond to the predefined ":=" and "=".

There is a potential problem with the use of limited private. If "=" is declared in the package specification, then that declaration of "=" applies throughout the package specification and body. If a previous version of "=" was available for the type and was needed in the package body, it would not be available. One solution to the problem is to use a derived type in the private part. "=" is then declared for the derived type. For example,

```
package ...
    type EgType is limited private;
    ...
    function "=" (Left, Right : EgType) return Boolean;
    ...
private
    type Data ...
    ...
    type EgType is new Data;
    ...
```

The following is an example of a generic procedure to exchange the values of two variables.

```
generic                               -- procedure specification
    type Element is private;
procedure Exchange (A, B : in out Element);

procedure Exchange (A, B : in out Element) is   -- body
        Temp : Element := A;
    begin
        A := B;   B := Temp;
end Exchange;
```

The following illustrates the use of the procedure Exchange.

```
with Exchange;                 -- no use clause with a procedure
    ...
    procedure Exchange_Integers is new Exchange(Integer);
    ...
    Exchange_Integers(X,Y);        -- X and Y of type integer
```

Generic procedures or packages can be used for sorting. If the type of the Element is private or limited private, then a comparison function for two variables of type Element must be included as a generic function parameter. The example below also includes as parameters an unspecified discrete index type and an unconstrained array type with components of type Element.

```
generic                                   -- specification
    type Index_Type is (<>);
    type Element is private;
    type Element_Array is array(Index_Type range <>) of Element;
    with function "<" (Left, Right : Element) return Boolean;
    procedure Sort (A: in out Element_Array);

procedure Sort(A: in out Element_Array) is        --      body
    -- this is selection sort  each pass puts largest in place
    Max_Loc : Index_Type;
    Max_Val : Element;
begin
    For I in reverse Index_Type'Succ(A'First)..A'Last loop
            Max_Loc := A'First;
            For J in Index_Type'Succ(A'First)..I loop
                If A(J) > A(Max_Loc) then Max_Loc := J;  end if;
                end loop;
            Max_Val := A(J);
            A(J) := A(I);
            A(I) := Max_Val;
            end loop;
    end Sort;
```

The following illustrates the use of the generic procedure Sort in a program.

```
with Sort;
   ...
   type Realarray is array(Integer range <>) of Float;
   procedure SortFloat is new Sort(Integer,Float,Realarray,"<");
   ...
```

The function "<" used by SortFloat will be the standard operator "<" for comparison of real numbers. To sort in descending order, replace the actual parameter "<" by ">".

There are three forms for a formal generic function parameter.

```
with function "<" (Left, Right : Element) return Boolean;
with function "<" (Left, Right : Element) return Boolean is <>;
with function "<" (Left, Right : Element) return Boolean is less_than;
```

For the first form an actual parameter must be supplied. In the second form, the actual parameter may be omitted if a function "<" is visible at the point of instantiation "<", of course, has to have two parameters of the type of the actual parameter for Element. The second form is not permitted for the function "=". The third form is like the second except that the default is provided by a function named less_than.

An excellent development of generics is included in the text by Smith and Frank [10].

9. OUTLINE OF THE COURSE

The examples in Section 1 were handed out to assist students with a Pascal background in their first programming assignment. The program was to read data lines from a file until end of file or the arrays were filled. The form of the data lines were: columns 6-27 contained a name, columns 30-36 a real number, columns 40-46 a real number, columns 50-54 an integer, and columns 56-60 an integer. The rest of the columns were blank. The maximum number of names that could be stored was 56. Two-dimensional arrays were to be used for the storage -- these had to be declared using unconstrained arrays. Processing of the data included input, counting the number of remaining data lines, printing a table of output with a heading and a summary consisting of the number of data lines processed and the number of remaining data lines, an interactive search for a name (to stop when the first character of the name read was a '$'), averaging a specific column and printing names where the last value was greater than the average; and finding the smallest in a specific column. Completion of the first programming assignment put the students who had CS1 in Pascal at approximately the level of those who had it in Ada.

The second programming assignment was to extract words from a data file and store them in order in a sorted array of records. A record contained a word and its frequency count. The program was to use an external package to request the name of a file from the user and to open it. If the file was an input file, then the name of the file was to be a parameter. If after three tries, an appropriate name was not entered a programmer declared exception was raised. This exception was handled in the main program by terminating the program. The main program included procedures to select the length of the array (maximum of 150) and the length of a word (maximum of 20). For each of these, the user was allowed three tries and if an appropriate value was not entered a programmer declared exception was x@3 raised and handled by the main program. Exceptions were discussed in class using the material in Smith and Frank [10] -- this included an example program like that indicated for Valid_Range. A version of the file copy program without procedures and exceptions was handed out.

The third programming assignment was to input an infix expression, convert it to a postfix expression, and then evaluate the postfix expression. Students were provided with pseudocode from Horowitz and Sahni [6]. This pseudocode uses a stack, but allows access to top. It does not use a queue, but the postfix expression can be considered as a queue. Students were required to create two generic packages one for a stack and one for a queue and to use these packages in the program. Generics were discussed in class using the material in Smith and Frank [10]. The first generic stack in Section 8 was handed out.

The fourth programming assignment was to write code for singly linked lists -- procedures to insert in order, print, copy, merge, and to update the list using insert and delete were required.

The fifth programming assignment revised the second to use a binary search tree.

CS2 at Metropolitan State College of Denver is a fourhour semester course. Topics covered included searching, sorting, singly linked lists, doubly linked lists, circular linked lists, binary trees, tree traversal, binary search trees, heaps, and hashing. Elementary topics in software engineering were also included.

The searching methods discussed for arrays were linear and binary. The linear search code in Section 2 was handed out using records with a key (this was done to provide an example of records) and included one with the value sought inserted at the end of the array. For sorting, the methods discussed were bubble, selection, insertion, shell, merge, quick, and heap. Handouts for these algorithms were usually in Pascal and sample exam questions were provided.

11. Summary

There are now several good choices for a CS1 text using Ada -- these are by DeLillo [4], Feldman and Koffman [5], Savitch and Petersen [9], and Smith and Frank [10]. All of these include generics and exceptions. There is still a problem in finding a good choice for a CS2 text using Ada. Access types, linked lists, binary trees, and abstract data types are included in DeLillo [4] and Savitch and Petersen [9]. These could be supplemented as necessary. Either [4] or [9] could then be used for both CS1 and CS2. The data structures text Stubbs and Webre [11] is at a level higher than CS2.

12. REFERENCES

1. Barnes, J.G.P., "Programming in Ada", 3rd ed, AddisonWesley, 1989
2. Barnes, J.G.P., "Programming in Ada plus Language Reference Manual", 3rd ed, Addison-Wesley, 1991
3. Cohen, Norman H., "Ada as a Second Language", McGrawHill, 1986
4. DeLillo, Nicholas J., "A First Course in Computer Science with Ada", Irwin, 1993
5. Feldman, Michael B. and Koffman, Elliot B., "Ada : Problem Solving and Program Design", Addison-Wesley, 1992
6. Horowitz, Ellis and Sahni, Sartaj, "Fundamentals of Data Structures", Computer Science Press, 1983
7. Montague, Patricia T., "Teaching CS1 with Ada", Proceedings of the 1991 ASCUE Summer Conference, p. 90-108
8. "Reference Manual for the Ada Programming Language", ANSI/MIL STD1815A1983, United States Department of Defense, February 17, 1982 (available from the U.S. Government Printing Office, price $16)
9. Savitch, Walter J. and Petersen, Charles G. "Ada : An Introduction to the Art and Science of Programming", Benjamin/Cummings, 1992
10. Smith, James F. and Frank, Thomas S, "Introduction to Programming Concepts and Methods with Ada", McGraw-Hill, scheduled to be published in 1993 pbW3
11. Stubbs, Daniel F. and Webre, Neil W., "Data Structures with Abstract Data Types and Ada", PWS-Kent, 1993
12. Volper, Dennis and Katz, Martin D., "Introduction to Programming Using Ada", Prentice-Hall 1990

Teaching Matrix Algebra Using MATLAB

Dr. Patricia T. Montague
Department of Mathematical Sciences
Campus Box 038
Metropolitan State College of Denver
P.O. Box 173362, Denver, CO 80217-3362
303-556-4518
montague@zeno.mscd.edu

ABSTRACT

MATLAB is an interactive scientific and engineering software package that can be used for matrix computations and for solving numerical linear algebra problems. It is available for a number of different computers including the Macintosh, MSDOS personal computers, and the VAX; student versions exist for MSDOS and Macintosh computers. This presentation will demonstrate various features of MATLAB and discuss how it can be used in teaching matrix algebra and linear algebra.

MATLAB is a commercial software package produced by The MATHWORKS, Inc. The student version is sold by PrenticeHall. MATLAB provides an interactive system and a programming language that can be used to solve numeric problems. It can be used for matrix computations and for solving numerical linear algebra problems. In addition, it includes elementary statistical commands. Several different application toolboxes are available such as spline toolbox, chemometrics toolbox, robust control toolbox, and a neural network toolbox. The Signals and Systems Toolbox is included with the student version.

The name MATLAB stands for matrix laboratory. The first version of MATLAB was written at the University of New Mexico and Stanford University in the late 70's. The basic data element of MATLAB is a matrix whose size is specified as it is used. MATLAB does not contain type declarations or dimension statements.

The Student Edition of MATLAB for the PC requires at least 320KB of memory, MSDOS version 3.1 or higher, a hard drive with at least 3 MB of free space, a floppy disk drive, and a graphics card (CGA, EGA, HGC, VGA, ATT, or compatible). Color graphics are available on systems with a VGA or an EGA with 256KB of video memory. The Student Edition also uses a math coprocessor if available.

The Student Edition of MATLAB for the Macintosh requires at least 1 MB of memory, system software version 6.0 or higher, a single 800KB disk drive and a hard disk or two 800KB drives, and the Monaco 12 point font.

The current Student Edition is based on the professional version 83 3.5 of MATLAB. In the Student Edition, each vector, or matrix, is limited to 1024 elements and the metafile and graphics postprocessor feature for graphics hardcopy is not available. The professional version is available for machines such as Sun, DEC VAX, DEC RISC, Hewlett Packard/Apollo, Silicon Graphics, IBM RS/6000, Convex, Alliant, and Cray.

1. GETTING STARTED

To start MATLAB, one types "MATLAB" from the operating system prompt. The MATLAB banner appears and then the MATLAB prompt which is ">>". One then types a MATLAB command followed by a return. MATLAB executes the command immediately. Two or more commands can be separated by commas. MATLAB is called an expression language since most of its commands have the form expression or variable = expression. Be default MATLAB is case sensitive and MATLAB commands are in lower case this can be changed by using the "casesen" command. Variables and function names are formed by a letter, followed by any number of letters, digits, and underscores. Only the first 19 characters of a name are remembered.

Special Commands

help command	to access the online help facility for information about the command specified
casesen off	makes MATLAB not case sensitive
casesen on	makes MATLAB case sensitive
casesen	toggle for case sensitivity
who	displays variables and the amount of space left
whos	displays detailed information about the variables
clear	clears the workspace
clear A B C	clears the variables A, B, and C
what	lists .m and .mat files
dir	lists all files
type filename	displays the content of the file specified -- if no extension is given, .m is used
flops	displays the count of floating point operations used
flops(0)	resets the count to 0
quit	terminate program
exit	terminate program
<CTRL-C>	stops execution of a MATLAB command -- may need more than one

Special Characters

%	anything on a line following the "%" is a comment
!	indicates that the rest of an input line should be issued as a command to the operating system -- for example: to access a small editor -- the editor may be accessible without using the "!"

The numbers used in MATLAB can be real or complex. "pi" has the value of pi unless it has been redefined. "i" and "j" have the value of sqrt(1) unless they have been redefined. The complex number 3 + 4i would be given in MATLAB by "3+4*i". It should be note that there is no space between the 3 and the +. If one types the number as "3 + 4*i", then one has entered two numbers into MATLAB -- the number 3 followed by the number 4i.

One can specify a short or long format for numbers. The default is a short format with 5 decimal digits except for zero which is always displayed as 0. If all the values in a matrix are integers, then the matrix is displayed with all values in integer format. The format of a number is important. A number displayed as 0.0000 is not exactly zero. A scale factor may appear in the display. If there exists an element in the matrix with magnitude larger than 1000 or less than 0.001, a common scale factor is used.

Formats

format short	5 digit scaled fixed point
format long	15 digit scaled fixed point
format short e	5 digit floating point
format long e	15 digit floating point
format hex	hexadecimal
format +	+/- format -- displays + for a positive number, for a negative, and a blank for zero
format bank	fixed dollars and cents
rat(A,'s')	displays the matrix A in string form using rational numbers (i.e., fractions) -- result is for display purposes only and can not be used with arithmetic operations

In addition one can specify

format	returns to the default formats
format compact	suppresses excess linefeeds and results in a slightly more compact display
format loose	default display

The variable "eps" is a permanent variable which is referred to as the machine epsilon. Its initial value is the distance from 1.0 to the next largest floating point value which for a computer using IEEE floating point arithmetic is approximately 2.2204E-016. "eps" gives the relative accuracy of numbers.

MATLAB does not include a built-in editor. The following table lists the editing features provided by MATLAB.

LastLine Editing and Recall Keys

Up Arrow	Recall previous line
Down Arrow	Recall next line
Left Arrow	Move left one character
Right Arrow	Move right one character
Ctrl-Left Arrow	Move left one word
Ctrl-Right Arrow	Move right one word
Home	Move to the beginning of the line
End	Move to the end of the line

Esc	Cancel current line
Ins	Toggle between insert and overtype mode
Del	Delete character at the cursor
Backspace	Delete character to the left of the cursor

2. ENTERING MATRICES IN MATLAB

To enter a matrix in MATLAB -- from the prompt ">>", type

```
A = [2 3 4; 5 6 7]
```

MATLAB will display an echo:

```
A =

  2   3   4
  5   6   7
```

To enter a matrix without an echo, add ";" to the end of the line:

```
A = [2 3 4; 5 6 7];
```

An alternate way of entering A would be to type one row per line.

```
A = [2 3 4
   5 6 7];
```

Note: On the VAX this style is awkward -- the return at the end of the first line results in a blank screen with the ability to type at the bottom of the screen.

If the elements of a row do not all fit on one line, then an ellipsis consisting of 2 or more periods, "..", followed by a carriage return can be used to indicate that the line continues to the next line. Commas can be used as separators between matrix elements.

The matrix elements entered can be expressions including functions such as "sqrt" or "abs". Trigonometric functions are also available. If specific numbers are not used, then MATLAB must be able to evaluate all variables used.

```
x = -5; y = 4; B = [x y; sqrt(y) abs(x)]
B =
   -5   4
    2   5
```

One can change an individual element in a matrix. If no ";" is used, then the resulting matrix is echoed.

```
A(2,3) = 8

A =
  2   3   4
  5   6   8
```

One can extend the size of a vector or matrix by specifying an element. For example:

```
X = [3 8 9]; X(5) = 6

X =
  3   8   9   0   6
```

It should be noted that X(4) has been given the value zero.

In all the examples so far, a name has been specified for the result of the MATLAB calculation. If no name is provided, then MATLAB uses the variable "ans".

```
[3 4; 5 7]

ans =
  3   4
  5   7
```

MATLAB has provided commands to create some commonly used matrices (A is a mxn matrix and not a scalar):

eye(m,n)	mxn matrix with one's down the diagonal and zeros elsewhere -- same as eye(A)
eye(n)	nxn identity matrix
zeros(m,n)	mxn matrix with all entries zero -- same as zeros(A)
zeros(n)	nxn zero matrix
ones(m,n)	mxn matrix with all entries one -- same as ones(A)
ones(n)	nxn matrix with all entries one
rand(m,n)	mxn matrix with all entries random numbers -- same as rand(A)
rand(n)	nxn matrix with all entries random numbers

The distribution used to determine the random numbers can be uniform or normal. The default is normal.

rand('normal')	changes the distribution to normal
rand('uniform')	changes the distribution to uniform
rand('dist')	returns the current distribution
rand('seed')	returns the current value of the seed
rand('seed',n)	sets the seed to n -- 0 is the initial value for the seed

M-files allow the use of an editor in creating MATLAB commands and allow the repeated use of MATLAB commands without reentering the commands each time. It is also possible to use M-files to create new functions an example of this is included in Section 8. In a multiuser or networked system, the file named matlab.m is reserved for the system manager. The M-file matlab.m is automatically executed when MATLAB is started. If an M-file named startup.m exists, it is invoked by the execution of matlab.m.

4. KEEPING A RECORD OF WHAT WAS DONE

MATLAB provides diary files as a way to keep a log of what was done in a MATLAB session (graphs are not included in the file and the filename can not be "on" or "off"). The diary files are ASCII files. They are automatically saved at the end of a MATLAB session (there is a save command in MATLAB, but that is not for diary files). The following are commands related to diary files.

diary filename	starts the diary in the specified filename. If this file already exists, then the log is appended to the end of the file (in some versions the old file is destroyed)
diary	toggles the diary on and off. If no filename has been chosen for the diary, then the default name "diary" is used
diary on	turns the diary on -- uses the current name filename or the default name "diary" if no filename has been specified
diary off	suspends the diary

For teaching purposes, diary files provide a way to verify that the student did the required exercises. Students can use comment statements to label the exercises and to provide explanations or conclusions. Alternatively, these can be added by a text editor or by a word processing program or by hand. Diary files will generally contain lines of errors due to a mistyped statement or an incorrect attempt at solving a problem. This output is only with difficulty readable for grading purposes. This problem can be alleviated by requiring the student to edit the diary file before submission.

MATLAB also provides a way to save the variables used in a MATLAB session and to retrieve the variables for a later MATLAB session. These are done through the use of the "save" and "load" commands. Files used for this are called MAT-files and are full precision binary MATLAB format files. The name for a MAT-file must have the extension .mat. MAT-files can be used to transfer variables between various computers. Each MAT-file contains a signature that indicates the machine that wrote the file. The load command checks the signature and performs numeric conversions. Full numeric precision is obtained when moving between IEEE arithmetic machines such as the PC and Macintosh. Precision is lost when files are exchanged with the VAX.

load temp	loads the variables saved in temp.mat into the current MATLAB session
load	loads the variables in matlab.mat
save temp	saves all variables in use during the current MATLAB session in the file temp.mat

save	saves all variables in use in matlab.mat
save temp X	saves the variable X
save temp X Y	saves the variables X and Y

Some students decide to use "save" for their diary file. This results in a file which is a mixture of text and binary values. For this reason students should be instructed to not use "save" for diary files. The discussion of MAT-files should also be postponed until after students have learned how to use diary files.

There is another version of "load"/"save" that involves only one variable and uses ASCII files. These ASCII files are called ASCII flat files -- the data is stored in ASCII form with fixed-length rows with spaces separating the numbers. Each row is terminated by a new-line (carriage return). An ASCII flat file can be edited with a text editor. The variable name is given by the name of the file (without the extension).

save temp.dat X /ascii	saves the variable X in the ASCII flat file temp.dat in 8 digits
save temp.dat X -ascii	same as above
save temp.dat X /ascii /double	same as above except 16 digits
load temp.dat	loads the variable in the ASCII flat file temp.dat as the variable temp

5. BASIC OPERATIONS ON MATRICES

MATLAB contains commands that perform the basic operations on matrices. The matrices A and B have to have the appropriate sizes for the operations to be defined.

A + B	sum of the matrices A and B
A - B	matrix B is subtracted from the matrix A
A*B	matrix A is multiplied by matrix B on the right
c*A	every element of the matrix A is multiplied by the scalar c
A^n	nth power of A (n > 1) -- A^0 is eye(A)
A'	complex conjugate transpose of the matrix A (same as the transpose of A if A is real)
A.'	transpose of the matrix A
conj(A)	complex conjugate of the matrix A
conj(A')	transpose of the complex matrix A
trace(A)	sum of the diagonal elements of the matrix A
A.*B	result is a matrix where the elements are products of p73 the corresponding elements in A and B
A.\B	result is a matrix where the elements are the quotients of the elements in A by the corresponding elements in B
A.^B	result is a matrix where the elements are the elements of A raised to the powers given by the corresponding elements in B
A+k	same as A + k*ones(A)

For vectors MATLAB provides three norms:

norm(X)	Euclidean length of the vector X. This is sqrt(X'*X) for X a column vector and sqrt(X*X') for a row vector. This is the same as norm(X'), norm(X,2) and norm(X',2).
norm(X,1)	1-norm of X which is the sum of the absolute values of the elements of X. This is the same as sum(abs(X)).
norm(X,inf)	infinity norm of X which is the maximum of the absolute values of the elements of X. This is the same as norm(X,'inf') and max(abs(X)).

The dot product (or inner product or scalar product) of the two real column vectors x and y can be obtained by x'*y or sum(x.*y). MATLAB does not provide a command to calculate the cross product of two row vectors.

6. ADVANCED OPERATIONS ON MATRICES

rank(A)	rank of the matrix A
det(A)	determinant of the nxn matrix A
rref(A)	row reduced echelon form of the matrix A
rrefmovie(A)	displays the step by step calculations used to determine the row reduced echelon form of A
inv(A)	inverse of the nxn matrix A
[L,U] = lu(A)	LU decomposition of A -- A = L*U values of L and U are displayed
[L,U,P] = lu(A)	LU decomposition of A -- P*A = L*U values of L, U, and P are displayed
[Q,R] = qr(A)	QR factorization of A -- A = Q*R values of Q and R are displayed
[U,S,V] = svd(A)	singular value decomposition -- A = U*S*V' values of U, S, and V are displayed
pinv(A)	pseudoinverse of A
Q = orth(X)	columns of Q form an orthonormal basis for the column space of X -- Q has rank(X) columns
Q = null(X)	columns of Q form an orthonormal basis for the nullspace of X -- number of columns of Q is the nullity of X (that is, the number of columns of X minus rank(X))

Note that one can also find the inverse of the nxn matrix A by row-reducing [A eye(n)] to [I B] where I is the nxn identity matrix and B is the inverse of A. The command "inv" is more accurate and will give a warning message if the inverse calculated is questionable.

The rank of a matrix can be determined by using "rref" and counting the number of non-zero rows in the resulting matrix. The "rank" command is considered to be more accurate.

7. SOLVING SYSTEMS OF EQUATIONS

A linear system of m equations in n unknowns can be solved using MATLAB in a variety of ways. Suppose the system is represented by AX = b where A is an mxn matrix, X is a nx1 matrix and b is a mx1 matrix. Let aug = [A b] be the augmented matrix of the system.

Method 1: rref(aug)

The resulting matrix is the row reduced echelon form of aug and it can be used to determine the general solution of A*X = b. The method used is Gaussian elimination with partial pivoting and a default tolerance is used to test for entries which should be replaced by zero. The form "rref(aug,tol)" allows the user to specify the tolerance, "tol", to be used.

Note: One can use "rrefmovie(A)" to show a movie of the algorithm in action -- each step is displayed. This is done without interruption -- if one is interested in investigating the actual steps performed by "rref", then one should do "rrefmovie" with a diary on.

Method 2. Elementary row operations done step by step using MATLAB

One would do elementary row operations to determine the row reduced echelon form of aug and then determine the general solution of A*X = b from it. The elementary row operations are done as follows.

a. To exchange rows i and j of aug, do

```
temp = aug(i,:); aug(i,:) = aug(j,:); aug(j,:) = temp
```

(this just displays the final version of aug) or

```
aug([i j],:) = aug([j i],:)
```

b. To multiply row i of aug by the scalar c, do

```
aug(i,:) = c*aug(i,:)
```

c. To replace row i of aug by row i plus c times row j, do

```
aug(i,:) = aug(i,:) + c*aug(j,:)
```

The step by step method can be used for homework and takehome exams. Since MATLAB has done the arithmetic, one only needs to look at the pattern of the steps chosen by the student. An edited version of a diary file should result in only the relevant steps included.

Method 3: X = inv(A)*b if A is invertible

This method is not efficient.

Method 4: X = A\b if A is invertible

The solution is computed using Gaussian elimination. b can consist of more than one column. A warning message is printed if A is badly scaled or nearly singular.

Method 5: X = A\b if A is mxn and not square

This method determines a solution X in the least squares sense to the overdetermined or underdetermined system. If the system is consistent, then X is a particular solution of A*X = b. If the system is inconsistent, then the resulting X is such that A*X is close to b. If A is rank deficient, i.e., rank(A) < n, a message is given. If k is the rank of A, then X has at most k nonzero elements per column.

The condition number of A is determined in MATLAB by the command "cond(A)". If the value of the condition number of A is large, it is a warning that small changes in the elements of A may produce large changes in the elements of "A\b". The computer can lose "log10(cond(A))" decimal places to roundoff errors in Gaussian elimination.

Method 6: X = backsub(A,b) if A invertible, and an upper or lower triangular matrix

However, it is usually faster to use X = A\b.

Method 7: null(A) and A\b

The columns of the matrix produced by "null(A)" give an orthonormal basis for the nullspace of A (i.e., the solution space of AX = 0). A particular solution can be found by using "A\b". The general solution is then found by taking an arbitrary linear combination of the columns of A and adding the particular solution to the result.

Method 8: Cramer's Rule. A is nxn and invertible. Let Ai be the nxn matrix obtained from A by replacing the ith row of A by b. Then for each i

X(i) = det(Ai)/det(A)

This can be done in MATLAB by using a "for" loop to calculate the elements of X and then display X.

```
d = det(A); X = zeros(n,1);
for j=1:n, Aj = A; Aj(:,j) = b; X(j) = det(Aj)/d; end, X
```

Method 9. Jacobi, iterative method. A is nxn and invertible. Let N = diag(diag(A)). Choose an initial value for X and then repeatedly calculate X by

X = N\(P*X + b))

Section 8 includes a function which does the Jacobi method.

Method 10. GaussSeidel, iterative method. A is nxn and invertible. Let N = tril(A). Choose an initial value for X and then repeatedly calculate X by

X = N\(P*X + b))

If one is interested only in determining the existence of a solution, then rank should be used. AX = b has a solution if and only if rank(A) is the same as rank([A b]). The rank command can also be used to determine the number of parameters in a consistent system. If A is mxn, then the number of parameters is n-rank(A).

8. CREATING MATLAB FUNCTIONS

It was mentioned earlier that a file with extension of .m could be a MATLAB function. Functions can be used to extend MATLAB providing, for example, the cross product of two 3-vectors, Cramer's rule, or an iterative solution method.

The following function performs the Jacobi method for solving a nonsingular system AX = b iteratively. It has been adapted from Hill [2, p. 368]. It should be noted that the function has parameters and returns a value. The variables used in a function must be either parameters or local variables.

```
function y = jacobi(A,b,X,maxit,tol)
% X  --   initial guess  a column
% maxit -- maximum number of iterations to be performed
% tol  --  tolerance for determining convergence
N=diag(diag(A)); P=NA;
k=1; lastX=X;
while k<=maxit
        X = N/(P*lastX+b);
        if norm(lastX - X,inf) <= tol
                disp('steps for convergence'),k
                break
                end
        k=k+1; lastX=X;
        end
if k > maxit
        disp('maximum number of iterations exceeded')
        end
y=X;
```

When a function, say "eg", is used that MATLAB does not recognize, it will search for a file named eg.m. If found, then the function "eg" is compiled into memory for later use in the current session.

Functions could be used to create MATLAB commands that could be useful in teaching. These could extend MATLAB as in the Jacobi example or modify existing commands. An example of the latter, would be a swap(A,i,j) function that would exchange the ith and jth rows of the matrix A. There is a disk of auxiliary MATLAB functions for the Kolman text [3]. It includes a function called reduce that uses a menu structure to let a user select which elementary row operation to be performed next.

9. EIGENVALUES AND EIGENVECTORS

MATLAB provides commands specifically for the calculation of the characteristic polynomial of a matrix and its eigenvalues.

poly(A)	characteristic polynomial of the nxn matrix A -- given as a row vector whose elements are the coefficients of det(xI - A) ordered by descending powers of x
roots(p)	gives a column vector with elements that are the roots of the polynomial represented by the vector p with coefficients ordered by descending powers
roots1(p)	same as roots(p) except the roots are more accurate when there are repeated roots
roots(poly(A))	gives the eigenvalues of A as a column vector
eig(A)	gives the eigenvalues of A as a column vector
[P,D] = eig(A)	gives a diagonal matrix D whose diagonal elements are the eigenvalues of A and a matrix P with columns that are eigenvectors of A such that A*P = P*D . The eigenvectors are scaled so that the norm of each is 1.0.
[P,D] = eig(A,'nobalance')	has no preliminary balancing step

Ordinarily, balancing enables more accurate computation of the eigenvalues and eigenvectors. However, if the matrix contains small elements that are really due to roundoff error, balancing may scale them up to make them as significant as the other elements. Balancing may then result in incorrect eigenvalues for the matrix.

10. SOME MISCELLANEOUS MATLAB COMMANDS

MATLAB uses a row vector to represent a polynomial. The elements of the vector are the coefficients of the polynomial ordered by descending powers. Some of the commands for polynomials are as follows.

polyval(p,a)	p represents a polynomial, a is a scalar result is the value of the polynomial evaluated at a
polyvalm(p,A)	like polyval except A is a matrix
roots(p)	p represents a polynomial result is a column vector with elements that are the roots of the polynomial
roots1(p)	same as roots(p) except the roots are more accurate when there are repeated roots
poly(r)	r is a column vector containing the roots of a polynomial -- poly(r) is a row vector representing the polynomial
conv(p,q)	p and q represent polynomials -- the result represents the product of the two polynomials
[q r]=deconv(p,d)	p, d, q and r represent polynomials q represents the quotient of p by d r represents the remainder

MATLAB includes commands for graphics. The command "plot" creates linear XY plots. One can also do "loglog", "semilogx", "semilogy", "polar", "mesh" (3-dimensional mesh surface) and "contour" plots.

plot(Y)	Y a row vector -- plots the points (i,Y(i))
plot(X,Y)	X, Y row vectors -- plots the points (X(i),Y(i))

Some other MATLAB commands are listed below.

sum(A)	if A is not a row, then the result is a row where each element represents the sum of the elements in that column of A; if A is a row, then the sum of the elements in A
size(A)	gives size of A
length(X)	gives length of the vector X -- number of elements
magic(n)	gives an nxn magic square

11. EXPERIMENTS

Some matrix algebra and linear algebra courses are being taught using a program like MATLAB to provide students the opportunity to discover theorems and properties.

An earlier example of such exploration problems would be to have students determine when $A + B = B + A$ and $AB = BA$.

Another exploration problem would be to provide the students with matrices A, B, and C. The student is to calculate the trace of the matrices AB, BA, AC, CA, BC, and CB. Then the student is to make a conjecture about the relationship between the trace of PQ and QP. The conjecture can be tested by using rand to create matrices P and Q.

12. SUMMARY

MATLAB provides a powerful tool for doing calculations in matrix algebra and numerical linear algebra. It is easy to learn and can be used in a variety of ways in a course in matrix algebra or linear algebra. It provides commands that accurately perform standard algorithms and provide warning messages if appropriate. Since it is also a programming language, MATLAB can be extended by the teacher and, if appropriate, the students.

13. REFERENCES

1. Goldberg, Jack L., "Matrix Theory with Applications", McGraw Hill, 1991

2. Hill, David R., "Experiments in Computational Matrix Algebra", Random House (now McGraw-Hill), 1988

3. Kolman, Bernard, "Introductory Linear Algebra with Applications", 5th ed, Macmillan 1993

4. Marcus, Marvin, "Matrices and MATLAB, A Tutorial", Prentice-Hall, 1993

5. The MathWorks Inc., "The Student Edition of MATLAB, Student User Guide", Prentice-Hall, 1992

6. The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, MA 017601520, phone: 5086531415, fax: 5086532997, Email: info@mathworks.com

LanguageWriter: A Multimedia Tool for Instructional Development

Robert L. Oakman
Jay A. Waller
Reginald A. Riser
Fred Fenimore
Department of Computer Science,
University of South Carolina
Columbia SC 29208
803-777-2401
oakman@usceast.cs.scarolina.edu

ABSTRACT

LanguageWriter (Macintosh) allows teachers to prepare their own coordinated text and sound instructional materials. Instructors can record their own digitized sounds, relate the sounds to written text, and save the results in a lesson format. Students can later access the lesson, containing both sound and text, on the computer network. They can select words or phrases and play back the sounds corresponding to their selection. They can record their own voices, compare their speech to the instructor's, or answer questions in written form. With extensions incorporating digitized video, **LanguageWriter** offers multiple modalities for lesson development under the control of the classroom teacher.

In the past computer-assisted instruction in the language classroom has not been used to its fullest potential, partially because until recently the speech capabilities of computers have not been adequate. Major sound extensions to inexpensive Macintosh computers have fostered the development of a new program called **LanguageWriter**. The program expands the notion of teacher/student interaction to allow flexible combinations of text and voice communication for language instruction.

Today's Macintosh computers permit recording of sound with fine voice and sound reproduction. The ability of the teacher to include recorded conversational materials along with the written version of the texts makes **LanguageWriter** especially useful for students of foreign languages and English as a second language, as well as for bilingual and dialect-disadvantaged learners.

In **LanguageWriter**, instructors can record their own sound files, relate pieces of the sound to written text, and save the results in a lesson format. Using a easy-to-use graphical interface built into the program, the teacher examines recorded materials as a sound spectrograph and marks individual words or phrases. The combined text and sound files are then stored on a hard disk in a computer network for later student retrieval.

Students access the lesson, which contains both sound and text. They can then select words, phrases, or sentences with a mouse and play back the sounds corresponding to their selection. They can easily record their own voices and compare their speech to the instructor's by listening to their sounds and the teacher's sound. They can also type responses to the text materials that they have been reading and hearing.

1.0 THE TECHNOLOGY OF LANGUAGEWRITER

The **LanguageWriter** software comes in two versions: **LangWriterInst** and **LangWriterStudent**. The instructor's software includes features for recording and typing in texts for student use. The student version of the package allows retrieval of the text materials and the associated voice files. It also permits creation of voice recordings and typed responses that may be saved for the teacher.

By means of a sample session, let us examine how both an instructor and a student might use the program. The instructor opens **LanguageWriter** by double clicking on the **LangWriterInst** icon in order to create a new text and sound file. Once a name has been chosen for the file, the program opens a blank sound window that consists of three parts. The top part of the window is the sound wave editor, where the sound portion of lessons is displayed. The middle portion of the window contains all of the sound controls necessary for the instructor to create a lesson. The bottom part of the window is the text editor where the typed version of the lesson will be keyed.

Creating a lesson involves three steps: the sound must be recorded, the text must be typed, and the sound and text must be marked and synchronized together. It makes no difference whether the sound or the text is input first, but both must be present for the marking process. Instructors can prepare lessons using the method that is easiest for them. They may first choose to type in the text; then they can record the segments by reading the text from the screen. After recording, they then go back and mark the words for each sound segment. Another strategy is to record a single segment, type in the corresponding text, mark the words in the segment, and move on to the next segment.

Let us follow the second method of preparing the lesson segment by segment through the complete recording, marking, and saving process. The instructor first records the initial sound segment. Recording begins as soon as the "Record" button is pushed and the instructor voices the segment of text. Pressing the "Stop" button terminates the recording. The complete sound wave form of the recorded passage then appears in the sound wave box in the top of the window. At this time, the instructor should save the recorded sound with the "Save" button. Writing out each sound segment to disk saves memory in the computer, since sound files are expensive of storage.

To play back the entire segment, instructors can use the "Play" button. If they made an error in recording or are unhappy with the sound, they can record it again by going through the same process. Satisfied with the recording, teachers type the accompanying text in the writing box in the bottom portion of the sound window. Now ready to mark the sound, they select a portion of the sound with the mouse in the sound wave box. They can listen to the sound by pressing the ōPlayō button in the middle section of the window. They can then adjust the selection with the mouse and the "Play" option until they are satisfied that they have found a meaningful segment of sound, such as a word or phrase. To aid the instructor, the sound menu contains a "Zoom In" command that magnifies a selected portion of a sound wave by scaling that portion to fit within the wave box. While the wave is magnified, the instructor can continue to play and mark more finely tuned selections of sound.

Having determined the portion of the sound to mark as a word or phrase, the instructor highlights the corresponding word or phrase in the text box. To set the mark for the section means to coordinate the sound and text together. Pressing the "Mark" button stores the sound and text positions for the words marked within the current sound segment. Once the instructor has marked all of the words or phrases within the segment, the "Save" button writes the sound and mark data to the disk.

To create a new segment, the instructor clicks on the "New" button. Using this same procedure, the instructor can go through the same steps to record the next sound, create the text, and mark the appropriate segments, normally words or phrases. At any time, the instructor can go back to a previous segment to redo the recordings or markings. When the instructor is finished, the sound window is closed; and all changes to the file, which now contains text and voice recordings, are saved to disk and locked.

Students start the **LangWriterStudent** version of the software by logging onto the network with their names and class numbers. They open sound files just as they open any other file, by selecting the "Open" command from the File menu and then selecting the sound lesson that the instructor has prepared. Using the options under the sound menu, the student has a choice of playing a segment, playing the entire passage, or recording a response. If the student chooses to play a segment, then the sound segment containing the current text cursor position is played back to the student. As the segment is played back, the text that corresponds to the sound is highlighted so that the student can see what is being played.

Students can also select a part of the passage, such as a word or a phrase, that they want to hear. They highlight the appropriate text with the mouse and choose the "Play Selection" command from the sound menu. If the student selection does not match exactly the word and phrase divisions that the instructor has set, then the highlighted text is updated to show all of the words that are played back to the student. For example, suppose a lesson contains the word "can't," and the student only highlights the word "can." In this case, the highlight will be updated to include all of the word "can't", which will then be voiced to the student.

At any time, students can record their own voices by choosing the "Record" option from the Sound menu. The program presents the student with controls analogous to a tape recorder. To start recording, the student presses the "Record" button and then vocally repeats some part of the lesson (or perhaps responds to the original material if it were a question). With the "Stop" and "Save" buttons, the student's recording is written to a file for later review by the instructor. Students can then play back their own sounds by choosing "Play Recording" from the Sound menu. By alternating between playing the instructor's recording and their own recordings, they can compare the sounds and detect possible pronunciation errors. Students can continue to record, play back, and practice sections of the lesson. Or they can respond to their assignment in writing in the Message window at the bottom of their screen. When they are finished with their lesson, saving their files writes out on disk both the text and sound files they have created.

2.0 THE PEDAGOGY OF LANGUAGEWRITER

A lesson in **LanguageWriter** can easily be organized by the teacher as an oral and written test. Perhaps the instructor has both recorded and typed in a series of Spanish questions. The students can later read and hear them, record their Spanish answers in their own sound files, and type English translations in the window set aside for their writing responses.

Or they could type their answers in Spanish in their writing space. All student files, whether text or voice, are then saved for later checking by the teacher.

Perhaps **LanguageWriter** is being used for students with English dialect difficulties. Such students can read materials but pronounce them differently from standard speakers. In LanguageWriter they get to hear standard pronunciation as they are reading the text. After they prepare their voice recordings, it is easy for them to compare their dialect versions of the text to received pronunciations with the written text in front of them.

LanguageWriter is also appropriate for blind students. Perhaps they are supposed to answer some questions. Their teacher records the text for them and stores it as a sound file. Later blind students can hear the questions and, using a keyboard designed for blind learners, type their responses. Of course, they will not be able to see the text prepared by the teacher.

LanguageWriter has been designed to offer a great variety of teaching and learning styles in language classes using a combination of sound and text. It allows teachers a method of easily recording sound and intermingling the sound with text. Students have the ability to access the sound-and-text files prepared by the instructor and record their own sounds for comparison or type their responses in the text window.

3.0 NEW EXTENSIONS TO THE PROGRAM

Currently **LanguageWriter**, Version 1.0, handles only languages in the Roman alphabet. New extensions already in development will incorporate facilities for a variety of different, less familiar character sets for preparation of language lessons. Currently envisioned are Russian, Greek, Hebrew, Arabic, Japanese, and Chinese. Apple Computer announced in October, 1992, their support of all of these character sets and fonts for Macintosh under the name of WorldScript. We have tested a number of them, and are planning to incorporate them into the **LanguageWriter** environment when WorldScript has stabilized to allow teachers to develop language materials in these less common languages. We note that Chinese and Japanese are particularly difficult, and success is not assured.

In addition, current ideas of language pedagogy stress the success of interactive video. Students can replay segments of real television or videotape on the computer and simultaneously be presented language materials to test mastery of native-speaker dialogues and interactions. Using the Macintosh digitized video extension called QuickTime, segments of live video and still pictures can be incorporated into **LanguageWriter**. Technology that has previously been available only on expensive multimedia computers (for example, one model IBM Multimedia computer costs more than $5000) is now possible for delivery of language lessons on a low-priced Macintosh more affordable in low-budget foreign language instructional departments.

4.0 SUMMARY

Pedagogically **LanguageWriter** in its current form provides teachers with a generic, flexible software tool for preparing content materials of their choice in both text and sound rather than those prepackaged by others. By contrast, hypertext instructional authoring tools such as Hypercard, which also incorporate text and voice, sometimes along with animation and video, are normally preprogrammed for content by the designers. The combining of text, voice, and now full-motion video in **LanguageWriter** broadens the whole concept of teacher-student interaction as an educational tool in many areas of language pedagogy. And it is all under the content control of the classroom teacher.

Configuring A Computer Lab For Multimedia Applications

Thomas A. Pollack
A.J. Palumbo School of Business Administration
Duquesne University
Pittsburgh, PA 15282
412-434-1639
pollack@duq3.duq.edu

ABSTRACT

The future of classroom-integrated technology is becoming more exciting, complex and powerful. The full integration of technology into the educational environment is critical for future progress. Multimedia is a significant technology component which will have a tremendous influence on future teaching and learning.

Configuring a computer laboratory to provide for the development and use of multimedia materials is a major, and somewhat expensive undertaking. Some of the problems and solutions in such an endeavor will be examined. A possible equipment configuration will also be presented and potential applications will be discussed.

INTRODUCTION

In the Fall of 1992, the A.J. Palumbo School of Business Administration at Duquesne University adopted a mission statement that said, in part, that we will be the "premier teaching institution in the region". The implications of a commitment to such a mission are simple: we must devise innovative teaching/learning processes/strategies to achieve fulfillment. The age-old lecture method will no longer suffice as a presentation method.

Two prominent initiatives underway in our school are cooperative learning and multimedia education. It is our hope that the two strategies will merge and complement each other rather than serve as two separate and distinct methodologies in the educational process. Multimedia education is certainly conducive to cooperative learning techniques. Multimedia applications are well-suited for group collaboration and problem-solving.

COOPERATIVE LEARNING

The United States educational system has come under much criticism in recent years. Some educational leaders claim that the quality of undergraduate education is not commensurate with the needs of today's students. Educational costs continue to rise while attention to the improvement of teaching techniques is ignored by many of our top universities. At the same time, many schools are faced with an incoming student body that lacks a high standard of academic preparedness. Add to this the reality that many of our disciplines require that more material be presented, and we have a serious dilemma. Prospective employers are seeking students who have benefitted from practical and meaningful educational programs. Competition in the job market is exceedingly intense.

Many educators believe that college students do not optimize their learning potential if they are allowed to be passive while they are learning. It is important to get students actively involved in the learning process. Designing cooperative interaction which requires students to explain what they are learning to each other, learn each other's point of view, give and receive support from classmates, and help each other dig below the superficial level of understanding allows students to maximize their learning (Johnson, 1990). Students will be required to coordinate their efforts with others on the job, and they should have the opportunity to practice the social skills required for cooperative achievement. The method of cooperative learning is well-suited to group problem-solving in computer/systems - related courses. It is an active learning strategy that occurs everyday in information systems organizations. Interaction and the inherent learning which occurs allows students to be as "good as they can be". Intellectual engagement in the learning process makes the learning experience personal, an attribute not generally present in the lecture method.

MULTIMEDIA

Multimedia uses the power of the computer to combine voice, music, text, still and full-motion video, and graphics to form an interactive, instructional package. Students and teachers are given the chance to become interactive participants in the learning process, and multimedia provides for intuitive, investigative learning (Kotlas, 1992). William Graves (1993) states that "learners retain about 20 percent of what they hear, about 40 percent of what they hear and see, and about 75 percent of what they hear, see and d". Diana Oblinger (1992a) concludes that, in addition to lowering the overall cost of instruction, students are able to complete courses in one-third the time of traditional instruction when using multimedia, while reaching competency levels of up to 50 percent higher. Oblinger also opines that "multimedia will be to the '90's what the personal computer was to the '80's." Multimedia is viewed as a means of engaging the full range of human senses in the learning process. Graves (1993) endorses project-oriented participatory learning as the most effective form of education.

Market Intelligence, a California-based market research firm reports that the worldwide multimedia marketplace is on the brink of a seven year explosion, peaking at $24 billion in 1998. The forecast predicts that hardware and software sales worldwide will quintuple over the next six years, at a 25% compound annual growth rate (World Multimedia, 1993). A recent Wall street Journal article (4/22/93) indicates that AT&T is strategically positioning itself to compete with Microsoft Corporation in what is described as a "much anticipated but still-evanescent (multimedia) revolution."

All of this industry recognition validates what visionaries have been saying for some time, "Multimedia is not just about a better way to use computers to perform tasks; it's a better way to communicate, a better way to investigate, and it's the natural way that humans learn and interact" (Multimedia Source Guide, 1993-94).

The Institute for Academic Technology (IAT) is a partnership between the IBM Corporation and the University of North Carolina at Chapel Hill on behalf of higher education. Since its 1989 inception, the IAT has grown into a national training and development center focused on the innovative use of computers in education and training (IAT, 1993). This, of course, includes multimedia. The IAT is a creditable source of much information on multimedia.

LABORATORY CONFIGURATION

During construction of our Multimedia Laboratory at Duquesne University, we visited the IAT Auditorium. The auditorium has a horse-shoe seating arrangement that seats up to 55. The facility contains a multimedia-enabled computer system with Internet connection and a ten-foot rear projection system (IAT, 1993).

Two years ago, we decided that we needed a computer laboratory that could serve as a teaching facility. The lab we had in mind had to have certain characteristics, including:

1. Networked, state-of-the-art personal computers
2. Wide range of software offerings
3. Capacity to serve needs ranging from beginners to graduate level group decision support processes
4. High quality projection system
5. Comfortable surroundings

We did an appraisal of what represents current state-of-the-art as well as future applications and decided to commit to a networked computer laboratory with multimedia-enabled computer systems.

Working in conjunction with an architect and somewhat limited by the shape of the space available, we chose a design route in our facility at Duquesne that is considerably different than the IAT facility. We settled on a six-tiered seating arrangement that can accommodate 36 students. Our facility contains a ten-foot flat screen and an instructor's master station.

Our School of Business has been an MS-DOS-based computer facility since 1985. We were somewhat biased toward staying with an MS-DOS/Windows-based hardware solution. However, we considered other possibilities. We tried to accommodate our multiplicity of purposes for the laboratory as we assessed potential solutions. Although we like what the Apple Macintosh platform offered in multimedia, it did not suit our business curriculum. We also looked at workstation solutions, including Digital and NeXT. We kept coming back to personal computers running Windows.

Ultimately, we decided to commit to a DOS/WINDOWS environment with WINDOWS-based applications predominating. We had difficulty arriving at the right mix of hardware and software. We were slightly ahead of market offerings in our configuration plans. Widespread choice of "multimedia kits" was not available. We liked the IBM Ultimedia solution, but we were not impressed with the power or the price of the personal computer included in the kit.

We solicited quotes from major vendors and major clone manufacturers. We also assessed multimedia components for standardization with existing standards as well as hardware compatibility. By mixing and matching, we ultimately selected Digital personal computers with a 486, 50 MHz microprocessor and a NEC Multimedia Kit.

Basically, in configuring a multimedia educational facility, there are two delivery system options to consider. Portable delivery systems can vary in size, depending on the number of devices the user wishes to incorporate. Portable systems can be moved about. Griffin (1992) identifies the following components for a portable delivery system:

1. Overhead projector ($250)
2. LCD panel capable of displaying full-motion video adequately ($5,000)
3. Laserdisc player ($1,000)
4. Two powered speakers ($200)
5. Multimedia computer and monitor with CD-ROM, Digital Audio, Full-Motion Video and Software ($4,000)
6. Overhead projector screen ($100) or light-colored wall (preferably white)
7. Rolling cart ($200)
8. Audio mixer (to mix audio sources - $300)

The total "rough price" for a portable delivery system would be approximately $11,000.

In contrast to the above, a typical fixed delivery system would cost an additional $6,000 to $14,000, depending largely on the type and quality of the projector selected.

We elected to assemble a fixed delivery system, and we spent a considerable amount of time deliberating on a projection system. We considered projectors from Sony, Barco, GE and Esprit. Based on capabilities, price, and quality, we selected the Esprit 2000D Projector at a total installed cost of $13,141.00. This is a three-gun projection system and its clarity is excellent. This system produces workstation quality resolution. Prices for comparable units ranged in cost from about $10,000 to well over $20,000.

Platforms available for delivering multimedia have mushroomed since the summer of 1992. Spring 1993 offers an overwhelming array of multimedia-ready microcomputers. A multimedia-ready computer needs sound, graphics, a CD-ROM drive and a large hard disk. Standards have been recommended by the Multimedia PC Council (MPC) who stand behind the MPC logo. The MPC-established minimum standards for a DOS-compatible machine are as follows (Multimedia Source Guide, 1993-94):

1. 386 SX-class IBM compatible
2. 4MB of RAM
3. S-VGA 256 color graphics
4. 350ms-600ms CD-ROM drive
5. Sound card with 16-bit audio
6. 80MB hard disk

The Duquesne University configuration per workstation is as follows:

1. DEC 486-50MHz PC
2. 4 MB of RAM
3. S-VGA 256 color graphics
4. 300Kbytes/second CD-ROM drive with 680MB/disk capacity
5. "Media Vision" sound card with a SCSI interface and a 8 bit audio record capabilities includes:
   - 44-pin connector that attaches to the Audio Board
   - 50-pin SCSI connector that attaches to the CD-ROM Reader
   - 15-pin connector that connects with a compatible joystick
   - Left and Right Line In cables which connect to external audio devices
   - Left and Right CD In cables that attach to the CD-ROM reader
   - MIDI Input and Output cables

6. 110MB hard disk
7. 2 Amplified speakers
8. Stereo headphones
9. Microphone

All of the multimedia items were included in a NEC Multimedia Upgrade Kit and the cost was $900 per kit. The DEC 486-50MHz PC's were volume priced at $2,500 apiece.

The 36 multimedia-enabled computers in the Duquesne Multimedia Laboratory are networked via NOVELL ($12,000) and also have access to the Internet. The cost of the file server for this lab was approximately $9,000. The lab also has a five-speaker surround-sound stereo system with receiver ($700). To enable users to capture video clips, a video spigot adaptor card was purchased and installed on the instructor's master workstation at a cost of $500. The development software in use is Asymetrix Multimedia Tool Kit which was included as part of the Multimedia Upgrade Kit. The software lends itself to high productivity and is very user-friendly.

Items to be added for this summer include the following:

1. Stereo VCR ($500 est.)
2. Computer-controlled laser disk player (Pioneer, $700 est.)
3. CD-ROM-R record device (Pinnacle Micros, $4,200 est.)
4. Read/write magneto-optical drive to CD-ROM type data until CD-ROMs are pressed (Pinnacle Micros $3,200 est.)
5. Connectivity to cable television and satellite feeds.

The installation of these components will complete the creation of a multimedia laboratory that is unparalleled in our regional educational community. This will allow for a multitude of multimedia and video-based education options. Creativity is the only limitation. We have begun the faculty training process. We will have a 36 station laboratory valued at approximately $175,000, and we must assure that interested faculty have the knowledge necessary to utilize the facility.

SUMMARY

I have attempted to make a case for the fusion of the computer and various media formats as a means of revitalizing the instructional process. Multimedia offers us an opportunity to change the way we teach and, at the same, time change the way students learn. Coupled with cooperative learning techniques, the learning process can become truly interactive. By nature, computers make the learning experience for students richer and more interactive (Oblinger, 1992b). Technology offers many benefits to the educator and the student. Oblinger (1992b) cites the following advantages of using computers in teaching:

1. Self-pacing by the student who wishes to spend more time on a topic.
2. Teacher - student dialogue by working through the system.
3. Achieving mastery without humiliation or discouragement.
4. Teacher in top form, even on an "off day".
5. No media limitations.

6. Less time required - armed forces studies reflect a time savings of about 30%.
7. Simulations are possible for dangerous and expensive real-life situations.

There are numerous opportunities to use multimedia-based instructional materials. Ranging from business schools to law schools to medical schools, multimedia education efforts have been effective. Instructors and students do not have to be computer experts to be successful. An instructor who understands fundamental cooperative learning principles and fundamental multimedia concepts can transform courses from routine to spectacular. Creativity, imagination, and time are our only constraints.

REFERENCES

Course and Services Guide. (1993, Spring). Chapel Hill: University of North Carolina, The Institute for Academic Technology.

Graves, William H. (1993, Spring). Multimedia Manifesto. Campus Tech, pp. 9-13.

Griffin, Steve (1992). Delivering Multimedia in the Classroom (TRP-01). Chapel Hill: University of North Carolina, The Institute for Academic Technology.

Johnson, D. W., Johnson, R. T., & Smith, Karl A. (1990). Cooperative Learning: An Active Learning Strategy For The College Classroom. Baylor Educator, 15(2), 11-16.

Keller, John J., (1993, April 22). AT&T Is Trying Hard To Get a Major Role In Multimedia Future. The Wall Street Journal, p.1.

Kotlas, Carolyn. (1992). Evaluations of Multimedia Technology in Higher Education: Bibliography (IRG-11). Chapel Hill: University of North Carolina, The Institute for Academic Technology.

Multimedia Source Guide (1993-94). (Suppl. 1993-94). The Technological Horizons In Education Journal.

Oblinger, Diana (1992a). Introduction to Multimedia in Instruction (TRP-03), 1-9. Chapel Hill: University of North Carolina, The Institute for Academic Technology.

Oblinger, Diana (1992b). Teaching and Learning with Computers (TRP-06). Chapel Hill: University of North Carolina, The Institute for Academic Technology.

O'Connell, Brian (1992,July). Distributed Multimedia. DEC Professional, 11(7), 36-44.

World Multimedia Hardware and Software Markets (1993). Mountain View, California, Frost & Sullivan Market Intelligence.

The Use of Laptops by Professors in the Computer Science and Systems Classroom

R. Waldo Roth, Chairman
Computing and System Sciences Department
Taylor University
Upland, IN 46989-1001
317-998-5269
wlroth@tayloru

ABSTRACT

Taylor University recently had the good fortune of internal funding to provide a laptop computer as a teaching and productivity aid for each faculty member in the Computing and System Sciences (CSS) Department. This presentation will review the criteria for hardware and software (H/S) selection, the equipment chosen, the applications utilized thus far, and the strengths and limitations of the concept of the "laptop in the classroom". The specific implementations of some of this support equipment and its use in the CSS Department at Taylor will be described and/or presented. [Note: Whenever specific H/S is mentioned or described, this is done for illustrative purposes and should not be interpreted as an endorsement of any of these products by either the CSS Department or Taylor University.]

REASONS AND CRITERIA FOR SELECTIONS

In the summer of 1992 the CSS Department at Taylor University decided they needed new portable, laptop computers to improve personal productivity plus expand course and classroom flexibility. Among the reasons and local issues at Taylor which led to this decision were:

- Teaching in many different classrooms with a variety of overhead computer projection systems including none in some classrooms. (Flexibility)
- The complexity and cost of faculty having different machines at home and office. (Productivity)
- The need for more effective visual aids with the current generation of students. (Better Pedagogy)
- The requirement to move to a graphics user interface (GUI) environment for most faculty lectures and work. (Currency)

Of course, the biggest stumbling block in working through this decision was the cost of the equipment. After reviewing a variety of possibilities, it was concluded that $3,000 per faculty member could provide more than adequate hardware and software to accomplish this goal. Through a creative and unique internal grant, the necessary funds were made available.

The criteria for the equipment to be selected included the following general and specific criteria:

SPECIFICATIONS

* The systems chosen had to support a good GUI (compatibility mandated we not consider Apple Macintosh equipment at this time).
* Since our department is a heavily IBM compatible and UNIX workstation environment we wanted machines to support Windows 3.1. This was a natural transition for us from our previous DOS environment.
* We decided not to specify X Windows support on these machines. However, even one year ago a portable SPARC/UNIX workstation was an option (at significantly greater cost).
* The software needed to include a good presentation graphics package for making classroom and conference presentations. (This is the basis for this ASCUE Conference presentation and demonstration.)
* The initial ceiling cost was $5,000 per faculty member including classroom projection support, and we were fortunate to come in slightly under this limit.
* Although we desired built-in color monitors, we found this pushed the price above the limit defined above. We chose units which drove VGA Plus monitors, but themselves had only 64 gray-scale levels (EITHER/OR) which has worked out reasonably well.
* It was decided to select identical hardware for all faculty members to make backup and support easier.
* Software options would be customized for each faculty person with common items purchased at quantity prices and coordinated by our technical support faculty and secretary.
* Large amounts of hard disk space—at least 200 megabytes.

SOFTWARE REQUIREMENTS

These selections have evolved over this academic year with additional software purchased frequently to support new faculty and course needs and challenges. Some additional funding has come from the department for these purchases, but personal faculty funding has occurred as well. (See Appendix A for the list of individual selections.)

HARDWARE SELECTED

We chose Standard Computers from Sager Midern Computer, Inc., 18005 Cortney Court, City of Industry, CA 91748. We have few regrets too date, but there were many other choices which were also quite competitive. We have since located other satisfied users of the Sager laptops.

The machines came in at under $3K, having 8 meg of RAM, and 240 meg of disk space. One of our few disappointments was that the systems sent did not include built-in mice or trackballs.

SUPPORT HARDWARE ACCESSORIES

A variety of mice or trackballs were evaluated and a different selections were made based on individual facility preferences. This has worked very well, although built-in trackballs would have saved non-trivial inconvenience. Also purchased were docking stations, external keyboards (not used by all), external/desktop 15 inch VGA Plus monitors which have proved essential for long hours of work.

CLASSROOM SUPPORT

At the last ASCUE Conference our plan was to buy one color LCD panel at $8,000 and one grey- scale unit for much less. We were surprised to find at least one vendor, Boxlight Corporation of Poulsbo, Washington, offered a very effective color display panel at about $3500. Therefore we purchased two such color panels. This has proved to be a wise decision as these two units have successfully supported five full-time faculty and occasional part-time faculty use.

SUCCESS OF THE PROJECT TO DATE

This new teaching technique has been an exciting and very successful adventure for the CSS Department faculty at Taylor. Among the merits of this program have been the following:

* Significant faculty gains in productivity. The efforts of lecture redesign, plus the hauling of the units to class, office and home have really had great payoff. Faculty morale, productivity, and in my case organization have been dramatically improved.

* The equipment has been very valuable both in computer classes (demonstrating programming studies and teaching end-user software), system analysis classes (CASE demonstrations), and general presentations such as the one given at this conference. These laptops have been less useful in mathematics oriented classes (such as Discrete Structures) unless end-user packages are involved (such as in the Management Science classes or other courses using Mathematica or Maple).

* Students have been pleased with the results. In fact, students have been able to make selective use of this equipment for presentation to external agencies such as Eli Lilly (course research presentations) and Argonne Laboratories (student scientific reports).

* To date, we have not found any other equipment we would have selected instead of these machines. There are no regrets in our decision and only a few things we would change as described below.

LIMITATIONS AND A WISH LIST

The few modifications we would like to made in the choices and uses of this equipment include (in descending order of importance):

1) A requirement for new, high intensity overhead projectors. They are not easy to find, select, or afford. The current solution leaves students in a partially lit classroom and "wow, do they get sleepy!"

2) Battery life of less than 1 1/2 hours is barely adequate. Newer machines advertise greater performance which would be helpful, especially with some faculty members' schedules. AC power is usually accessible, if not conveniently, so when required the major impact is on set-up time and faculty patience.

3) Built-in trackballs (or equivalent) would definitely be preferable.

4) Although the units do drive color displays, the lack of a built-in color monitor has been a slight frustration (especially the either/or type of display). Despite a significant price drop. the cost would probably still make this a necessary decision even if the choice was being made today. This has not been a major problem, but an inconvenience.

5) One faculty member has run into the 240 megabyte hard drive limit, but he is has not yet used standard compression software such as Stacker or that included with DOS 6.0.

6) One faculty member would prefer and could use a SPARCstation laptop although this solution is still very expensive.

ADDITIONAL OBSERVATIONS

1) This has been a very exciting academic year for our department since the 30% of the time we are using these systems in the classroom has been very rewarding and well worth the 5 to 10 minute set-up time.

2) Presentations to outside organizations from high schools to the National Science Foundation have been impacted by the use of this equipment.

3) Presentation packages like Lotus Freelance for Windows 2.0 are extremely easy to utilize effectively and quickly.

4) The author has found this technology extremely useful in organizing himself. My lectures are 70% prepared already for 1993-94 and I know which diskette they are on. No more lost or misplaced overhead foils!

5) Classes from computer literacy (demos of laboratory software and multimedia) to sophisticated CASE tools (i.e., POSE), artificial intelligence software (i.e., ECLIPSE), graphics software (PHIGS based), and simulation products such as SIMSCRIPT II have been made more attractive and far more instructive.

CONCLUSION

It is an obvious fact that cost is still the number one consideration and limitation in such a decision. However, costs are coming down, and all of us should find creative ways to fund such personal productivity tools, to enhance our classroom instruction, and to increase student involvement in our lecture/presentations. Whether by using multi-window grading solutions, multimedia development and presentations, or just for personal projects at home, Taylor computer science faculty instruction (and lives) will never be the same.

APPENDIX A

SOFTWARE IN USE

A partial list of the software in use by various faculty in the department in descending order based on the number of installations (out of 5 possible):

Software Classification	Number of Faculty Using	
Presentation Graphics	(5)	
Spreadsheets	(5)	
Word Processing	(5)	(Not a Windows version yet)
Programming Languages	(n)	(Individual requirements and preferences)
PC-SOLVE Software	(3)	(A convenient problem solving tool)
On-Line Bible	(3)	
Scheduling Software	(2)	
Management Science Packages	(2)	
CASE Software (POSE)	(1)	
Worldmap and Automap	(1)	
GIS Software	(1)	

And the list goes on...

The Senior Capstone Course for Computer Science and Computer Information Systems Majors

Donna Satterfield
Franklin College of Indiana
501 East Monroe
Franklin IN 46131
317-738-8252

Franklin College requires all majors to take a capstone course. For Computer Science (CS) and Computer Information Systems (CIS) majors, the capstone course takes the form of a two semester sequence. The Fall semester course earns students 1 credit hour and the Spring semester course 2 credit hours.

This two semester sequence requires students to design and develop a computerized system. Students are responsible for all phases of the life cycle, setting up all walkthroughs, providing documentation, etc. At the conclusion of the course, students conduct a formal presentation of their project with the sponsor, faculty, computing staff and an outside evaluator. In addition to the systems project, students are also required to perform maintenance on an existing system. This is usually a system that was set up by seniors in a previous year.

BACKGROUND INFORMATION

When I came to Franklin College, 4 years ago, the senior seminar course was a 2 hour course offered during the Spring Semester of the senior year. Students were expected to design and develop a computerized system AND were prepared for the senior oral comprehensive exam. The course had several problems as identified to me by the faculty member teaching the course at that time. Among them were:

1. Students were expected to design and develop a computerized system within a 14 week period.

2. Students often did not complete the system until finals week which left very little time for user testing of the system.

3. Student developed systems were often unreliable, due to the short amount of design and development time allowed and the very limited testing performed on the system.

4. Computer services center staff were reluctant to become involved in the maintenance of these projects as that often required a rewriting of the system and the center operates with a very minimal staff.

5. There was a feeling that the senior oral comprehensive exam was not an effective indicator of the degree of knowledge students had gained from their education.

GOALS FOR THE REVISED SENIOR SEMINAR

One of my charges when I came to Franklin was to help with the revision of the CIS program. We realized that the program did not meet the needs of our majors as well as it could. The revision of the CIS program began with a review. This was accomplished by sending out questionnaires to alumni, our Computer Advisory Board (a group of business men and women who advise us concerning their information needs and the direction they think the program may need to take in the future), and other colleges. The results were evaluated and our Advisory Board met with us to discuss our findings. Two of the things we discovered our program lacking in were technical writing and presentation skills.

We felt that the Senior Seminar course could be revised to help better prepare our students in these two areas. Specifically, we wanted to give our students more experience with technical writing and more opportunities to practice their presentation skills. At the same time that the course was being revised to offer students these experiences we decided to try to find a way to solve the five problems which had been identified by the instructor.

REVISIONS

In order to solve the limited design development time, the course was expanded to be a two semester sequence, from the previous one semester course. By making the Fall semester course one credit hour and leaving the spring course two hours, we were able to be consistent with other departments on campus whose senior capstone courses were three credit hour courses. This change solved many of the problems we had identified with the course as it was originally designed.

By having two semesters in which to design and develop a system, we could require the student to turn the system over to the user earlier in the spring semester. This would provided the user with more time for testing the system. The expanded design and development time along with more thorough testing would, we hoped, result in more reliable systems.

Our attention then turned to the reluctance of the computer services center staff to perform maintenance on student developed systems. While more complete design, development, and testing should result in more maintainable systems than had previously been possible, the size of the computer center staff was still a deterrent to their performing maintenance. It was decided that, since the course had been expanded to cover two semesters instead of the original one, we could require additional work of the students. After all, they should not receive three hours of credit for the same amount of work that previously had received only two just because they had four extra months in which to do the work. So, a maintenance component was added to the senior capstone course.

This maintenance project would serve to relieve the computer center staff from any commitment to maintain student developed systems while at the same time provide the sponsors (persons for who the system was developed) with the assurance that there would be someone available to make necessary changes to the system over time. As an added benefit, students would now have the opportunity to work on maintaining a current system that they did not develop. This, we believed would be a plus as most of the work performed in industry is maintenance in nature rather than development.

These changes should solve many of the problems we had identified with the course but still did not address the senior oral comprehensive, whether it was a valid exam or not, nor had we addressed the need for better technical writing and presentation skills. We decided that since the oral comprehensive exam was not a valuable component of our program, we could replace it with a formal presentation of the senior project at the end of the spring semester. This would give students the opportunity to perform a post-implementation review with the sponsor and certain staff and faculty members (selected by the instructor). During this presentation, students would be evaluated based upon their ability to properly conduct the presentation, answer questions, and relate information regarding the project to a group of people with a wide range of knowledge (or lack of knowledge) about the project.

It was also decided that students would be expected to perform informal walkthroughs of their project at the conclusion of each of the life cycle phases. It was hoped that this would help them to better understand the project they were working on, to insure it met the user's needs, and to provide a less-threatening environment in which to practice their presentation skills.

In addition to the walkthroughs and formal presentation, students would be required to write a formal report at the conclusion of each of the life cycle phases. These were to be handed out to participants in the walkthroughs prior to the walkthrough. These were to be graded on content, grammar, terminology, completeness, etc. and would be the basis for the informal walkthrough.

Finally, students would be required to write complete user's manuals and technical reference manuals. The user's manual would be turned over to the user along with the system and would be used as a tool in training the user on the new system. The technical reference manual would be presented to the instructor and specified evaluators for review prior to the formal presentation. This manual would then be used by other students in maintaining the system in the future. Students would then be required to update these manuals as appropriate whenever maintenance was performed on the system.

THE CURRENT SITUATION:

We are currently in the third year of changes to the course. Not all changes could be made at the same time. Students had to be taught how to perform presentations and walkthroughs, how to write user's and technical manuals, the steps in the system development life cycle, etc. Also, I had to learn how to organize the course so that all of the new features and requirements could be accommodated.

The first year of the revision, students were required to turn their systems over to the user by mid-term. This policy still remains. The students were not organized enough to get this done and therefore, projects were not turned over until approximately two weeks before the end of the semester. This was an improvement and did allow for some testing time.

During this first year, students were required to write a user's manual but were not required to write a technical manual or to perform a maintenance project. Each student was required to conduct a formal presentation during the last week of classes. The only walkthrough implemented that year was at the conclusion of the fall semester.

It was discovered that the students had made several incorrect assumptions about the project and had to fall back and redesign some components. Also, the lack of regular walkthroughs lead to less than adequate contact with the sponsor and delays in completing the project.

During the second year adjustments were made to require students to perform walkthroughs at the conclusion of each of the life cycle phases. Formal documentation was not a requirement for these walkthroughs. Students passed out some documentation during these walkthroughs but many needed items of information was left out (like system objectives). Again, students were late in turning over their projects to users. But they did manage to get them in a week or two earlier. User manuals and "technical" reference manuals were required at the formal presentation.

Also during the second year, students were required to perform a maintenance project. These were performed informally requiring minor modifications and very little documentation changes. No deadlines were set on these projects except that they had to be completed by the end of the year.

This year, year three, all projects were turned in on time (by mid term) complete with user's manuals. All walkthroughs were conducted on schedule with appropriate formal reports delivered to walkthrough participants at least two days before the walkthrough. Technical manuals were to be written and delivered at least one week prior to the formal presentation.

Maintenance projects, again this year were informal with no deadlines, but required documentation and a written review of the maintenance work performed.

THE FUTURE:

We expect that in the next couple of years the maintenance component of the course will be as fully developed at the project component. Also we will begin to bring in an outside evaluator to review the documentation (formal reports, user's manuals, technical manuals, etc) and to evaluate presentations.

In addition to these changes, we will be offering students who plan to attend graduate the option of doing a year of research. This part of the course is, as yet, undeveloped. We will have our first student performing research during the next year.

We are pleased with the progress we have made with our senior seminar course and are looking forward to future modifications to improve it even further.

Computer Literacy - An Electronic, Paperless Course

Donna Satterfield
Franklin College of Indiana
501 East Monroe
Franklin IN 46131
317-738-8252

Computer Literacy courses used to generate a large volume of paper and require the submission of large numbers of student disks. Now with the aid of PCmail (VAX mail) students can not only submit their assignments electronically, but exams and grading are also accomplished electronically. Having the electronic connection increases communication between the instructor and students and among the students themselves. This has been accomplished at Franklin College over a two year period.

HOW IT ALL STARTED

The impetus for this paperless course came out of our Academic Computer Computing committee. We were looking at how to improve student use of computers on campus. We had noticed that several schools were moving to a "paperless classroom" and wondered how it would work at Franklin College. As a member of the committee who utilized computers extensively in my courses, I volunteered to use Computer Literacy for Business as a "pilot" course.

The Literacy course seemed to be the logical place to begin as with today's networking capabilities, students need to be exposed to more advanced uses of the network than simply retrieving software from a larger computer and pulling it down to a smaller one. Which, up until this point, was what they did. I decided that if I was going to do this, the "paperless" classroom should be phased into.

This decision was arrived at for several reasons. First, I had not used electronic mail myself except for the occasional VAX mail message to our network supervisor or a "phone" conversation with upper level students as they were working in the lab. Therefore, I had to learn to use it. Second, I realized that changing to a "paperless" class would require changes in the structure of the course. Not only would I have to fit in the teaching of electronic mail, but I would also have to reevaluate how I communicated with my students and graded papers. In addition to these changes, I would have to implement some means by which to encourage students to communicate electronically with each other. Finally, as happens every time something new is tried on the computer, I knew it would be a time consuming project and would, most likely, cause some problems. Either from my lack of knowledge, or changes that needed to be made in the software or hardware that were not foreseen, or from (believe it or not) problems caused by students using the new system.

I decided to phase in this new "paperless" classroom in three steps. The first step was to become a "diskless" and "paperless" assignment grader. This seemed to be the most logical place to begin for two reasons. Reason 1 being that it usually took several days to grade assignments. This required students to either have extra disks (not a bad thing) or to not do any work until they got their disks back (this is what usually happened). Also, grading from students' floppy disks seemed to be inviting viral infection into my system. Reason 2 was the fact that this would be the simplest change to make. Very little change had to be made in my course or the network to allow for electronic submission of assignments rather than floppy disk submission.

The second phase was to make the course truly "paperless". This would require students to take exams on the computer instead of paper. We phased this in during the Fall semester at which time I teach two sections of literacy. I gave paper exams in one section and computer exams in the other. Then a comparison was made of the quality of the exams to see if computerized testing might serve any purpose other than reducing the amount of paper used.

With this phase complete, the course was basically paperless. The only handouts given in the class is the course syllabi on the first day of class. The only paper turned back in is an occasional printout of an assignment, just to make sure they know how to print. This left the third phase to be introduced. During the third phase I was looking for a way to involve students in electronic mail other than for submission of assignments. I wanted them to communicate with one another electronically. To accomplish this I made a couple of revisions in the structure of the course. Prior to this all assignments were individual assignments, now I assign half of the assignments as individual work and half as group work. This helps a little, but the biggest change is in test building. Half of the exam is comprised of True/False and Multiple Choice questions. Previously these came from a test bank (the purpose of which was to encourage reading of the text rather than sole reliance on lecture). Now I assign each student to a particular section of the text to write questions on. They submit these questions to me and I choose the questions I wish to use on the exam. They may also submit them to one another for study. The idea is "if you don't send me your questions for this exam, you won't get mine for the next". This starts them communicating, then they build on this with questions and comments to one another about the exam.

DEVELOPMENT & SYSTEM CHANGES

With the assistance of our academic support coordinator, Matt Stuve, we set the course up so that students could submit their assignments electronically. This required a great deal of work on the part of our Network Manager. Students in the course already had accounts on the VAX computer but were only using the VAX (as they perceived it) as large secondary storage device which held the software packages they were using. These students were not equipped to deal with VAX mail and we needed some way to submit documents without changing their original form. For example, WordPerfect documents needed to stay WordPerfect documents not ASCII, and Spreadsheets needed to remain spreadsheet form, etc.

Another factor to be considered in setting up this electronic course was the time needed to teach students to submit their assignments electronically. The literacy course is taught as 1/3 applications. During this third of the semester (1 hour per week for 14 weeks) students are expected to become knowledgeable in wordprocessing, spreadsheets, database processing, and DOS. To fit in time for additional training in electronic mail would be very difficult. Therefore, the method of transmission had to be very simple, preferably menu driven.

PCmail offered the solution to both of these problems. First Pcmail allows documents to be transmitted as binary attachments to a mail message. The document can then be extracted from the message and accessed through whatever application software it was created in. Secondly, PCmail is a menu driven system which can be taught quickly. The basics of PCmail can be covered in a one hour class session with suggestions for using more of the features sent to students as mail messages.

The final change that had to be made occurred as a result of problems encountered with testing. I needed someway to send make exams available to students at the beginning of class but not before class started. In the beginning, the only way to do this was to send them as binary attachments to mail messages. Students were then required to extract the attachment, go into a wordprocessor and then retrieve and take the exam. This often took anywhere from 10 to 15 minutes just to get started. If there was any problems, it could take longer. Obviously when I had designed an hour long exam, students could not afford to spend a fourth of their time or more just getting started.

The solution to this problem was fairly simple and has proved to be a valuable tool for many things. A procedure was set up whereby faculty members can UPLOAD files to students. We have had a DOWNLOAD procedure for years but to upload files required submitting the file to computer center staff at 24 hours in advance so that it could be put on the network. Of course a truly organized person would not have a problem with this, but I often don't have the exam completely ready until nearly exam time. With the UPLOAD feature, I can walk into class. and have the file ready and loaded into students computers within 5 minutes. This is little more than it takes to pass out an exam.

BENEFITS AND PROBLEMS

The biggest problem is that this system has increased the amount of time I spend on the course. That is of my own choosing. I allow any student submitting work early to make corrections and resubmit. This requires extra time. With the old system, maybe 5% of the students would take advantage of this, now nearly 50% do. That takes time. Some submit their work as much as three times before they are satisfied with their grade. Also, students feel more comfortable communicating with me. They send me messages when they are having problems with an assignment and I make sure to respond within 24 hours. They also let me know when they think I have been unfair or they have a question about grading. They rarely did this in the past. Somehow, being able to sit down and type out your frustrations with someone seems easier than waiting until you can meet with them face to face. By the time they can catch me in my office to talk to me they have usually talked themselves out of discussing with me how they feel. (You know the old grade fear thing). I even had a student send me a two page message explaining why he thought he should get an extra two points on a exam question. He go the points.

The benefits are numerous. By submitting assignments early, students are learning more. Most of us learn more from our mistakes than we do from the things we do right. They learn as they correct incorrect assignments. The increased communication lets me know how students feel, when they do not understand and assignment or topic and allows me to clarify problems.

Also, if I have students ask about something, I can send a message to the entire class explaining it if I think it is necessary. That way they don't have to wait from 2 to 5 days to find out needed information. The information channel is more open.

I find exam answers are more complete and I give better responses to missed questions than I did previously. And, as I said earlier, students let me know when they don't understand why I scored them the way I did. This provides a great additional teaching tool for me. As an example I had one student this year who made a 59 on his second exam. He sent me a message expressing his concern. I went through the exam again and explained in detail why I graded as I did. On the next exam he scored and 88. What was the difference, he knew what I was looking for in discussion questions because he had taken the time to question earlier. Of course if students are still unhappy . they know that I am open to discussion and will usually come by my office to talk with me. I actually have more students in my office now than I did before!

The final advantage I will mention. is the quality of assignments and exams. I ask more comprehensive questions on the exam than I did before. the student supplied questions are actually more difficult than the test bank questions but they don't complain. I think exams are a teaching tool as much as an evaluation tool and therefore, the more comprehensive questions gives me the opportunity to teach more. Also. the application assignments given are more detailed. Rather than just covering simple concepts, I cover some advanced topics. Basic concepts are covered in class and individual assignments cover these, but for group assignments, I send them hints on more advanced topics and give them assignments to utilize them. For example, instead of just working with a spreadsheet, students must take information from a spreadsheet. print it to a file, and incorporate it into a wordprocessing document which discusses the data. Database data is exported to spreadsheet or imported from spreadsheet.

These are things I could not have done without the ability to communicate with my student on a regular/frequent basis. True these things require more of my time, but at Franklin the emphasis is on teaching and any method which improves teaching is worth the extra effort it takes.

Using Hypermapping and Embedded Cognitive Strategies in the Development of Educational Hypermedia

Gary J. Senn
University of South Carolina-Aiken
Ruth Patrick Science Education Center
171 University Parkway
Aiken SC 29801
803-648-6851 ext 3558

Phillip B. Horton
Florida Institute of Technology
Science Education Department
150 W University Blvd.
Melbourne FL 32901

There is great potential and opportunity for hypermedia in education. As hypermedia continues to grow in popularity, research is needed to evaluate its effectiveness and to identify effective implementation strategies in the educational setting. Educational researchers have suggested a number of areas for further research. For example, Heller (1990) and Bowers and Tsai (1990) have suggested that studies must focus on the design and development of hypermedia. Hypermedia development includes information presentation on the screens and navigational aids that facilitate the use of hypermedia by students.

A major problem with hypermedia is that users become disoriented. The disorientation results in users becoming lost in "hyperspace" (Conklin 1987). A leading remedy is to use a system map. The map can provide visual cues indicating where the user is within the hypermedia environment. The map can also show users where they have been and provide choices for where to go next (Kinzie & Berdel, 1990).

Another recommended remedy for hyperspace disorientation is to have cognitive learning strategies embedded in the software to provide navigational aid to the user (Jonassen, 1991). The embedded strategies provide hints or suggestions that may enable the user to experience more from the hypermedia.

This study sought to determine the effects of hypermapping (system maps) and embedded cognitive strategies on biology achievement and completion rate of hypermedia courseware. Additionally, the study explored the relationships of ability level versus student achievement and completion rate.

HYPERMEDIA CONTENT

The content of the hypermedia courseware used for this study is titled, "The Ecology and History of the Great Smoky Mountains National Park" (Senn 1992). The software was in the form of a "stack" written using HyperCard for the Macintosh computer system. There are six sections of the tutorial: abiotic factors, biotic factors, vegetational communities, ecological concepts, humans in the park, and park features. The main theme is ecology with a small amount of information regarding the history of the people in the region and the establishment of the Great Smoky Mountains National Park in Tennessee and North Carolina. Besides the information indicated by the six main section titles, the tutorial also includes information on human impact on the environment, exotic species, endangered species, biomes of North America and biosphere reserves.

RESEARCH DESIGN

The design of this study was a pretest-posttest control group design. There were four treatment groups in this study. The control group received the hypermedia science tutorial. The hypermapping group received the tutorial and the hypermapping enhancement. The suggestions group received the tutorial and the embedded cognitive strategies enhancement. The combination group received the tutorial and both enhancements. The treatment was limited to two 50 minute periods. 261 students were randomly assigned to one of the four treatment groups.

Students were also grouped according to prior achievement. Prior achievement or ability was determined from the pretest which consisted of the grades the students earned during the grading period prior to this study. Students in the high ability group received a grade of A or B, students in the medium ability group received a grade of C, and students in the low ability group received a grade of D or F.

INSTRUMENTATION

The biology achievement posttest was a paper and pencil test that consisted of fifty multiple choice items with four choices per item. The instrument to measure the student completion rate was learning path data. As students used the tutorial, their progress or path was recorded. There were 94 cards available for the students to visit. The completion rate score was the number of cards out of 94 that each student visited.

HYPERMAPPING ENHANCEMENT

The hypermapping enhancement consisted of five cards with a series of flow charts that provided a map of the system. When students accessed the hypermapping enhancement, they were given three pieces of information. The button representing the card they just left flashed on and off. All of the cards that the student had already visited were highlighted and all of the cards that the student had not visited were not highlighted. When the student was finished with the map, he had the option of returning to the card he had just left, going to another map, or going to any one of many related cards.

EMBEDDED COGNITIVE STRATEGIES ENHANCEMENT

The embedded cognitive strategies enhancement consisted of suggestions to the students that had two basic forms. The first type of embedded cognitive strategy provided information on terminology that might be new to the student. The second type of embedded cognitive strategy provided the student with links to related topics.

DATA ANALYSIS

The a priori alpha level for significance testing was set at .05. For this study, the researcher desired to detect a medium effect size (.15 standard deviations) and had a sample of 261 students. Based on this information, power was determined to be greater than 0.99 for this study (Cohen & Cohen, 1983).

The data were analyzed using ANOVA. There were two factors of interest in this study, treatment groups and ability groups. Each of these factors was analyzed versus the two dependent variables, posttest and completion rate scores. Both factors were also compared in a two factor ANOVA to determine if there were any interaction effects. No interaction effects were detected.

The ANOVA for treatment group membership versus posttest scores demonstrated that there were no significant differences in student achievement between the treatment groups after completion of the hypermedia tutorial ($F = 0.19$, $p > .05$).

The ANOVA for ability group membership versus posttest scores resulted in a significant Ftest ($F = 3.83$, $p < .05$). There was a significant difference between the high and medium ability groups and between the high and low ability groups. There was not a significant difference between the medium and low ability groups. Low ability students received as much benefit from the hypermedia courseware as the medium ability students.

The ANOVA for treatment group membership versus the completion rate scores yielded a statistically significant Fratio ($F = 3.507$, $p < .05$). The means for the control, hypermapping and combination groups were all within two points of each other and did not reveal a significant difference. The mean for the suggestions group was significantly lower than the other three groups.

One explanation for this finding is that students in the suggestions group might have been disoriented when they used the extra linkages available with the embedded cognitive strategies enhancement. The fact that students in the suggestions group had lower completion rate scores than students in other groups is consistent with reports from other researchers that users become lost in hyperspace (Heller, 1990; Richards, Chignell & Lacy, 1990; Conklin, 1987; Kinzie & Berdel, 1990).

The suggestions group scored significantly lower on completion rate than the combination group. The only difference between these groups was the hypermapping enhancement. If the provision of more links causes students to become disoriented and unsure whether they had visited certain sections of the courseware, then the addition of hypermapping seems to reduce this effect.

The Fratio from the ANOVA for ability level groups versus completion rate demonstrates that there were no significant differences between the ability level groups ($F = .749$, $p > .05$). Completion rate scores were not related to ability levels.

IMPLICATIONS

1. Hypermedia developers should consider including hypermapping enhancements in their software. When there are many branches and links between cards in hypermedia courseware, users may become disoriented and become lost in hyperspace. The suggestions group in this study had the largest number of linkages available and the largest potential for disorientation. The combinations group had these same linkages but also had the hypermapping enhancement available. The mean completion rate score for the combinations group was significantly higher than the suggestions group. These findings suggest that hypermapping can help users overcome their disorientation and take advantage of more of the hypermedia material.

2. Hypermedia developers should consider using embedded cognitive strategies more often. The findings of this research support that students will take advantage of embedded cognitive strategies if they are made available. It would be best if developers who use the embedded cognitive strategies also include a hypermapping enhancement.

3. More experience with hypermedia courseware might help increase the achievement levels of low ability students. In this study, low ability students receive as much benefit from the hypermedia tutorial as medium ability students.

REFERENCES

Bowers, D. & Tsai, C. (1990, February). HyperCard in educational research: An introduction and case study. Educational Technology, pp. 1924.

Cohen, J. & Cohen, P. (1983). Applied Multiple Regression/Correlation Analysis for the Behavioral Science (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.

Conklin, E. J. (1987). Hypertext: An introduction and survey. Computer, 20(9), 1741.

Heller, R. S (1990). The role of hypermedia in education: A look at the research issues. Journal of Research on Computing in Education, 22(4), 431441.

Kinzie, M. B. & Berdel, R. L. (1990). Design and use of hypermedia systems. Educational Technology Research and Development, 38(3), 6168.

Jonassen, D. H. (1991). Hypertext as Instructional Design. Educational Technology Research and Development, 39(1), 8392.

Richards, T., Chignell, M. H., & Lacy, R. M. (1990). Integrated hypermedia: Bridging the missing links. Academic Computing, 4(4), 2426, 3944.

Senn, G. J. (1992). The ecology and history of the Great Smoky Mountains National Park [Computer program]. Melbourne, FL: Florida Institute of Technology, Science Education Department.

Computer Technology For Elementary and Middle School Students

Gary J. Senn
Jeffrey M. Priest
Ruth Patrick Science Education Center
University of South Carolina at Aiken
171 University Parkway
Aiken, SC 29801
803-648-6851 ext 3558

Computer Technology For Elementary and Middle School Students (CTFEAMS) is a program designed to help teachers incorporate computer technology into their classrooms and to provide handson computer experience to students in Aiken County South Carolina. This program was made possible through a grant to the Ruth Patrick Science Education Center (RPSEC) by the South Carolina Commission on Higher Education.

The RPSEC was established to provide a permanent program to enhance and promote high quality science and mathematics education within the Central Savannah River Area (CSRA) of South Carolina and Georgia. The RPSEC is a cooperative community effort including the University of South Carolina - Aiken, local school districts, and the private sector. The RPSEC offers educational institutes, workshops, courses, and science camps for teachers and students that increase their basic knowledge of the sciences and make learning and teaching science relevant and fun. The programs at the RPSEC emphasize innovative, handson approaches to science that excite the educator about teaching science, which in turn encourage and promote student interest in science. Since the National Science Foundation first funded the RPSEC in 1987, over 3,000 teachers and 50,000 students have attended programs sponsored by the RPSEC.

Integrating instructional technology into school curricula is a great concern. A major problem is providing funds to purchase the necessary hardware and software to make the technology available to teachers. Another major problem occurs when the funds are finally available and the process of integration begins. Too often the teachers are not provided adequate training to integrate the instructional technology into their classes. In this case, instructional technology is available but is not integrated. The result is that many classrooms in many schools have computers available but these computers are not being used to their potential if at all.

The first step in beginning CTFEAMS was to identify teachers to participate in the program. Participating teachers were selected from schools within 25 miles of the RPSEC in order to make transportation to the computer classrooms at the RPSEC feasible. In all, seventeen teachers from nine schools were selected to participate in the program. Seven of the teachers were from middle schools and ten of the teachers were from elementary schools. There were six teachers of SOAR (gifted/talented) students and 11 teachers of "average" students.

The CTFEAMS program provided students and teachers with handson experiences in science and mathematics using computer technologies. The purpose was to train teachers how instructional technology can be integrated into their curriculum and to demonstrate how this can be done using the students of the teachers involved in the program. This training and demonstration was done through the two components of CTFEAMS.

The first component of the program consisted of a graduate course for the teachers. This course gave teachers training in the use of microcomputers in the classroom. The experiences from the course demonstrated how teachers could incorporate instructional technology into their classrooms and provided them with the information necessary to make educationally sound decisions when they have the opportunity to purchase technology oriented materials for their classrooms.

In the beginning of the course, the teachers evaluated computer assisted instructional programs in science and mathematics. Most of these programs were available in the Apple IIGS classroom at the RPSEC. The teachers also viewed several computer assisted instructional programs in the Apple Macintosh classroom at the RPSEC. Both of these classrooms were made available through a grant from Apple Computer.

In the next section of the course the teachers learned how to use the integrated word processor, spreadsheet, and data base program, Microsoft Works 2.0 for the Macintosh. The teachers learned how to generate a variety of instructional materials using the word processor. A key aspect of this was, "The computer word processor is not a glorified typewriter." The point of this quote was to help the teachers understand that the word processor could help them process the words and not just print them on the paper. The teachers created a computerized gradebook using the spreadsheet aspect of the integrated package. They learned how to write simple functions that would allow them to calculate student grades more easily, efficiently, and accurately. The teachers used the data base to develop records of their students that included assorted pieces of information that the teachers could use as reference material. Finally, the teachers developed a method to incorporate all three components of the integrated package to produce student progress reports that could be sent to the parents.

The final portion of the course was an introduction to hypermedia. The teachers learned some basic features of authoring programs using HyperCard 2.1 for the Macintosh. With this program the teachers created "stacks" to present content material to their students. The teachers also created stacks that controlled a laser videodisc player which displayed still and motion video segments on a separate TV monitor.

The second component of CTFEAMS involved the students of the teachers involved in the first component. After the teachers had received training and experience in the course, they accompanied their students to the RPSEC at USC-Aiken. While at the RPSEC, the students received handson experience with instructional technology. All of the activities in which the students were involved were covered during the graduate course that the teachers took prior to the student program. Therefore, the teachers were already familiar with the activities their students were doing and were able to provide the students with guidance and direction. The teachers related the activities at the RPSEC to the learning activities at their schools. Since the teachers were actively involved in this program they gained handson experience in integrating instructional technology into education.

There were 653 3rd through 8th grade students involved in the program. Approximately 50% of the students were gifted/talented while the other 50% were "average".

Each group of students made six, twohour visits to the RPSEC with their respective teachers. During the first visit, the students received a tour of the RPSEC. Part of the design of the RPSEC was to provide science and math learning activities for visitors. After the tour the students had some handson science and mathematics activities using the Apple IIGS computer classroom. When the students made the second visit to the RPSEC, they received an orientation to the Macintosh computer and learned how to use the word processor. During the third visit, the students used some computer assisted instructional programs and learned how to use the painting features that are available in many Macintosh computer paint programs.

During the fourth visit, the students received some introductory information in the use of HyperCard for the Macintosh and created a science stack that they completed during the fifth visit. During the last part of the fourth visit, the students were involved in some science activities using computers and laser videodisc players. During the sixth and final visit, the students printed their stacks and then were involved in more science activities that integrated the use of the computer and laser videodisc player.

The students were given a pre/post test on computer literacy. A comparison of the pretest and posttest scores indicated that there was a statistically significant increase in the students' computer literacy scores after the program was completed. This demonstrated that the program did have an effect on the computer literacy of the students involved.

A final aspect of the CTFEAMS program included the training of undergraduate students at the University of South Carolina at Aiken (USCA). Eight undergraduate students were involved in the program as teaching assistants. The USCA students were trained to assist in teaching and tutoring students throughout the program. Six of the eight students were education majors and CTFEAMS provided them with some experience in teaching young people before their student teaching programs began.

The response from all of the people involved in the program was positive. The USCA students gained valuable experience in interacting with 38th grade students. The 38th grade students responded with excitement to all of the activities during each of their visits. The teachers repeatedly reported that the students looked forward to the activities at the RPSEC and desired to have more time doing the activities.

The teachers gained valuable experience in incorporating instructional technology into science and mathematics education. The teachers in CTFEAMS still have a problem with incorporating the instructional technology into their classrooms, however. The underlying question and dilemma is not what to do with computers in their classrooms. These teachers are confident in that area. The problem now is how to get hardware and software into their classrooms so that they can begin using the instructional technology that they learned through this program.

The Development of a College/University Microcomputer Purchasing Program: A Conceptual Model

Lisa J. Shepherd
William J. Moressi
Don E. Gardner
School of Business Administration
Winthrop University
Rock Hill SC 29733
803-323-2281
moressiw@winthrop.edu

ABSTRACT

The authors of this paper provide a conceptual model of a computer purchasing program. This model should assist appropriate personnel in the establishment of a prototype that will best meet the needs of their campus community. The process of developing such a system includes defining such resources as equipment, personnel, facilities, and finance and marketing.

Some processes include surveying potential clients, setting software and hardware standards, selecting vendors, purchasing inventory, installing and distributing inventory, doing orientation, training, and user support and providing servicing and maintenance.

The model is a blueprint of activities that flow in both a parallel and serial time line. It shows the interdependence and interrelationship of the various elements. From this general schema, individuals may develop a prototype program.

INTRODUCTION

The purpose of this investigation is to present a conceptual model for a campus computer purchasing program that can be used by universities and colleges. Establishing a computer purchasing program is becoming increasingly important. Because of technological advances and changes in the education process, there is a greater need for all students and faculty to have access to this technology. Therefore, higher educational institutions must respond to the changing environment and the needs of the campus community. Establishing a computer purchasing program will contribute to extending the quality of academic achievement and provide an excellent recruitment tool for attracting students.

The scope of this project is broad. This report is designed as a generic blueprint of a computer purchasing program that can be used by most colleges and universities.

In this report, we will present a conceptual model of a computer purchasing program that can be adapted by most colleges and universities. Within the presentation, we address the following issues:

1. Financial (venture) considerations
2. Human resource requirements
3. Maintenance options
4. Hardware and software standards

PROCEDURES

We did a review that included institution interviews, first year student survey, literature searches. This presentation is focused primarily on information obtained from these resources.

We began by searching for colleges and universities that have established computer purchasing programs on their campuses. The initial search was conducted through the 1992 Directory of Computing Facilities in Higher Education by Charles H. Warlick. This Directory has information on purchasing programs which include total dollar sales by type of system. At first, we confined ourselves with institutions in North and South Carolina. We then had to expand our search to include institutions throughout the country. Once this list was completed, we proceeded by conducting telephone interviews with persons responsible for establishing and/or maintaining the institution's program. Some of our telephone interviews lead to on site interviews.

The questions we asked were designed to provide information regarding successes, as well as unexpected problems, with a computer purchasing program. We asked specific questions concerning the institutions program structure, maintenance for the systems, and financial and human resource policies. From the interviews, we were able to determine key pieces of information about computer purchasing programs.

To provide us with essential information concerning the student body, we developed a survey questionnaire for the first year class at Winthrop (see Appendix). Our main intent was to determine the student interest level in a computer purchasing program. Interviews were conducted with The Office of Assessment at Winthrop University to develop the questionnaire. The questionnaire was sent out just prior to the Fall semester of 1992 to 787 first year students enrolled at Winthrop University.

The responses to the questionnaire were entered into a database. From this database, we were able to determine key pieces of information about the incoming first year class. Some of the key issues that we were concerned with were: where the student will live while in school, what the student will be majoring in, what is the student interest level in buying a computer through the program, how much they are willing to contribute as a down payment, how quickly will they are willing to pay off the balance, and what platform would they purchase.

After the questionnaire was mailed, we requested a national database search from the campus library. This search was made through Dialog and included databases from ERIC, Microcomputer Index, Computer Database, and Magazine Index. The search was based on key words and phrases. The key words were education, microcomputers, sales, computers, colleges and universities. Less than 10 articles relating to University microcomputer sales were found. Relevant articles were selected and abstracts obtained for potential use. Most of the articles reviewed the effects that university sales had on retail sales five years ago or more.

RESULTS

The results from the first year student survey furnish information about student ideas concerning a computer purchasing program. The survey shows that 72% of the first year students would definitely or probably participate in a computer purchasing program. DOS based systems would be purchased by 73% while Macintosh would be purchased by 27% of the first year students. Approximately 40% of the students surveyed indicated that they expect the price of a computer to be between $1000 - $1500. Thirty nine percent would be willing to pay of the balance of the loan in two years.

THE CONCEPTUAL MODEL (see Figures 1a &1b).

EVENT 1 *REVIEW/ADAPT MODEL*
Adapt the model to specific institution.

EVENT 2 *DETERMINE MARKET SEGMENT TO SURVEY*
Determine the segment of the campus community that will be surveyed. This survey will help in determining how much of the campus community will participate in this program.

EVENT 3 *DETERMINE SURVEY QUESTIONS*
The survey questions should be structured in such as way as to provide key information pertaining to your particular institution. It may be necessary to determine what the students will be majoring in and what platform they will purchase.

EVENT 4 *CONDUCT & ANALYZE MARKET SURVEY*
Carry out the survey of the determined campus market.
Analyze the market survey to help estimate sales.

EVENT 5 *DECIDE SCOPE OF OPERATIONS*
From the analysis of the market survey, determine the appropriate size and scope of the computer purchasing program.

EVENT 6A *DEVELOP DIRECTOR'S JOB SPECS & SKILLS INVENTORY*
Define the responsibilities of the sales director.

EVENT 6B *DEVELOP ADVERTISING PLAN*
Determine the most effective way to inform the campus community of the computer purchasing program.

EVENT 6C *DETERMINE PART-TIME STAFFING REQUIREMENTS*
Using the results of the market survey, determine additional staffing requirements. This additional personnel could be made up of student employees, work-studies or interns.

EVENT 6D *REVIEW FIXTURES, EQUIPMENT, SPACE AND STORAGE NEEDS*

EVENT 7A *PREPARE PERSONNEL DESCRIPTION*
Define the specific qualifications that the sales director must have.

EVENT 7B *OBTAIN QUOTES FOR FIXTURES*
After estimating the operations space, calculate the fixtures that will be needed. Secure vendors quotes for these fixtures.

EVENT 8A *DETERMINE WAGE SCALE*
Define the salary range for the sales director.

EVENT 8B *DEVELOP FACILITIES PLAN*
This may come in the form of office areas, a store or both. Estimate space for the workers to talk with customers. There should also be office space for the coordinator. It will also be necessary to determine the physical space needed to receive and temporarily store computer orders.

EVENT 8C *DEVELOP FIXED ASSETS PLAN*
Determine the fixed assets that will be needed. This will include office furniture and accessories and the cost for each.

EVENT 9 *ESTIMATE EXPENSES & INCOME*
From the information gained in the previous steps, determine how much money will be needed to cover expenses and how much money can be expected in income.

EVENT 10 *PREPARE BUSINESS PLAN*
The Business Plan will evaluate the projected business operations of the program. The Business Plan will cover a broad area and will be used for the institution approval process.

EVENT 11 *INSTITUTION APPROVAL*
Receive approval from the University/College to begin a computer purchasing program.

EVENT 12A *COMMITTEE ESTABLISH & RECOMMENDS GENERIC APPLICATION STANDARD*
It is beneficial for the various academic departments to be involved with recommending software application standards. Each academic area will have individual ideas to contribute that will be helpful in selecting software applications. This committee will need to address standards for word processing, spread sheets, data bases, communications, presentations and graphics.

EVENT 12B *OBTAIN VENTURE CAPITAL*
Determine the source that will provide the necessary start up money for the program. This step will most likely require the involvement of the institutions financial department. There may be possible involvement of the institutions Foundation as well as outside sources (corporate and banking).

EVENT 13A *SELECT APPLICATION SOFTWARE*
Committee recommendations will assist the director in effectively choosing software standards for the program.

EVENT 13B *SELECT OPERATING SYSTEMS*
The operating systems that should be offered for sale will vary among institutions and may include offering a variety of platforms.

EVENT 13C *DEVELOP DIRECTOR ORIENTATION PROCEDURES*
Design orientation procedures for the incoming sales director. The director should be informed of the institutions purpose of the program as well as the model used in developing the program.

EVENT 13D *START RECRUITMENT PROGRAM*
Begin institution proceeding for recruiting the sales director.

EVENT 13E *ESTABLISH PROCEDURES TO RENOVATE SPACE*
This will include any remodeling, painting as well as special wiring for demos.

EVENT 13F *ORDER FIXTURES & EQUIPMENT*

EVENT 14A *SELECT HARDWARE OPTIONS*
Specific options (models, parts, etc) must be selected to be offered for sale in the program. This step should include selecting several different models which should give the customers an acceptable range to choose from. The "acceptable range" should depend on the specific institution. Once this has been determined, limit the purchases to the acceptable range.

EVENT 14B *HIRE FULL TIME DIRECTOR*
In the majority of cases, the institution will require a full time director/ coordinator. If the institution is small or expects sales to be low, they may want to consider hiring a part time director/coordinator instead.

EVENT 14C *DO INTERIOR FINISHING*
Using the interior finishing plan, perform the necessary renovations on the operations space.

EVENT 15A *ESTABLISH CLIENT CONTRACT TERMS*
If the institution is planning to help the student finance his or her computer purchase, contract terms must be established. Included in this step is how much of a down payment must be made by the customer and how soon they must pay the balance due.

EVENT 15B *ESTABLISH SUPPORT AREAS OF PROGRAM*
How the program will impact other administrative areas of the University? As the contract terms are established, impact areas will be discovered.

EVENT 15C *ESTABLISH PRODUCT PRICING*
Determine a price range for the systems. The range should be able to include inexpensive systems and reasonably priced, high performance systems.

EVENT 15D *DIRECTOR ORIENTATION*
Carry out the orientation procedures developed for the sales director.

EVENT 16A *CREATE CLIENT CONTRACT, CREDIT PLANS, ETC.*
Create the individual contracts and credit plans, and official check-in sheet to be used when the orders arrive, etc.

EVENT 16B *SOLICIT BIDS FROM VENDORS*
Solicit bids for micro computer vendors along with bids from software vendors.

EVENT 16C *DEVELOP PROGRAM TO TRAIN PART TIME EMPLOYEES*

EVENT 16D *INSTALL FIXTURES AND EQUIPMENT*

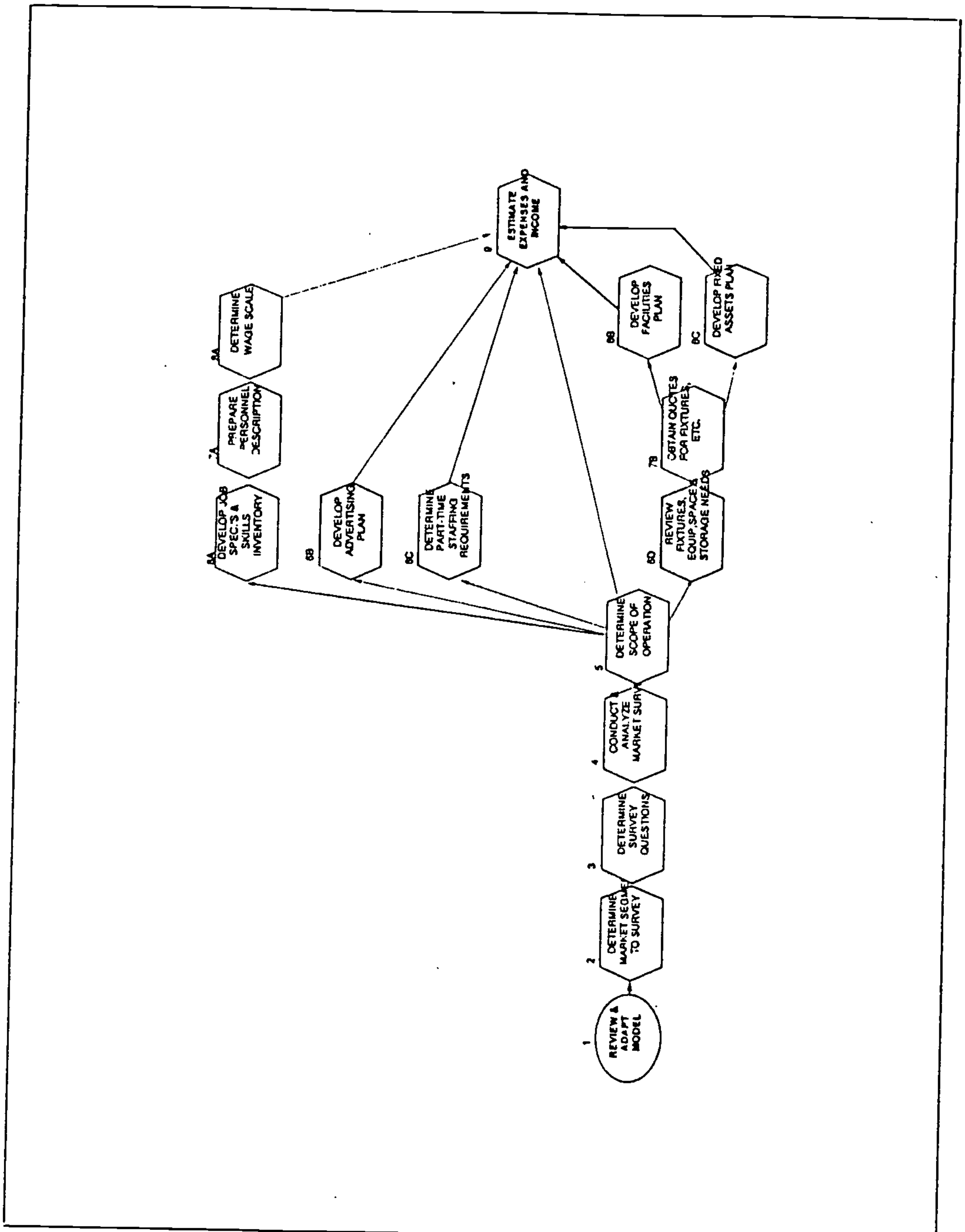


Figure 1a: PERT Model

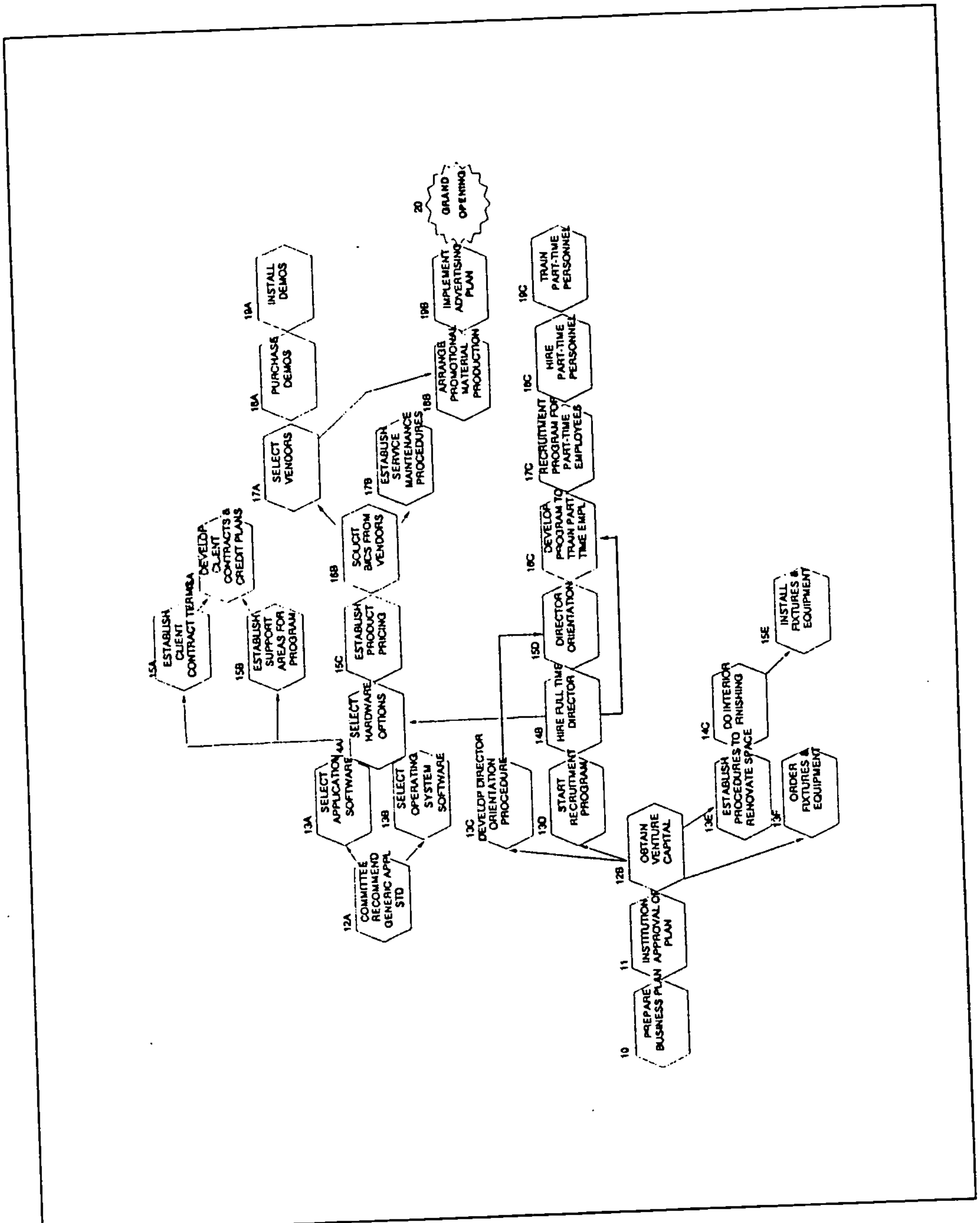


Figure 1b: PERT Model

The following table is a list of time (weeks) schedules for the PERT events. The letter a is optimistic time, b is pessimistic time, and m is most likely time. The value for t (estimated time) is calculated from the formula $T=(a+4m+b)/6$. All values are expressed in weeks. L.E. is the longest event time for a 'set' of parallel Events.

#	EVENT	a	b	m	T	L.E. TIME
1	REVIEW/ADAPT MODEL	2.0	6.0	3.0	3.3	3.3
2	DETERMINE MARKET SEGMENT TO SURVEY	1.0	2.0	1.0	1.2	1.2
3	DETERMINE SURVEY QUESTIONS	1.0	2.0	1.5	1.5	1.5
4	CONDUCT & ANALYZE MARKET SURVEY	4.0	8.0	5.0	5.3	5.3
5	DECIDE SCOPE OF OPERATIONS	2.0	4.0	3.0	3.0	3.0
6A	DEVELOP DIRECTOR'S JOB SPECS & SKILLS INVENTORY	2.0	3.0	2.0	2.2	
6B	DEVELOP ADVERTISING PLAN	2.0	3.0	2.5	2.5	2.5
6C	DETERMINE PART-TIME STAFFING REQUIREMENTS	1.0	2.0	2.0	1.8	
6D	REVIEW FIXTURES, EQUIP, SPACE & STORAGE NEEDS	1.0	2.0	2.0	1.8	
7A	PREPARE PERSONNEL DESCRIPTION	2.0	3.0	2.0	2.2	
7B	OBTAIN QUOTES FOR FIXTURES	4.0	8.0	7.0	6.7	6.7
8A	DETERMINE WAGE SCALE	1.0	2.0	1.5	1.5	
8B	DEVELOP FACILITIES PLAN	2.0	5.0	3.0	3.2	3.2
8C	DEVELOP FIXED ASSETS PLAN	1.0	3.0	2.0	2.0	
9	ESTIMATE EXPENSES & INCOME	2.0	4.0	3.5	3.3	3.3
10	PREPARE BUSINESS PLAN	2.0	4.0	3.5	3.3	3.3
11	INSTITUTIONAL APPROVAL OF PLAN	6.0	8.0	7.0	7.0	7.0

#	EVENT	a	b	m	T	L.E. TIME
12A	COMMITTEE ESTABLISH & RECOMMENDS GENERIC APPLICATION STANDARDS	2.0	4.0	3.5	3.3	
12B	OBTAIN VENTURE CAPITAL	8.0	12.0	12.0	11.3	11.3
13A	SELECT APPLICATIONS SOFTWARE	1.0	3.0	2.0	2.0	
13B	SELECT OPERATING SYSTEMS	1.0	3.0	2.0	2.0	
13C	DEVELOP DIRECTOR ORIENTATION PROCEDURES	1.0	3.0	2.0	2.0	
13D	START RECRUITMENT PROGRAM	4.0	6.0	5.5	5.3	5.3
13E	ESTABLISH PROCEDURES TO RENOVATE SPACE	2.0	4.0	3.0	3.0	
13F	ORDER FIXTURES & EQUIPMENT	1.0	2.0	1.0	1.2	
14A	SELECT HARDWARE OPTIONS	1.0	2.0	1.0	1.2	
14B	HIRE FULL TIME DIRECTOR	1.0	2.0	1.0	1.2	
14C	DO INTERIOR FINISHING	2.0	7.0	5.0	4.8	4.8
15A	ESTABLISH CLIENT CONTRACT TERMS	2.0	4.0	3.5	3.3	3.3
15B	ESTABLISH SUPPORT AREAS FOR PROGRAM	2.0	4.0	3.0	3.0	
15C	ESTABLISH PRODUCT PRICING	1.0	2.0	2.0	1.8	
15D	DIRECTOR ORIENTATION	1.0	1.0	1.0	1.0	
16A	CREATE CLIENT CONTRACT AND CREDIT PLAN	1.0	2.0	2.0	1.8	
16B	SOLICIT BIDS FROM VENDORS	4.0	8.0	7.0	6.7	6.7
16C	DEVELOP PROGRAM TO TRAIN PART-TIME EMPLOYEES	1.0	3.0	1.5	1.7	
16D	INSTALL FIXTURES AND EQUIPMENT	1.0	3.0	2.0	2.0	
17A	SELECT VENDORS	1.0	2.0	1.0	1.2	
17B	ESTABLISH SERVICE MAINTENANCE PROCEDURES	1.0	4.0	3.0	2.8	
17C	START RECRUITMENT PROGRAM FOR PART-TIME EMPLOYEES	2.0	4.0	3.0	3.0	3.0
18A	PURCHASE & RECEIPT OF DEMOS	3.0	6.0	4.0	4.2	
18B	PROMOTIONAL MATERIAL ARRANGEMENT & PRODUCTION	3.0	6.0	5.5	5.2	5.2
18C	HIRE PART-TIME PERSONNEL	2.0	3.0	2.0	2.2	

#	EVENT	a	b	m	T	L.E. TIME
19a	INSTALL DEMOS	1.0	3.0	1.5	1.7	1.7
19b	TRAIN PART-TIME PERSONNEL	1.0	3.0	1.0	1.3	
20	IMPLEMENT ADVERTISING PLAN	1.0	1.0	1.0	1.0	1.0
21	GRAND OPENING	0.5	1.0	0.5	0.6	0.6
	TOTAL WEEKS FOR LONGEST EVENTS OF EACH CATEGORY					83.2

DISCUSSION

Some general conclusions from our work are:

1. It is possible that only a portion of the money needed to start a computer purchasing program will be available. Therefore, the institution must consider the possibility of initiating a program on a limited basis.

2. It is consistently underestimated how much effort and time must be dedicated to start and maintain a program of this type. Therefore, most institutions will require a full time director. There is a unique case where students are desired for an essential position, and that is customer education and training.

3. Although there may be room for on-site maintenance in the future, it is important to outsource the maintenance in the early stages of the program.

4. Before the program is initiated, the institution must select hardware and software standards that will be widely used in many different academic areas. The institution must determine standards for operation systems and applications software such as word processing, spread sheet, data base, communication, and graphics.

We will elaborate on these and other points in more detail at our presentation. We hope for suggestions, comments, and recommendations.

APPENDIX

FIRST YEAR STUDENT SURVEY

Demographic Information

1. Your sex : Male Female

2. Your age:____

3. Your classification:
   First year Senior
   Sophomore Graduate student
   Junior Other

4. Which of the following best describes your roommates while at college?
   Parents Spouse or Children Other
   Peers I live alone

5. Which best describes your place of residence?
   House off campus A fraternity house Other
   Apartment off campus A sorority house
   Winthrop residence hall On Campus Apartment

Computing Information

1. Do you currently own a personal computer? yes no

2. Do you plan to bring a personal computer with you to Winthrop?
   yes no undecided

3. Do you plan to purchase a personal computer while you are a student at Winthrop?
   yes no undecided

4. If you wanted to purchase a personal computer, how much would you expect to pay?
less than $1000
$1000 - $1500
$1500 - $2000
$2000 - $2500
$2500 - $3000
More than $3000

5. If a plan were to be made available through Winthrop whereby students could purchase computers at educational discounts and then pay for them in installments that were included in each semester's tuition and fees payment, would you be likely to purchase a computer?
Almost definitely yes
Probably yes
Probably not (go to item 10 below)
Almost definitely not (go to item 10 below)

6. If you wanted to purchase a personal computer, how much money would you be willing to contribute as a down payment?
Less than $250
$250 - $500
$500 - $750
$750 - $1000
More than $1000
I would pay the total price at the time of purchase.

7. Would you be willing to pay the balance owed in:
1 year
2 years
3 years
4 years
I would pay the total price at the time of purchase.

8. How much would you be willing to pay each semester as a payment on a computer?
less than $100
$100 - $200
$200 - $300
$300 - $400
more than $400

9. If you were to purchase a personal computer today, which type would you likely purchase?
IBM or compatible (DOS based machine)
Apple/Macintosh
Workstation (Sun System, Next, etc.)
Other Specify:_____

10. What are do you plan to major in at Winthrop?

Business	Arts and Sciences
Computer Science	Undecided
Visual and Performing Arts	Other. Specify:_____

Comparison of Database Systems

Thomas J. Smith
Department of Mathematics and Computer Science
Manhattan College
Riverdale NY 10471
212-920-0427
tsmith@manvax.cc.mancol.edu

ABSTRACT

This paper describes the projects assigned in a course on database systems being offered for the first time at Manhattan College. The author attempts to blend into the course the usual material on database theory and some practical experience using important technologies. The latter experience is gained by working on two projects; the first is a team project using the Query By Example technology featured in the Paradox software package and the second is an individual project using Structured Query Language (SQL).

INTRODUCTION

During the spring, 1993 semester the author offered a course called Database Systems, a new elective course designed for upper level computer science majors and minors. The objectives in offering the course were to provide the students with both the traditional theoretical material on database systems and the opportunity to gain some practical experience with several widely available and important database implementations. The theoretical material, heavily emphasizing relational database theory, was presented from the text chosen for the course, C.J. Date's *An Introduction to Database Systems*, Vol. I, fifth edition. The practical experience came from working on two projects. The first was a team project in which each team attempted to provide a small business with a computerized database system. The second was an individual project in which students practiced creating and accessing a database using SQL, the industry standard relational database language. In this paper we will concentrate on the projects.

TEAM PROJECTS

The class of 26 students was divided into six teams (each team having four or five members), with the students self selecting the teams. Each team was responsible for finding a small business which was not currently computerized. The object of the game was to design and implement a computerized database system for the business. The team members selected their own team leader who was responsible for providing the instructor with regular updates on the progress of the project.

One constraint in the project was that it had to be implemented using Paradox, a popular and powerful microcomputer based database management package put out by Borland, International. There were several reasons for choosing Paradox. First, the package is available on the microcomputer network at the college (Manhattan College has a site license with Borland which allows us to make available on the network all of the Borland products, such as Turbo Pascal, dBASE, and Quattro). This availability meant that students had free access to the package and to the computers on which it would run. A second reason was that Paradox uses QBE, the Query By Example technology which appears to be growing in popularity. The emphasis in QBE is on the use of windows-like forms to allow users with little or no computer experience to access information without learning a programming language. Each team was advised to purchase one of the many paperback books, such as those by Cobb or Simpson (see References), to use as a reference for the project.

The businesses chosen by the students were a varied collection. One student's mother owns a floral business; another's father owns a pharmacy. Several students work at part-time jobs, one for a company which rents and installs tents, a second for a company that rents and sell supplies for parties, a third for a liquor store. Another team is attempting to computerize the files for a local YMCA's swimming program for children; one of the team members is an instructor in the program. This diversity of businesses makes for a nice cross section of problems to discuss. In particular, a business which does rentals exclusively presents several different problems from a business which does sales only. Several of the teams have received encouragement from the owners who have expressed interest in using the final product.

Early in the course, classroom lectures concentrated on the questions associated with relational database design and the normalization of tables. This material was presented at the outset so that the teams could incorporate as much of it as practical into the design of the tables to be used in the project. With the exception of a brief set of notes and one class session, spent in a microcomputer lab, in which the major features of Paradox were demonstrated, no class time was devoted to this software. Students had to learn the package on their own.

A major part of the requirement of the project is that the finished product must be user friendly, which translates into menu driven. Towards that end, students must learn to program in PAL, Paradox's programming language. The work amongst team members was divided so that some were creating and entering data into the tables, while the others concentrated on the forms and scripts (programs) needed to run the system.

At this writing the teams have been reasonably successful in analyzing the businesses so as to design database tables which reflect the actual operation. In several cases, the data tables serve to mirror a subset of the real data. The vast number of items for sale in the pharmacy, for example, makes it impractical to handle all of the inventory. The team has selected a reasonable subset to serve as a model. Mastering the programming language for the project is posing more of a problem, although students' familiarity with multiple languages from earlier courses (Pascal, C, FORTRAN) does make it somewhat easier.

The projects are to be finished by the last week of the semester. A written report, including all of the tables and code generated, must be submitted. In addition, each team will give a live demonstration to the class of the finished software system.

INDIVIDUAL PROJECTS

Paradox, the product chosen for the team project, does not use SQL, the industry standard language for database systems. Since SQL is the core of so many database products, particularly at the mainframe and minicomputer level. it was decided to incorporate some SQL into the course. Approximately three weeks were devoted to the various components of SQL, with most of the time devoted to DML, the Data Manipulation Language component.

The original intent was to include a second, short, project using SQL in conjunction with Oracle, a database management system which is widely used in the business world. Unfortunately, the new microcomputer version of Oracle has heavy hardware demands (at least six megabytes of RAM and approximately 40 megabytes of hard disk space) and the network version is expensive. These obstacles forced a change in plan. A reduced version of the original assignment was developed for the version of SQL which is available as part of dBASE IV. dBASE was originally developed by Ashton Tate but has recently been purchased by Borland, so dBASE IV is also available on the microcomputer network.

Several relational database tables which were used as examples during the class lectures were placed on the network file server. These tables were created by the author using dBASE IV to serve as the base tables for the SQL project. Students were given access to the tables so that the data in them could be loaded into SQL tables which they must create. (One of the difficulties with SQL is that it does not provide a mechanism for the easy insertion of multiple rows from within SQL. It is far easier to create the tables from outside SQL and then load the data into SQL tables.) To encourage individual effort, as opposed to team effort, each student must enter a small amount of data from within SQL which is unique to that student. Once the students have the SQL tables, they are asked to solve a series of problems which entail accessing the data using the SQL commands from DML. The results of these queries are saved to temporary tables which will be printed out from dBASE (i.e. outside SQL). This project will replace one of the in-class examinations.

Various difficulties with this project have cropped up since the network version of dBASE requires substantial fine tuning in order to get the SQL component to run smoothly. As an example, the dBASE default is to have all SQL databases stored as subdirectories of the directory which houses the dBASE system files. On the network, of course, the system files are kept on the file server which is not available to the students for writing files. Other problems have arisen because the dBASE version of SQL does not support several of the language's standard features.

SUMMARY

Assuming that all goes as planned, the students in the new course will have gained valuable experience dealing with real life issues confronted by software consultants. The team project forces them to address the problems encountered in setting up a computerized database system for an actual business. They must analyze and understand the operation of the business prior to attempting to model it. In this way they see a practical application of the theoretical material learned in class.

The second project is designed to provide at least a small amount of practice dealing with SQL, the industry standard database management language. The students know the importance of becoming familiar with SQL since there are numerous job advertisements for people with SQL experience.

REFERENCES

1. C.J. Date, An Introduction to Database Systems, Vol. 1, 5th ed., Addison Wesley, 1990.
2. Douglas Cobb, Paradox 3 Handbook, 2nd ed., Bantam Books, 1989.
3. Alan Simpson, Mastering Paradox 4 for DOS, Sybex, 1992.

A Computer-Based First Course In Differential Equations

Thomas Smith
Charles Stolze
Kathryn Weld
Department of Mathematics and Computer Science
Manhattan College
Riverdale NY 10471
212-920-0427
tsmith@manvax.cc.mancol.edu

INTRODUCTION

In recent years, instructors who teach calculus have had success in the use of a computer algebra system (CAS), such as Maple or Mathematica, as a pedagogical tool. The CAS has been employed to do graphics, numerical computation, and symbolic computation, enabling the instructor to focus on and develop concepts. This approach has spawned new ways to teach calculus -- through the use of laboratories.

The authors have done research on the use of a CAS to teach differential equations and it appears that there is great potential for making substantial improvements in the curriculum. With a CAS, the traditional first course which focuses on methods for solving differential equations can be expanded to include the use of numerical methods, graphing techniques, and mathematical modeling. Moreover, the use of a CAS allows the students to take a more active role in the learning process.

TEACHING CALCULUS WITH MAPLE

At Manhattan College, four faculty members are currently involved in giving regular assignments in calculus which involve the use of the CAS, Maple. Two of these classes meet in the laboratory 2 hours per week. We intend that eventually all sections of calculus will be supplemented by Maple in various ways, depending on the teaching style of the instructor. In order to accomplish such widespread reform we will offer a faculty workshop this summer for all mathematics faculty at the College. The workshop will focus on the use of Maple to teach calculus; it will be funded by a Grant from the GE Foundation.

Results from our work in teaching computer supported calculus demonstrate that not only does the list of topics change quite naturally when the course material is augmented by the use of a CAS, but it becomes natural to teach in a more open ended way, asking questions which require an interpretation of the mathematical material, analysis of information gleaned from graphs produced by the CAS, and so on.

The natural extension of our work in the calculus sequence is to introduce the use of a CAS into the differential equations course which most science students take as a sequel to Calculus III. This is desirable both because it introduces an experimental quality to the learning experience and also because the traditional course in differential equations, as taught at most undergraduate institutions, is seriously out of date.

A FIRST COURSE IN ORDINARY DIFFERENTIAL EQUATIONS: DRAWBACKS OF THE TRADITIONAL APPROACH.

Students who major in Mathematics, Science, or Engineering usually take a one-semester course in Ordinary Differential Equations during their sophomore year. This is primarily due to the fact that many problems in the applied sciences are best modeled using differential equations. In many institutions the syllabus and methods for teaching this course have not changed in thirty years. The course generally concentrates on the topics:

First Order Equations with Applications
Second Order Equations with Applications
Series Solutions
Numerical Methods
LaPlace Transforms
Systems of Differential Equations

It is common to leave the latter three topics for the end of the course as time permits.

While educators recommend that students who major in the mathematical sciences should take a course in differential equations, they also agree that the content of the course needs revision and that instructional methodology should change. A large portion of the course deals with a multitude of techniques for solving certain kinds of differential equations. Hubbard and West express the problem in a colorful way in their book Differential Equations, A Dynamical Systems Approach, published by Springer-Verlag: differential equations can be divided into two groups, the friendly ones, which can, with a lot of work, be successfully analyzed, and the strangers. "The strangers are quite a different problem. They are strange and mysterious. There is no reliable technique for dealing with them." Unfortunately, most differential equations fall into the second category, yet in the traditional course most of the allotted time is devoted to methods for solving the first,"friendly" type of differential equation, because those methods are easier to teach, despite being less useful in practice.

This does not mean that there are no techniques for dealing with the second class of differential equations. These techniques fall into two categories: numerical methods, and qualitative methods. Qualitative methods are time consuming without a computer, since they rely on assembling a vast amount of data about the graphical representation of a differential equation. Numerical methods are also computationally expensive when done by hand. Both of these methods, however, have become accessible with the advent of the computer algebra system. Better yet, a CAS can be used not only to generate a table of data associated with a numerical solution; it can also be used to graph the solution, rendering it much more understandable to the student. Hubbard and West argue that to continue to teach differential equations in the old style is like looking for your lost quarter under the street lamp, even though you dropped it one hundred yards away, because that is where the light is.

Our experience is that students in the traditional course are engaged in computation for computation sake and get minimal experience with applied problems or the dynamic flavor of the subject. Both numerical and qualitative methods, which demand extensive computation, are needed to tackle a typical differential equation which models a real world situation. The demands of extensive computation have made it difficult to teach these topics in a traditional course. As a result, these methods are usually studied at the end of the course and consequently most students perceive this material as isolated and not really "connected" to what has been studied earlier. In fact students do not really "believe in" the solutions they get with these methods, because they are so accustomed to the artificial problems found in most books. Thus the introduction to differential equations often leads to a misunderstanding of their use in common applications.

USE OF THE COMPUTER IN TEACHING DIFFERENTIAL EQUATIONS

Mathematicians and scientists nationwide have called for reform in the teaching of the calculus/differential equations sequence, and in particular for the integration of a computer algebra system. Use of technology requires an extensive rethinking of the way in which mathematics is taught. While a substantial amount of work on the calculus curriculum has been achieved, much of it supported by grants from NSF, much less has been accomplished in the subject of differential equations. For most students, differential equations is the "capstone" course in mathematics, the gateway to applications in their field, and the rationale for studying mathematics in the first place. Clearly this aim would be better realized if the computational obstacles to studies of some of the more practical applications of differential equations were eliminated.

Some work has been done. In addition to the book by Hubbard and West, manuals have been written which begin to address the need to involve the use of the computer in teaching differential equations. Most prominent among these are the Collegiate Mathematics Courseware Manual, by Alfors and West (with support from the PEW program) and Differential Equations Laboratory Workbook, by Borelli, Coleman and Boyce, published by John Wiley and Sons. The latter manual is designed for use with the program DEQSOLVE, which runs on the VAX/VMS system. The Ahlfors and West manual is designed for use with software written at Cornell for the MacIntosh computer.

Hubbard and West's approach effectively creates a new course. While tremendously appealing mathematically, we have taken a less radical approach at Manhattan College. Qualitative methods require mathematical maturity to interpret graphs, and numerical methods require an understanding of the approximation process, yet at Manhattan College, differential equations is offered as a sophomore course. Our approach has been to modify the traditional syllabus and design laboratory exercises which help students acquire the maturity, both in mathematics and in general problem solving, which will aid their ability to comprehend the new approach.

One goal of the new laboratory approach is to help to develop critical thinking skills which allow the student to engage in problems which are more open ended, and to encourage the student to be more active. Ideally, the student will begin to use the computer to experiment with finding solutions to open ended questions. With creative assignments, the use of a CAS would radically transform both the material taught and the method of teaching it, making the student a participant in mathematics, as opposed to the traditional spectator.

IMPLEMENTATION

Our first step was the restructuring of the syllabus, placing more emphasis on numerical techniques, techniques from linear algebra, investigation of applications and modeling real world phenomenon and less emphasis on techniques for finding closed form solutions. The CAS makes possible experimentation with graphical representations from the very beginning. This in turn leads naturally to the subject of systems of linear differential equations and their connection to linear algebra.

Integrating the use of the computer into the teaching of differential equations requires more than what is currently available in lab manuals. Currently, we know of only one differential equations manual, Differential Equations Laboratory Workbook by Borelli, Coleman and Boyce, published by John Wiley & Sons. Most of the labs in this manual, however, are not suitable for our students, since the authors presume knowledge of subjects which are not prerequisites for differential equations as taught at our college. However, it is potentially a useful reference for instructors who intend to create laboratory exercises, and we intend to adapt some of these labs where it is appropriate.

Ahlfors and West's Courseware Manual, devotes only a small chapter to differential equations, and uses the Analyzer program which runs only on the MacIntosh computer. The exercises for differential equations are fairly routine. Our plan has been to write projects which will engage students in in-depth exploration of specific topics over periods of about two weeks, and also to generate a series of shorter exercises, to be attempted in class in small groups.

In developing projects and laboratory exercises suitable for use in our classes we have found that much energy needs to go into the writing of these exercises, so that students are required to think about concepts and modeling issues by being asked to answer questions about their work at timely intervals. We have found that the method employed by Ithaca College in their NSF funded Calculus project, of requiring students to do projects in two steps, the first being an analysis of the problem which generates a strategy for solving the problem, is particularly useful for our students.

The following is a partial list of topics and related questions which we are in the process of developing into modules.

i) Qualitative Methods:

a. The use of a direction field command (available on a CAS) to generate the direction field associated with a differential equation. What types of functions might be solutions for such a direction field?

b. What insight is gained from examining the direction field together with the existence and uniqueness theorem for differential equations?

c. Which of the following curves could possibly be solutions to the differential equation associated with a given direction field? How do solutions behave as the independent variable increases? Which solutions are stable?

ii) Numerical Methods:

a. The use of numerical methods to generate approximate solutions to differential equations. The use of a CAS to generate and graph such solutions.

b. The use of a CAS to discuss error analysis in numerical solutions. The use of a CAS to model real data, such as data from the AIDS epidemic, to make predictions based on the model, and to analyze the effectiveness of the model.

iii) Qualitative Methods and Linear Algebra

a. The use of a CAS to compute eigenvalues and eigenvectors for a linear homogeneous system of differential equations. The use of the graphics capabilities of a CAS to study the qualitative behavior of solutions to systems of differential equations.

b. Use of the pendulum problem as a model for studying the stability of a system of equations.

c. Use of the predator-prey model to study the technique of approximation of a system by a linear system in order to analyze stability.

Some of these exercises have already been tested in the pilot course we are teaching during the Spring, 1993 semester. At the ASCUE Conference, we intend to report both on what worked and what didn't.

REFERENCES

1. Hubbard, J.H., B.H. West, Differential Equations, Springer-Verlag, 1991.

2. Borreli, Robert L., Courtney Coleman, William E. Boyce, Differential Equations Laboratory Workbook, John Wiley & Sons, Inc., 1992.

3. Ahlfors, West, Collegiate Mathematics Courseware Manual, 1991.

Development, Implementation and Incorporation of an Interdisciplinary Technology Classroom for Science and Math

Paul S. Tabor
Biology Department
Clarke College
1550 Clarke Drive
Dubuque IA 52001
319-588-6526

Faculty from the Departments of Biology, Computer Sciences and Mathematics have recognized the need to enhance students' investigative experiences and involvement in course content and in the learning process. Because there is extensive interdisciplinary interaction among faculty from these departments, we identified common requirements for curriculum enhancement including the integration of computer projects and computer explorations into daily class sessions and the availability of a closed, computer classroom environment for group investigations. Faculty focussed initially on the freshman-level courses because, like the national trend, students with strong interests in science as freshmen lost that interest in a major in the sciences by the end of their sophomore year. We realize that as the freshmen become familiar with the capabilities of the computer classroom it will make it easier to expand the incorporation of this technology into the upper division math and sciences curricula.

Clarke College was awarded an NSF Instrumentation and Laboratory Improvement grant, "An Effective Learning Environment using Computer Technology (Project ELECT)" to provide support for the facility. The ELECT facility has three components, a computer-based, multimedia classroom with 10 student workstations, a control multimedia workstation, an RGB/video projector, laserdisc player, color scanner and printers; a faculty authoring workstation identical to the control workstation, with PC-VCR and monitor; and a mobile computer workstation identical to the control workstation with an LCD projection pad and projector, VCR and large monitor/receiver. The faculty authoring workstation and ELECT classroom student workstations and control workstation are networked with the network server. In the ELECT classroom, the instructor can route information to any or all student workstations and students' work can be monitored at the control workstation and any student's screen can be projected to the whole class. The control workstation can control the multimedia capabilities (CD-ROM and laserdisc player) of the classroom as well. In the biology laboratory, the mobile workstation is used to make video recordings of observations made by students during lab experiments and digitally capture these images to read/write optical disks to be manipulated later as multimedia files in the ELECT classroom.

Distinct changes have resulted from the incorporation of the use of the ELECT facilities in the curriculum. A new course, *Introduction to Computers as Tools for Math and Science*, has been introduced to familiarize students with the information accessing and transferring capabilities of the networked, multimedia classroom and the application software tools for analyses and presentation on the system. Courses in Mathematics and introductory courses in Computer Science have added a weekly closed laboratory session in the ELECT classroom.

A second lab session meeting in the ELECT classroom has been added to the introductory biology lab courses.

Faculty in Mathematics have incorporated the use of networked *Mathematica* notebooks into courses with the following objectives: to allow an exploratory learning of ideas and patterns before they are actually presented in class, to supplement and extend the concepts presented in class, and to provide tutorial guides for students as they work through exercises. In designing the lab exercises appropriate to the curriculum, the faculty have concentrated on approaches for students to develop quickly a familiarity with *Mathematica*, placing emphasis on the graphing capabilities. In math, the faculty have found that students need explicit examples or models in order to take full advantage of the full capabilities of *Mathematica*. The most successful exercises were those that examined problems related to the central concepts of the course. With a computer classroom and software as powerful as *Mathematica*, there is a need to balance explicitly instructing students and allowing them to freely explore ideas.

Faculty in Computer Sciences have incorporated closed lab sessions which closely integrate defined programming tasks done in the ELECT classroom with the material presented in lecture. The tasks in the introductory course include interpreting, modifying and debugging codes and writing segments of additional code. The student lab reports submitted in a "handins" subdirectory included not only program codes but answers to questions about each step of the task. The tasks were designed so that students, working independently or in collaboration, become active participants in the learning process. The interactive nature of the networked control station and student workstations combined with the presentation capabilities in the computer classroom provides the opportunity for immediate feedback on a student's work from faculty or classmates. Computer Sciences faculty found that generally students who actively participated in the closed lab sessions were able to successfully complete the lab assignments. However, there was a greater diversity of scores on the lecture exams and programming assignments, with some students consistently performing more poorly than on laboratory tasks.

The ELECT classroom session has been incorporated in the *Principles of Biology Laboratory* course by the addition of a second lab period the day following the lab experiments. The students bring raw data and their preliminary notes of observations and, using spreadsheet and word processing applications, create data files. Student working files are saved on their own floppy disks and not on the file server. Examples of data analyses from related studies, resource information and thought questions are provided to give students new ideas on how they might apply their own results beyond the lab exercises. Students find this information on the network server in read and copy-only files in a subdirectory called "handouts". By choice, students usually do all data analysis and create graphs and tables for the lab report before doing any writing. Although students work in pairs on the lab experiments, each student is responsible for a lab report. The lab reports created from working files on floppy disks are submitted to a subdirectory called "handins". Students have no access to the files in this subdirectory. This procedure is analogous to slipping a lab report under the closed door of an instructor. I evaluate and comment on the lab reports and return the files to the "handouts" directory

When students are working independently on their lab reports in the ELECT classroom, they spend an average of two hours a week. This is in addition to the one-hour scheduled session per week. This represents four times the amount of time spent in previous years on preparing the conventional lab reports. Performance on the lab reports was exceptional: all students produced lab reports that had excellent introductions, results, presentations of results and discussions. Final lab exam scores for this year's course that incorporated the ELECT classroom session were significantly higher than the exam scores for the two previous years as shown in Table 1 below.

Table 1. Principles of Biology I Final Exam Score Averages[1]

Year	Mean Class Average	Median Class Average
1990	75.6	82
1991	72.0	74
1992	89.6	92

[1] classes of science majors range from 10 to 14.

These exams used short-answer questions and showed a greater breadth of understanding of concepts (e.g., enzyme structure and function and factors affecting enzyme activity) with the incorporation of the computer classroom session.

Students in *Principles of Biology I Lab* are required to take the *Introduction to Computers as Tools for Math and Science* course to become familiar with the information accessing and transferring capabilities of the networked, multimedia classroom and the application software tools for analyses and presentation on the ELECT classroom networked system. Notwithstanding the experience the students gained in that course, in the biology ELECT classroom session, the freshman-level students became highly proficient in data recording, graphical analyses of the results, statistical analysis and creation of graphics for their lab reports. The instant gratification of observing graphic presentations of their results was an important factor in the further investigations of their results that the students made. With the ELECT classroom capabilities, students had the opportunity to work with their own constructions of the lab results. The success that the students had with the concepts of the lab exercises is attributable, in part, to this process of investigative interpretations. The ELECT classroom provided an environment for group analysis which proved to be synergistic; novel conceptualizations and unique interpretations were developed by groups sharing information and working together that were not generated by the students individually. In the second semester biology lab course, students are responsible for choosing the problems that they want to address and designing their own experiments. They are actually writing their own lab manual. As a result, the students are now using the informational and multimedia resources of the computer network to a greater extent.

The Faculty Innovation Center
Ideas, Training, & Development Workshop

Robert Teague
Director, Academic Computing
Terrance Keys
Coordinator, Professional Development
Academic Computing
Monroe Community College
1000 E. Henrietta Rd., Rochester, NY 14623-5780
716-292-2174
rteague@eckert.acadcomp.monroecc.edu

ABSTRACT

How to spark the faculty's technological creativity in the curriculum through using the Faculty Innovation Center. This umbrella facility takes the faculty from idea generation to the final classroom product. Students benefit from colorful presentations, computerized reviews, well illustrated class notes, and energized instruction. The focus of this workshop will be to discuss the creation and usage of a faculty innovation center to inspire faculty to use computers in their classrooms. Topics will include how to secure faculty usage of the center by faculty incentives; inspiring ideas through professional development days; information sharing workshops to introduce technologies; training classes, consulting and partnerships to empower faculty to reach their goals, and adequate facilities to develop curriculum materials. The goal is to describe an easily customized plan for establishing and operating a faculty innovation center.

I. Introduction
   A. Monroe Community College is located in Brighton, NY, a suburb of Rochester, with a city center in downtown Rochester
   B. Enrollment is 14,100; 10250 FTE
   C. There are about 400 fulltime and 450 adjunct faculty members
   D. Origin of computer centers
      1. Electronic Learning Centers -- 1986
      2. Faculty Innovation Center -- Spring 1991
      3. Electronic Classrooms -- Fall 1991

II. Problem
   A. Lack of development resources, especially for emerging technologies (multimedia)
      1. Hardware
      2. Software
      3. Peripherals (CD, scanner)

   B. Faculty Usage
      1. Incentives
      2. Introduction to new technologies
      3. Tips & techniques
      4. Development Support Staff

C. Classrooms / Learning Centers
1. Need electronic classrooms and learning centers to use developed curricula
2. Normally none or limited number
3. Do they have same hardware/software

D. Funding
1. Can't provide all faculty with computers
2. Departments can't afford color laser printers, etc.
3. Software updates are frequent and expensive
4. Many demands of professional development funds

III. Solution Faculty Innovation Center

A. Resources
1. Hardware
   -- Top end computer platforms (486 DX, Macintosh Quadra)
   -- Target platforms (Centris 650) not always top end but at least capable of performing required tasks
2. Software
   -- Multimedia (Action, Director)
   -- Presentations (Persuasion)
   -- Software design (Hypercard, Toolbook)
   -- Desktop Publishing (PageMaker)
   -- Drawing (CorelDraw, Freehand)
   -- Utilities (translation, disk copying, disk maintenance, virus detection)

3. Peripherals
   -- Top end Purchase is not justified for only 1 or 2 departments
   -- Laser / Color printers
   -- Color Scanner (flatbed, slide)
   -- Zap Shot
   -- Video (cards, videodiscs, vcr)
   -- Audio (sound boards, speakers)
   -- CD's

B. Faculty Usage -- We have found that our faculty are selfmotivated when given resources with which to work
1. Training
   -- Professional Development weekends (for specific audience, department, or division)
   -- Demonstrations (software capabilities, new features, etc.)
   -- Basic Skills (word processing, DOS, Windows, etc.)
   -- Specialized topics (multimedia)
   -- Niche training (designed around specific group)

2. Tips & Techniques The Faculty Innovation Center is also a repository for documentation
   -- Handouts
   -- Manuals
   -- Periodicals
3. Developmental Support Staff

-- Faculty must be supported in their efforts; if they have no one to ask questions, they will become frustrated, and then lose interest
-- Trainer to offer classes on available software
-- Instructional Designer to assist individual faculty projects
-- Faculty Innovation Center Consultant to assist faculty using the equipment in the center

C. Classrooms
1. Learning Centers must be provided
2. Electronic classrooms are available on a collegewide, shared basis (MCC has 4)
3. Rolling Classrooms consisting of computers with capabilities similar to those in the FIC with a projection panel and overhead (these turn any classroom into a computerized demonstration room)

D. Funding -- How the Faculty Innovation Center eliminates problems
1. Equipment is available to all faculty members
2. The FIC can keep current with the latest software
3. The FIC can justify professional development for multiple departments or divisions because of the wide audience

IV. Funding the Faculty Innovation Center

A. Internal
1. Professional Development Grants
2. Good purchasing agent
3. Volume buy (site license)
   -- group with other departments
   -- institutional funding
4. Operating cost savings
5. Trades
6. Recycle
7. Student Staff/tutors
8. Expenses can be justified over the entire college rather than 1 or 2 departments
9. Rolling classrooms can be used in any classroom. by any faculty member
10. Networks

B. External sources
1. Demo Bundles -- many company offer demonstration bundles at highly reduced rates with the agreement that the equipment is available for demonstration (perfect for the FIC)
2. Academic discounts (IBM, Microsoft -- Be Creative!)
3. State contracts products are often offered at a highly reduced price
4. "Begging" -- Use the argument that the FIC is an excellent place to display products. use form letters. and send a lot
5. Grants
   -- Vendors (CLSG)
   -- Government (VATEA)
6. Negotiate with distributors for the best price

Networking Classrooms and Learning Centers Better Facilities With Fewer Resources

Robert Teague
Director, Academic Computing
Terrance Keys
Coordinator, Professional Development
Academic Computing
Brett Thompson
Network Manager
Academic Computing
Monroe Community College
1000 E. Henrietta Rd., Rochester, NY 14623-5780
716-292-2174
rteague@eckert.acadcomp.monroecc.edu

ABSTRACT

Should classrooms and learning centers be networked? Can increased equipment costs be offset by reduced operating expenses? Especially with shrinking budgets? This workshop presents DEC PATHWORKS installations that save on initial costs as well as annual staff and software expenses. Students and faculty both benefit from a consistent computer environment in classrooms and learning centers. The focus of this workshop is the implementation of networked classrooms and learning centers to reduce operational staff and improve resource availability, user interface consistency, preventive maintenance, and recovery turnaround time. Topics will include the integration of PCs, Macintoshes, and VAXes in a distributed computing environment with a consistent user interface; ideal configuration of file services to distribute applications across multiple servers and to aid in saving software licensing costs; and utilizing PATHWORKS features to aid in preventive maintenance (virus protection, server down time). The goals are to give the participants a basis for deciding whether a network would improve their learning centers and classrooms and to illustrate some recommended configurations.

I. Introduction

A. Monroe Community College is located in Brighton, NY, a suburb of Rochester, with city center in downtown Rochester

B. Enrollment is 13,000; 4,500 FTE

C. There are about 400 fulltime and 450 adjunct faculty members

D. Origin of computer centers
  1. Electronic Learning Centers -- 1986
  2. Faculty Innovation Center -- Spring 1991
  3. Electronic Classrooms -- Fall 1991

II. Problems

A. Time
1. Installing software
   -- Upgrading to new versions
   -- Making application modifications
2. Rebuilding hard drives
3. Virus problems
4. Printer problems

B. User environment problems
1. Menu programs inconsistent (Windows vs. Menu Master)
2. Different software or versions available in each classroom/learning center
3. Different printing methods (networked vs. local. laser vs. dot matrix, switch boxes)
4. Application default settings different in each classroom/learning center

C. Funding
1. Purchasing software licenses for every computer
2. Operational expense
3. Large equipment expense

III. Solution -- Network

A. Reduce time
1. Install software onto server
   -- Provides one software installation vs. 100
   -- Upgrades require one upgrade vs. many
   -- Modify application once.
2. Boot PC's from server
   -- Removes PC's reliance on local hard drive
      -- "Losing" its hard drive
      -- Eliminates lost chains
   -- Makes global modification of boot files simple (autoexec.bat, config.sys, startnet.bat)
   -- Provides robust environment
      -- Student safe
      -- Environment safe
3. Set applications to READONLY on server
   -- Prevents changes to application by students.
   -- Keeps application set consistent throughout the semester. (can't be deleted)
   -- Keeps application in existence! -- "Student Safe"
   -- Prevents viruses from infecting files
4. Place printers on network
   -- Provides easy reconfiguration of computer output
   -- Enables a defective printer's output to be easily redirected to a working printer.

B. Consistent user environment
1. Consistent user interface
   -- Application configured the same in each classroom/learning center
   -- Application set available in faculty offices
   -- Application set available off campus

-- Students access same personal data in any classroom/learning center

2. Same software or versions
   -- Application software is available in all classrooms/learning centers not just where installed on HD
3. Consistent printing methods
   -- Printing sharing method uniform over network
4. Application default settings the same in each classroom/learning center
   -- Each application is protected on server (settings can't be modified)

C. Savings

1. Software licensing
   -- Buy for number of maximum simultaneous users
   -- Set server to restrict number of users of an application
2. Operating savings
   -- Need fewer operational staff
   -- Much less time to install, modify, or upgrade classroom/learning center
   -- Use less knowledgeable staff (use student aides or tutors vs. professional)
3. Equipment savings
   -- Printers easily and efficiently shared.
   -- Purchase smaller hard drives since applications reside on servers
   -- Purchase fewer math coprocessors

IV. Funding

A. Internal

1. Good purchasing agent
2. Volume buy
   -- Group with other departments
   -- Institutional funding
   -- Network version
   -- Smaller number of software copies
3. Operating cost savings
4. Trade equipment
5. Recycle equipment
6. Expenses can be justified over the entire college rather than 1 or 2 departments

B. External sources

1. Demo Bundles many company offer demonstration bundles at highly reduced rates
2. Academic discounts (DEC, IBM, Microsoft Be Creative!)
3. State contracts products are often offered at a highly reduced price
4. "Begging" -- Use the argument that the learning centers are an excellent place to display products. Use form letters, and send a lot
5. Grants
   -- Vendors (CLSG)
   -- Government (VATEA)
6. Negotiate with distributors for the best price

Math Visualization With Spreadsheets

David S. Tucker
Division of Mathematical Sciences
Midwestern State University
Wichita Falls, TX
817-689-4414

Abstract

In this paper we investigate the use of spreadsheet programs that present information in the manner of a sophisticated graphing calculator, and show how these are used to aid visualization in mathematical problems. Advantages and disadvantages of the use of spreadsheet graphics, in place of graphing calculators, will be presented. The paper includes examples of spreadsheet macro programs that actually simulate certain convenient features of graphing calculators.

Introduction

Technology has brought to the classroom a new way of looking at mathematical problems. The ability of computer programs such as Derive, Mathematica, or Maple to quickly and easily graph complicated functions has brought a new meaning to "visualization" in mathematics. The introduction of inexpensive graphing calculators has allowed mathematics teachers to expect the student to take a much more visual approach to learning the subject. No longer is it necessary to spend half of the course teaching the student graphing techniques to be able to quickly draw a representative sketch of the function. When numerical approximations to solutions of equations are needed, the student has quick access to methods for finding them, or can simply graph the function on a calculator or computer and inspect it visually to estimate the roots.

One drawback to all of this, however, is the fact that symbolic manipulating computer programs such as those mentioned above, and sophisticated graphing calculators such as the TI-81, the TI-82, or the TI-85, can be very intimidating to certain students, particularly those who are majoring in non-scientific fields such as business or social sciences. There is, however, one piece of software that is commonly used by students in business and which has tremendous graphing capabilities. That is the spreadsheet program. The modern spreadsheet program is not only very good for handling accounting spreadsheet applications, but also has built-in mathematical functions and sophisticated graphing functions. Mathematicians have been using spreadsheets to teach mathematical concepts since the early 1980's when they were first introduced, and Dr. Robert S. Smith organized a session on use of spreadsheets at the Joint Meetings of the American Mathematical Society and the Mathematical Association of America in Baltimore in January 1992 [1]. Several of the papers at that meeting concerned graphing and visualization concepts. At the 1993 Joint AMS/MAA meeting in San Antonio, a session was presented on visualization in mathematics [2], and again, several papers dealt with the use of spreadsheets. The author presented a paper [7] at the first annual Rocky Mountain Small College Computing Conference in Longmont, Colorado demonstrating how spreadsheet graphics might be used in pre-calculus or calculus in place of the graphing calculator. Several textbooks and supplements have appeared recently, which use spreadsheets to augment the learning process in courses from finite mathematics to calculus to engineering mathematics. For example, see [3], [4], [5].

The comfort level of certain students with spreadsheet software is certainly higher than with a typical scientific calulator, but the graphing calculator has added a new dimension. The simple way in which students can enter functions, the simple graph menus with zoom and trace capabilities at first seem to make the calculator the device of choice. Students can carry them around in their pocket or backpack, and can whip them out at will to tackle any problem. On the other hand, non-science majors are tending more and more to rely upon computers as their tool of choice. With word processing and application software of many sorts, these students think of doing homework on the computer. For them, the use of a computer to solve problems appears natural. As computers become cheaper and more portable, this trend can only increase. The size of a computer screen is a plus, compared to the small size and limited display capabilities of current calculators.

What is needed, then is a compromise — a way to do with the familiar spreadsheet the same tasks that seem to be so natural with a graphing calculator. In particular, this includes zoom and trace capabilities. New versions of current spreadsheet programs come with more and more sophisticated graphing capabilities. For instance, Quattro Pro 4.0 comes with zoom and pan capabilities. Quattro Pro and Lotus both have capabilities of creating graphics windows which are active and will show changes in the graph as soon as (almost — depending upon the speed of your computer) the user makes changes in the data on the spreadsheet. In addition, these programs have an underlying macro programming language with user menu capabilites which allow the user to set up menus to access almost any capability of the spreadsheet. It is these features of modern spreadsheet programs that lead the author to investigate what might be involved in trying to create a spreadsheet program which immitates a graphing calculator. This paper shows how this was done and discusses the results.

Spreadsheet graphs - what are they?

In order to understand some of the capabilities and limitations of a spreadsheet, we should first look at how a spreadsheet graph is created. The user sets up a table of values for x in one column, and a table of values for the function in a second (nearby) column. Actually, most spreadsheets allow for up to six functions to be graphed simultaneously, in which case the user creates up to six columns of y values. In a graphing calculator, this is done automatically by first defining the function, and then defining a range over which the function is to be graphed. The calculator then creates its own table of values and plots points. These can be either connected with straight lines, or plotted as separate dots. The same is true for the xy-plots on spreadsheets. Once the tables are set up, the user can use the graph submenu to define a range of values for x and a range of values for y. When the user either selects the View option or presses the F10 key, the program will then plot the points from the table. These can be either plotted as distinct points, or (more commonly) be connected by straight lines. One major difference here is that the number of points plotted is set by the user. The program plots the points in the specified range in the table (unless there are too many for the resolution of the screen, of course).

Up to this point, there is really little difference between the two devices, except in the way the functions are defined. For instance, on a TI-85 calculator, one might type y1 = x^2+2x or y2 = xxx-3xx+5x-2. On the spreadsheet these same two functions would look like

+A10^2+2*A10 in cell B10 and +A10^3-3*A10^2+5*A10-2 in cell C10. These formulas are then copied down their respective columns for, say, one hunmdred rows (to give 100 points to plot). It would take the creation of a parser to be able to imitate the simple method of defining a function on the spreadsheet. Perhaps that is a topic for another paper.

Imitating a Graphing Calculator

In order to make the spreadsheet imitate the graphing calculator, we need to resort to user defined menus. These are discussed in some detail in [6]. For the purposes of this paper, we will use Quattro Pro 3.0 in WYSIWYG mode. The menus are broken down in a way to mimic those in a TI-85. The first menu is a graph menu. This menu includes the basic options as illustrated in the Figure 1, Appendix A. The "y(x)=" option allows the user to select one or more of the four defined functions as the active function (see Figure 2). A spreadsheet can graph up to six functions. This application uses one of the six for the trace function and another for the cursor movement functions (done with the arrow keys on the calculator). This leaves four graphs for actual user-supplied functions. These functions having been defined in the appropriate columns in the manner described above.

The range option (Figure 3) places the cursor in the range display and queries the user for values for xMin, xMax, yMin, yMax. The menu selections may be made using either a mouse or the cursor movement keys. Note that, in an attempt to immitate the calculator, this menu violates the suggested feature of having each menu item start with a different letter so that it can be selected simply by pressing that letter. If one uses a mouse for menu selection, this defect poses no real problems. If one does not use a mouse, then the cursor movement keys can be used to select the menu item.

The zoom option (Figure 4) is designed to immitate the zoom feature of the calculator. There is a zoom factor "zoomfact" which can be changed by the user. In order to accomplish a zoom, the program needs to know the location of the center of the portion of the graph in the display. This can be calculated as the average of the maximum and minimum x values and the average of the maximum and minimum y values. Due to the nature of a spreadsheet, these values can be stored in a cell outside of the displayed range and updated whenever the range values are changed.

In order to zoom in or out, it is necessary to change the minimum and maximum values of x and y in the range. This is how the calculator does it. The algorithm used here is to calculate the new values by the formulas:

xMin = xCenter - (1/zoomfact)(xMax - xMin)/2
xMax = xCenter + (1/zoomfact)(xMax - xMin)/2
yMin = yCenter - (1/zoomfact)(yMax - yMin)/2
yMax = yCenter + (1/zoomfact)(yMax - yMin)/2

To zoom out, we use the same formulas, replacing 1/zoomfact by zoomfact. Due to the problems of recalculation of formulas, it is sometimes not convenient to calculate a value and assign it to a variable as you might in writing a Pascal program. The spreadsheet will sometimes make several assignments before recalculating, so that if one calculation depends upon the result of the previous one, the previous cell might not be updated yet. As a consequence, this spreadsheet calculates values for the next zoom each time the range is changed, and those values are stored in case they are needed. Whenever a recalculation is required, we use either {calc} or {recalc} in the macro routine.

Each of the submenus in this spreadsheet include a "graph" option. This does nothing more than give the command to display the graph (Figure 9) using the full screen. Usually, the user of a spreadsheet wants to set up the graph, view it, and then go on to something else. Even with the graphing calculator, once a graph has been constructed, selecting certain menu items removes the graph from the display to make room for the next menu. The graph option works in this way, in that as soon as a key is pressed, the display goes back to the menu.

There are, however, certain times when you want to actively make changes to the graph – you want to see certain computed values on the screen and see the graph too. In particular, this is true on the graphing calculator with the trace and zoom options. The current location of the cursor is shown on the graph at the bottom of the screen. Quattro Pro and Lotus 1-2-3 have an option which allows for this also. That is the window option. The user is allowed to select a range (in this case, a portion of the visible screen) for a graphics window, and then the current graph will be displayed in the window. The advantage to this is that as you make changes to the data which defines the graph, the graph in the window is updated. The drawback, however, is that in order to put the graph in the window, the program must scale it down to fit the range of the window. Each time the graph is updated, the computer must re-create the image. There are lots of computations involved. The program uses a temporary disk file, and if the computer is slow, or the disk is slow, the delay can be annoying. On the other hand, since the window is smaller than the screen, the screen can be arranged so that the menu options and certain important values are on the screen with the graph, and not on the image of the graph as with the calculator. Figure 7 illustrates what the screen looks like with the graph insert turned off. The graph can be viewed at any time by selecting the Graph option off a convenient menu.

In order to set up the trace function (Figure 5), it is necessary to be able to designate one point on the graph with a special symbol. The way this was done was to use one of the six graphs as the trace graph. Whereas, on the four function graphs the option to connect the points with lines only was selected, on the trace graph the points are plotted using symbols, namely a plus (+) symbol. We want, however, to plot only one point, and that point coincides with a point on the graph being traced. The graphing calculator starts a trace by selecting the point on the curve lying on the vertical line through the center of the graph. The marker is moved left or right along the graph using the cursor movement keys. The location of the marker is displayed at the bottom of the screen.

These same steps can be done on a spreadsheet by creating a couple of cells to hold the x and y coordinates of the trace marker. The location of the trace marker is visible as long as the trace menu is on, but will disappear when we exit from the menu. The exit routine also turns off the trace marker. Notice also that, as on the graphing calculator, the trace marker can be made to trace any one of the four curves (the curve should be displayed on the screen).

The crosshair marker (Figure 6) behaves much like the trace function, except that the marker is not restricted to points on a curve. The intent of this marker is two-fold. First, the user can estimate a value on the graph without actually tracing the graph. Secondly, on the graphing calculator, the crosshair is used to select a point for the center of a zoom operation. The menu here is set up to do that, although the actual routine for this operation is not currently included in this example.

Conclusions

The purpose of this exercise was, in part, to see whether the functions of a graphing calculator could be imitated on a spreadsheet. The answer is clearly yes. There are tradeoffs, however. The spreadsheet graph is larger, and includes a scale as a a standard feature. On the other hand, the spreadsheet places the scale for the y-axis along the side, and does not actually plot a y-axis in the middle of the screen. By plotting a background grid, this disadvantage pretty much disappears.

The graphing calculator has a convenient trace function which moves the cursor without redrawing the graph. The spreadsheet, on the other hand, has to redraw the graph whenever a change is made to it. This is a little disconcerting, but something that one can live with.

Neither device is particularly fast when it comes to redoing the graph. Actually, the computer has, perhaps, a slight advantage when the graph insert is turned off, and the graph is viewed with the Graph option. This is certainly the case when graphing several graphs on the same screen.

No attempt was made to try to include any of the sophisticated math capabilities in this application. For instance, the TI-85 has a function to solve for the x-intercept. This was not included here, although the student can adjust the range until the graph crosses the axis on the screen, and then use the table of values for the graph, or the trace function, to get an approximate value.

Currently, there is no way to conveniently enter the new function. The user must exit from the Graph Menu and go to the column where the y-values are defined, set up the formula for the desired function, copy it down, and then go back to the menu by pressing ALT-m. Although this is a little inconvenient, and will be changed in future revisions, it is not hard to do.

The functionality of this spreadsheet, as far as it goes, is similar to that of the calculator. A student who has a higher comfort level with computers and spreadsheets, than with complicated scientific calculators, can adapt to the use of this fairly well. For the purpose of viewing and tracing graphs, it works quite well.

References

1. Abstracts of Papers Presented to the American Mathematical Society, Issue 79, Volume 13, Number 1, January 1992

2. Abstracts of Papers Presented to the American Mathematical Society, Issue 85, Volume 14, Number 1, January 1993

3. Etter, D M, Lotus 1-2-3: A Software Tool for Engineers, Addison-Wesley Publishing Company, Reading, MA, 1992

4. Spero, Samuel W, The Electronic Spreadsheet and Elementary Calculus: Graphing and Numerical Methods, HarperCollins Publishers, New York, NY, 1991

5. Spero, Samuel W, Topics in Finite Mathematics: An Introduction to the Electronic Spreadsheet, HarperCollins Publishers, 1993

6. Tucker, David S, "Spreadsheet Macros as a Tool for Teachers", Proceedings of the 1991 ASCUE Summer Conference, June 1991

7. Tucker, David S, "Spreadsheets as an Alternative to Graphing Calculator", The Journal of Computing in Small Colleges, Volume 8, Number 4, March 1993

Appendix A Sample Graphs

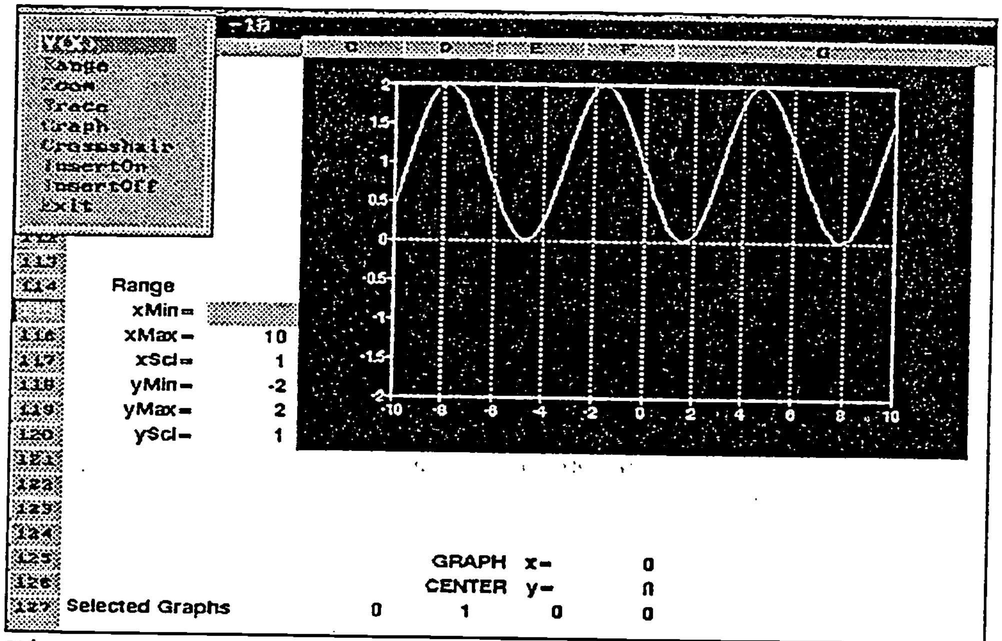


Figure 1 Graph Menu

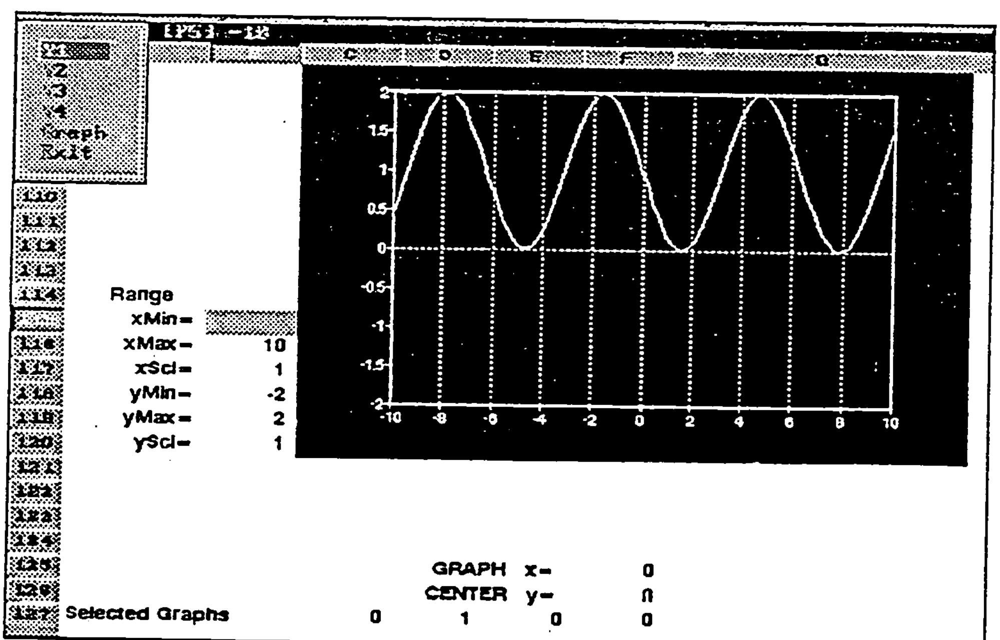


Figure 2 Graph Select Menu

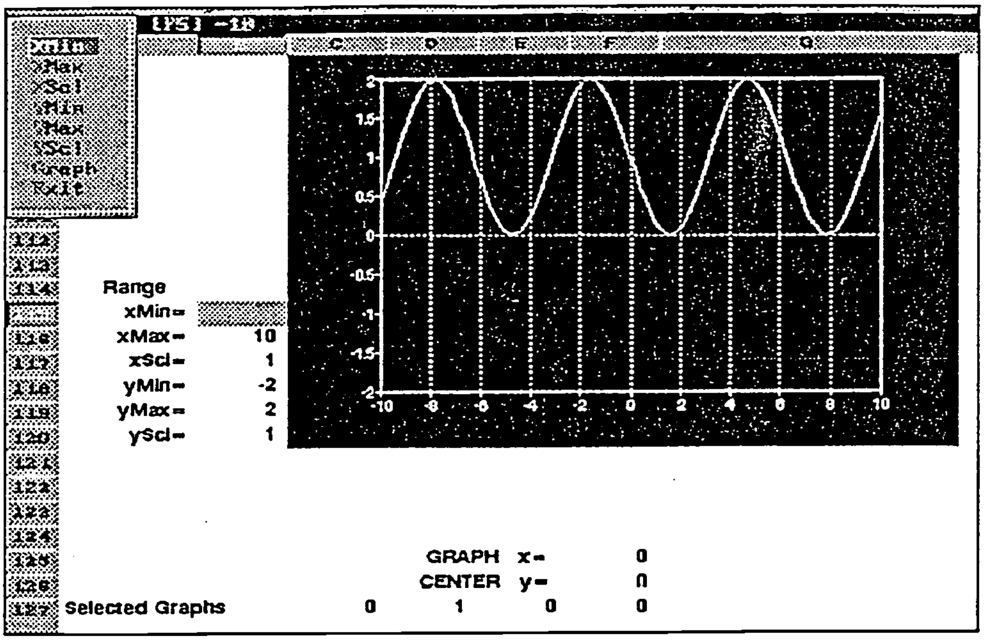


Figure 3 Range Menu

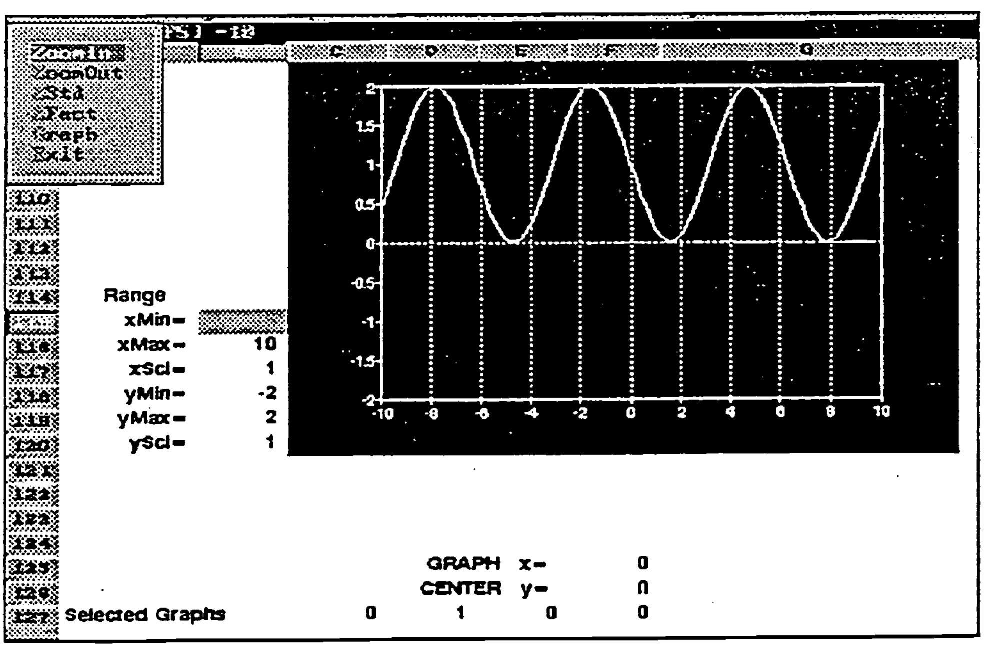


Figure 4 Zoom Menu

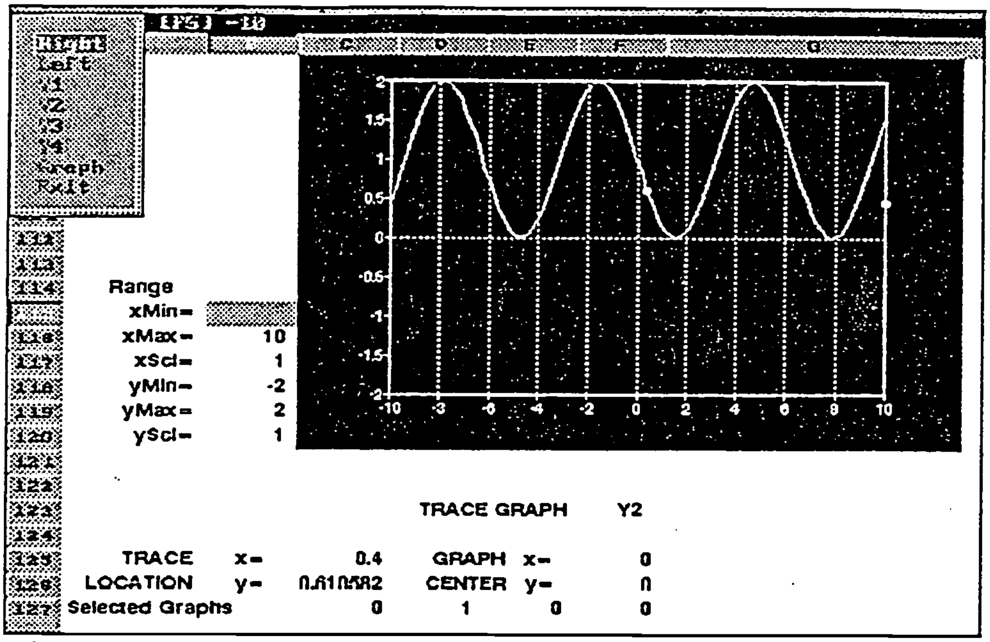


Figure 5 Trace Menu

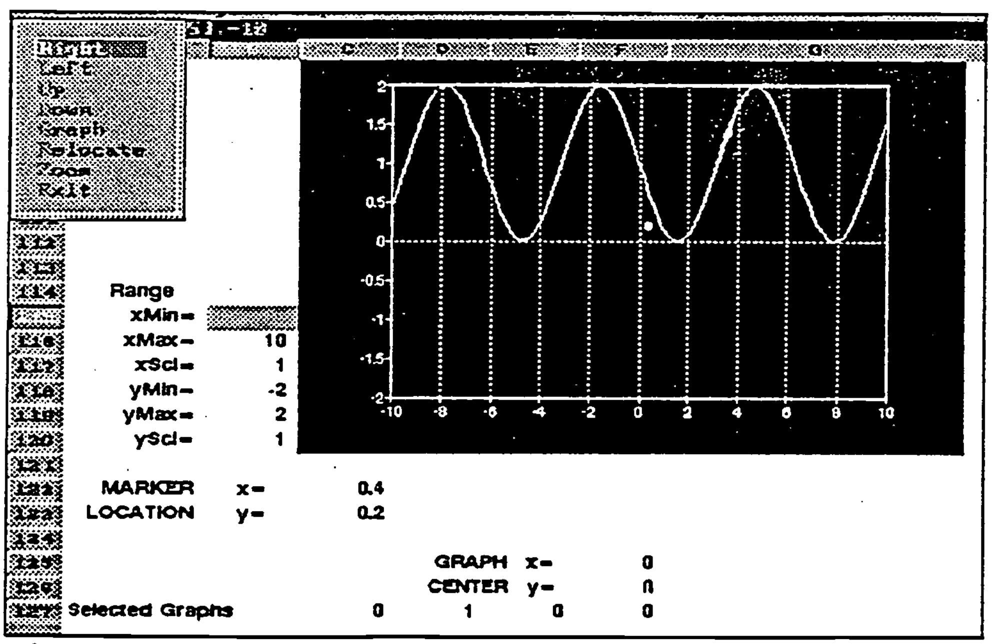


Figure 6 Crosshair Menu

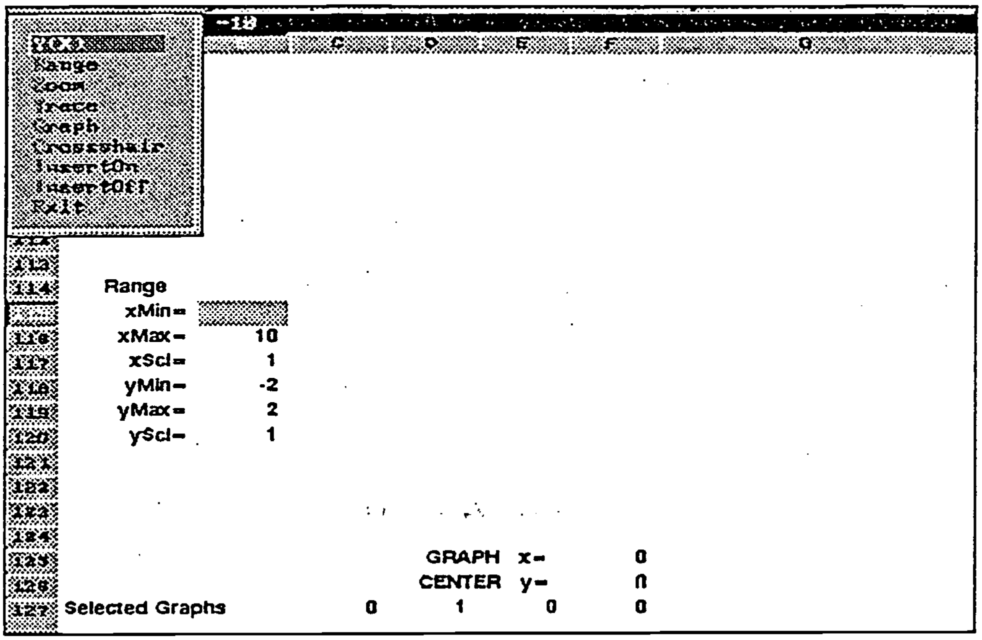


Figure 7 Graph Insert Off

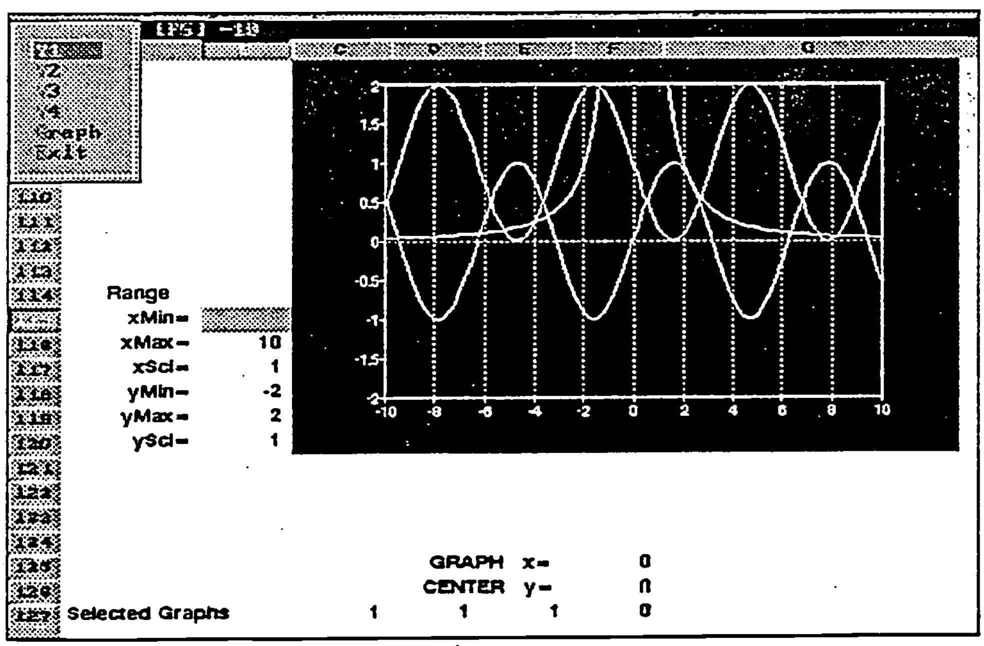


Figure 8 Multiple Graphs

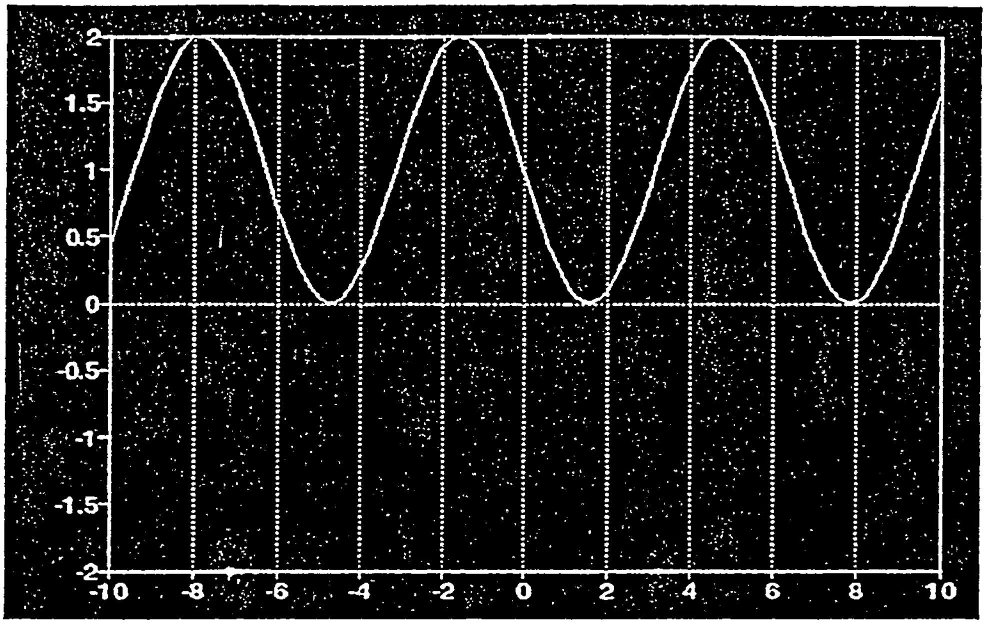


Figure 9 Graph Option

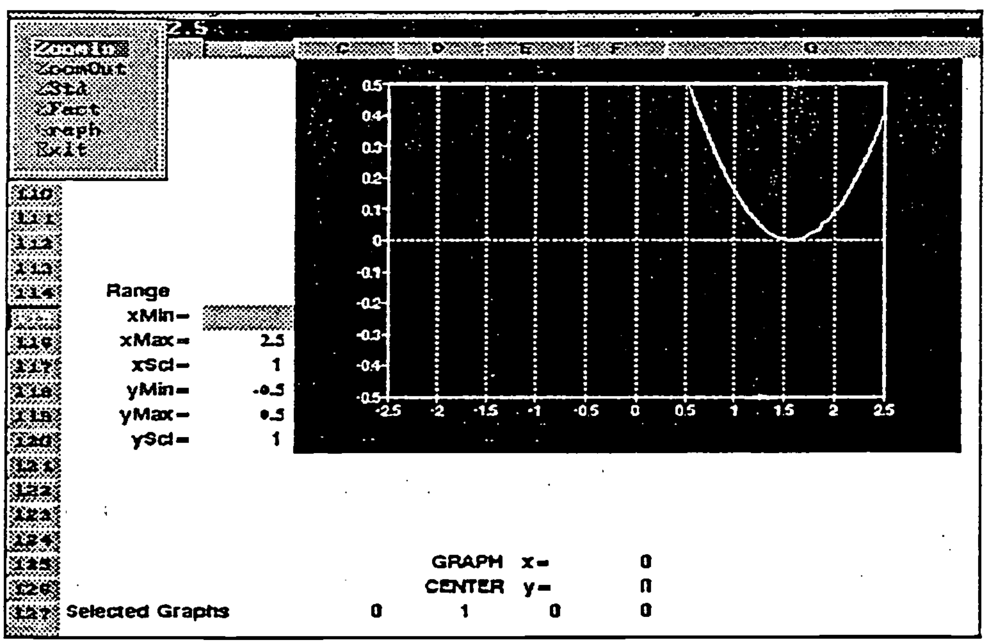


Figure 10 Zooming In

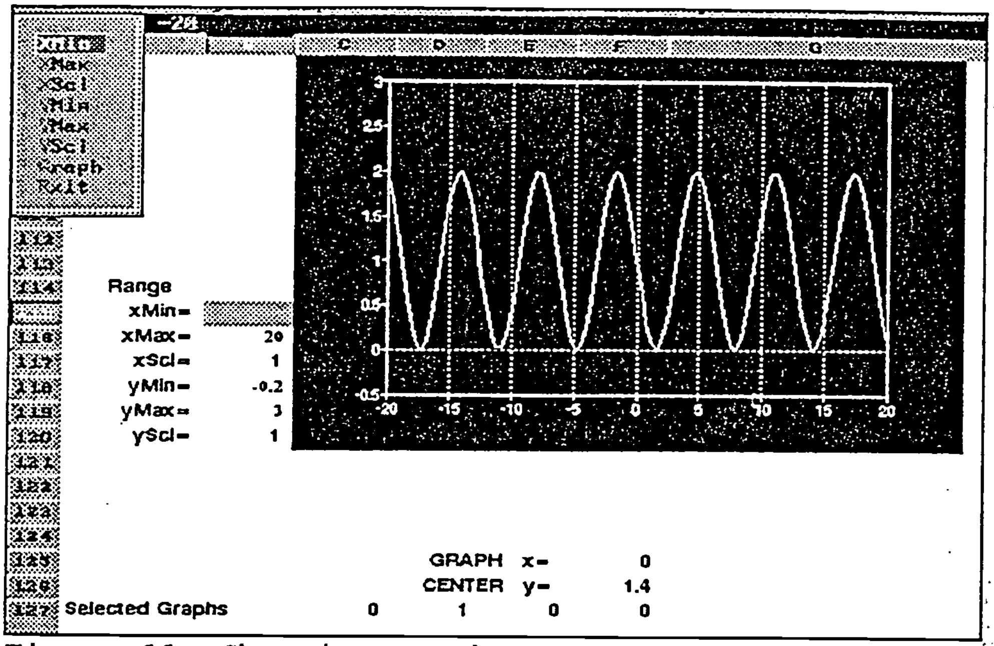


Figure 11 Choosing Another Range

X-Values	Y1	Y2	Y3	Y4	Cross-hair	Trace	Temp Values	Count
-10.00	0.54402	0.455979	0.03992		NA	NA	0.54402	1
-9.80	0.36648	0.633521	0.04156		NA	NA	0.36648	2
-9.60	0.17433	0.825673	0.04331		NA	NA	0.17433	3
-9.40	-0.0248	1.024775	0.04517		NA	NA	-0.0248	4
-9.20	-0.2229	1.22289	0.04715		NA	NA	-0.2229	5
-9.00	-0.4121	1.412118	0.04926		NA	NA	-0.4121	6
-8.80	-0.5849	1.584917	0.05152		NA	NA	-0.5849	7
-8.60	-0.7344	1.734397	0.05394		NA	NA	-0.7344	8
-8.40	-0.8546	1.864599	0.05663		NA	NA	-0.8546	9
-8.20	-0.9407	1.940731	0.05931		NA	NA	-0.9407	10
-8.00	-0.9894	1.989358	0.06231		NA	NA	-0.9894	11
-7.80	-0.9985	1.998543	0.06553		NA	NA	-0.9985	12
-7.60	-0.9679	1.96792	0.06901		NA	NA	-0.9679	13
-7.40	-0.8987	1.898708	0.07278		NA	NA	-0.8987	14
-7.20	-0.7937	1.793668	0.07686		NA	NA	-0.7937	15
-7.00	-0.657	1.656987	0.0813		NA	NA	-0.657	16
-6.80	-0.4941	1.494113	0.08613		NA	NA	-0.4941	17
-6.60	-0.3115	1.311541	0.09141		NA	NA	-0.3115	18
-6.40	-0.1166	1.116549	0.09718		NA	NA	-0.1166	19
-6.20	0.08309	0.916911	0.10362		NA	NA	0.08309	20
-6.00	0.27942	0.720585	0.1105		NA	NA	0.27942	21

Figure 12 Function Definition Area

Appendix B Macro Routines

Startup Menus

```
ALT-R  Range Menu Startup

A1:  '\r
B1:  '{windowsoff}{paneloff}
B2:  '{home}
B3:  '{goto}a105~
B4:  '{right}{down 10}
B5:  '{windowson}{panelon}
B6:  '{menucall rangemenu}
B7:  '{Branch \m}
```

```
ALT-M  Graph Menu Startup

A9:  '\m
B9:  '{windowsoff}{paneloff}
B10: '{home}
B11: '{goto}a105~
B12: '{right}{down 10}
B13: '{windowson}{panelon}
B14: '{menucall Graphmenu}
```

Range Menu

```
E1: 'rangemenu
F1: 'xMin
F2: 'Set xMin
F3: '{Windowsoff}{Paneloff}
F4: '{goto}b115~{windowson}{panelon}
F5: '{getnumber "Enter xMin: ",xMin}{recalc xMin}
F6: '{/ Block;Values}xCenter~tmpxcen~
F7: '{/ Block;Values}xMin~txMin~
F8: '{branch \r}

G1: 'xMax
G2: 'Set xMax
G3: '{Windowsoff}{Paneloff}
G4: '{goto}b116~{windowson}{panelon}
G5: '{getnumber "Enter xMax: ",xMax}{recalc xMax}
G6: '{/ Block;Values}xCenter~tmpxcen~
G7: '{/ Block;Values}xMax~txMax~
G8: '{branch \r}

H1: 'xScl
H2: 'Not Available
H3: '{Windowsoff}{Paneloff}
H4: '{branch \r}

I1: 'yMin
I2: 'Set yMin
I3: '{Windowsoff}{Paneloff}
I4: '{goto}b118~{windowson}{panelon}
I5: '{getnumber "Enter yMin: ",yMin}{recalc yMin}
I6: '{/ Block;Values}yCenter~tmpycen~
I7: '{/ Block;Values}yMin~tyMin~
I8: '/gyl
I9: @STRING(SYMIN,4)
I10: '~qq
I11: '{branch \r}

J1: 'yMax
J2: 'Set yMax
J3: '{Windowsoff}{Paneloff}
J4: '{goto}b119~{windowson}{panelon}
J5: '{getnumber "Enter yMax: ",yMax}{recalc yMax}
J6: '{/ Block;Values}yCenter~tmpycen~
J7: '{/ Block;Values}yMax~tyMax~
J8: '/gyh
J9: @STRING(SYMAX,4)
J10: '~qq
J11: '{branch \r}

K1: 'yScl
K2: 'Not Available
K3: '{Windowsoff}{Paneloff}
K4: '{branch \r}

L1: 'Graph
L2: 'Show Graph
L3: '{Graph}
L4: '{branch \r}

M1: 'Exit
M2: 'Previous Menu
M3: '{Return}

E15: 'Graphmenu
F15: 'Y(X)
F16: 'Specify function to graph
F17: '{Restart}
F18: '{Menubranch OnOff}
F19: '{Branch \m}
```

Graph Menu

```
G15: 'Range
G16: 'Set the range
G17: '{Restart}
G18: '{Menucall Rangemenu}
G19: '{Branch \m}

H15: 'Zoom
H16: 'Zoom in or out
```

```
H17: '{Restart}
H18: '{Menucall Zoommenu}
H19: '{Branch /m}

I15: 'Trace
I16: 'Trace the curve
I17: '{Restart}
I18: '{/ CompGraph:FFormat}S
I19: '/snrtrace~/snrtracegraph~
I20: '{Menubranch Tracemenu}

J15: 'Graph
J16: 'Display the graph
J17: '{Graph}
J18: '{Restart}
J19: '{Branch \m}

K15: 'Crosshair
K16: 'Specify a location on the graph
K17: '{/ CompGraph:EFormat}s
K18: '/snrmarker~
K19: '{menubranch crosshairs}
L15: 'InsertOn
L16: 'Show graph in inset
L17: '/gi~graphinsert~q
L18: '{Branch \m}

M15: 'InsertOff
M16: 'Turn off graph insert
M17: '/gh~q
M18: '{Branch \m}

N15: 'Exit
N16: 'Quit the menu
N17: '{Quit}
```

Zoom Menu

```
E25: 'Zoommenu
F25: 'ZoomIn
F26: 'Zoom in
F27: '/evzixmax~xmax~
F28: '/evzixmin~xmin~
F29: '/evziymax~ymax~
F30: '/evziymin~ymin~
F31: '/evxcenter~tmpxcen~
F32: '/evycenter~tmpycen~
F33: '/evxmin~txmin~
F34: '/evxmax~txmax~
F35: '/evymin~tymin~
F36: '/evymax~tymax~
F37: '{calc}
F38: @PROPER("{/ YAxis:Min}")&@STRING($YMIN,4)&"~"
F39: @PROPER("{/ YAxis:Max}")&@STRING($YMAX,4)&"~"
F40: '{Menubranch zoommenu}

G25: 'ZoomOut
G26: 'Zoom out
G27: '{/ BLOCK:VALUES}zoxmax~xmax~
G28: '{/ BLOCK:VALUES}zoxmin~xmin~
G29: '{/ BLOCK:VALUES}zoymax~ymax~
G30: '{/ BLOCK:VALUES}zoymin~ymin~
G31: '{/ BLOCK:VALUES}xcenter~tmpxcen~
G32: '{/ BLOCK:VALUES}ycenter~tmpycen~
G33: '{/ BLOCK:VALUES}xmin~txmin~
G34: '{/ BLOCK:VALUES}xmax~txmax~
G35: '{/ BLOCK:VALUES}ymin~tymin~
G36: '{/ BLOCK:VALUES}ymax~tymax~
G37: '{calc}
G38: @PROPER("{/ YAxis:Min}")&@STRING($YMIN,4)&"~"
G39: @PROPER("{/ YAxis:Max}")&@STRING($YMAX,4)&"~"
G40: '{Menubranch zoommenu}

H25: 'ZStd
H26: 'Reset to default settings

H27: '{Let xmin, -10}
H28: '{Let xmax, 10}
H29: '{let ymin,-10}
H30: '{Let ymax,10}
H31: '{Let xFact,4}
H32: '{Let yFact,4}
H33: '{Let tmpxcen,0}
H34: '{Let tmpycen,0}
H35: '{Let txmin,-10}
H36: '{Let txmax,10}
H37: '{Let tymin,-10}
H38: '{Let tymax,10}
H39: '{calc}
H40: @PROPER("{/
YAxis:Min}")&@STRING($YMIN,4)&"~"
H41: @PROPER("{/
YAxis:Max}")&@STRING($YMAX,4)&"~"
H42: '{Menubranch zoommenu}
I25: 'ZFact
I27: '{Windowsoff}{paneloff}
I28: '{home}{goto}zoomfactors~
I29: '{r 2}{d}{windowson}{panelon}
I30: '{getnumber "Enter x factor: ",xFact}
I31: '{d}
I32: '{getnumber "Enter y factor: ",yFact}
I33: '{Menubranch zoommenu}

J25: 'Graph
J26: 'Display the graph
J27: '{Graph}
J28: '{Restart}
```

J29: '{Menubranch zoommenu}

K25: 'Exit
K26: 'Return to Graph menu
K27: '{Restart}
K28: '{Menubranch Graphmenu}

Crosshairs Menu

E50: 'Crosshairs
F50: 'Right
F51: 'Move right one unit
F52: '{Restart}
F53: '{If mRIGHT> =50}{Let mRIGHT,49}
F54: '{Let mRIGHT, @mod(mRIGHT+1,51)}{calc}
F55: '{Menubranch Crosshairs}

G50: 'Left
G51: 'Move left one unit
G52: '{Restart}
G53: '{If mRIGHT< =-50}{Let mRIGHT,-49}
G54: '{Let mRIGHT, @mod(mRIGHT-1,51)}{calc}
G55: '{Menubranch Crosshairs}

H50: 'Up
H51: 'Move up one unit
H52: '{Restart}
H53: '{If mUP> =50}{Let mUP,49}
H54: '{Let mUP,mUP+1}{calc}
H55: '{Menubranch Crosshairs}

I50: 'Down
I51: 'Move down one unit
I52: '{Restart}
I53: '{If mUP=-50}{Let mUP< =-49}
I54: '{Let mUP,mUP-1}{calc}
I55: '{Menubranch Crosshairs}

J50: 'Graph
J51: 'Draw graph
J52: '{Restart}
J53: '{Graph}
J54: '{Menubranch Crosshairs}

K50: 'Relocate
K51: 'Make current crosshair the new center
K52: '{Call Relocate}
K53: '{Restart}
K54: '{Menubranch Crosshairs}

L50: 'Zoom
L51: 'Zoom with current crosshair location as new center
L52: '{Call Relocate}
L53: '{Restart}
L54: '{Menubranch Zoommenu}

M50: 'Exit
M51: 'Return to main menu
M52: '{Restart}
M53: '{Let mUP,0}
M54: '{Let mRIGHT,0}
M55: '{/ CompGraph;EFormat}N
M56: '/snhmarker~{calc}
M57: '{Branch \m}

Trace Menu

E60: 'TraceMenu
F60: 'Right
F61: 'Move right one unit
F62: '{Restart}{windowsoff}{paneloff}
F63: '{If mRIGHT> =50}{Let mRIGHT,49}
F64: '{Let mRIGHT, @mod(mRIGHT+1,51)}{calc}
F65: '{windowson}{panelon}
F66: '{Menubranch TraceMenu}

G60: 'Left
G61: 'Move left one unit
G62: '{Restart}
G63: '{If mRIGHT< =-50}{Let mRIGHT,-49}
G64: '{Let mRIGHT, @mod(mRIGHT-1,51)}{calc}
G65: '{Menubranch TraceMenu}

H60: 'Y1
H61: '{Restart}
H62: '{If FUN1=0}{Menubranch Tracemenu}
H63: '{If f1Defined=1}{Let Selection,1}{calc}
H64: '{calc}
H65: '{Menubranch TraceMenu}

I60: 'Y2
I61: '{Restart}
I62: '{If FUN2=0}{Menubranch Tracemenu}
I63: '{If f2Defined=1}{Let Selection,2}{calc}
I64: '{calc}
I65: '{Menubranch TraceMenu}

J60: 'Y3
J61: '{Restart}
J62: '{If FUN3=0}{Menubranch Tracemenu}
J63: '{If f3Defined=1}{Let Selection,3}{calc}
J64: '{calc}
J65: '{Menubranch TraceMenu}

```
K60: 'Y4
K61: '{Restart}
K62: '{If FUN4=0}{Menubranch Tracemenu}
K63: '{If f4Defined=1}{Let Selection,4}{calc}
K64: '{calc}
K65: '{Menubranch TraceMenu}
L60: 'Graph
L61: 'Draw graph
L62: '{Restart}
L63: '{Graph}
L64: '{Menubranch Tracemenu}
M60: 'Exit
M61: 'Return to main menu
M62: '{Restart}
M63: '{Let mUP,0}
M64: '{Let mRIGHT,0}
M65: '{/ CompGraph;FFormat}N
M66: '/snhtrace~/snhtracegraph~{calc}
M67: '{Branch \m}
```

Menu to Select or Deselect Graphs to Display

```
E70: 'OnOff
F70: 'Y1
F71: 'Toggle Y1 on or off
F72: '{Let fun1,@mod(fun1+1,2)}{RecalcFUN1}
F73: '{if +fun1=0}/gcmf1nqqqq
F74: '{if +fun1=1}/gcmf1lqqqq
F75: '{Menubranch OnOff}

G70: 'Y2
G71: 'Toggle Y2 on or off
G72: '{Let fun2,@mod(fun2+1,2)}{RecalcFUN2}
G73: '{if +fun2=0}/gcmf2nqqqq
G74: '{if +fun2=1}/gcmf2lqqqq
G75: '{Menubranch OnOff}
H70: 'Y3
H71: 'Toggle Y3 on or off
H72: '{Let fun3,@mod(fun3+1,2)}{RecalcFUN3}
H73: '{if +fun3=0}/gcmf3nqqqq
H74: '{if +fun3=1}/gcmf3lqqqq
H75: '{Menubranch OnOff}

I70: 'Y4
I71: 'Toggle Y4 on or off
I72: '{Let fun4,@mod(fun4+1,2)}{RecalcFUN4}
I73: '{if +fun4=0}/gcmf4nqqqq
I74: '{if +fun4=1}/gcmf4lqqqq
I75: '{Menubranch OnOff}
J70: 'Graph
J71: 'Draw graph
J72: '{Restart}
J73: '{Graph}
J74: '{Menubranch OnOff}

K70: 'Exit
K71: 'Return to main menu
K72: '{Restart}
K73: '{Branch \m}
```

Appendix C Variables, Temporary Variables, and Formulas

```
xCenter        (XMAX+XMIN)/2        txmin     -10
tmpxcen              0              txmax      10
yCenter        (YMIN+YMAX)/2        tymin      -2
tmpycen              0              tymax       2

    zixMax   +TMPXCEN+(TXMAX-TXMIN)/(2*XFACT)
    zoxMax   +TMPXCEN+(TXMAX-TXMIN)*XFACT/2
    ziyMax   +TMPYCEN+(TYMAX-TYMIN)/(2*YFACT)
    zoyMax   +TMPYCEN+(TYMAX-TYMIN)*YFACT/2

    zixMin   +TMPXCEN-(TXMAX-TXMIN)/(2*XFACT)
    zoxMin   +TMPXCEN-(TXMAX-TXMIN)*XFACT/2
    ziyMin   +TMPYCEN-(TYMAX-TYMIN)/(2*YFACT)
    zoyMin   +TMPYCEN-(TYMAX-TYMIN)*YFACT/2

ZOOM FACTORS
  xFact=                        4
  yFact=                        4
```

```
  TRACE   x=  @ROUND(@CELLINDEX("contents".DATABLOCK.0.51+MRIGHT).7)
LOCATION  y=  @ROUND(@CELLINDEX("Contents".DATABLOCK.SELECTION.51+MRIGHT).7)
  GRAPH   x=  @ROUND(+A181.4)
 CENTER   y=  @ROUND(+YCENTER.4)
```

Definitions for the Crosshair and Trace Graphs plus two other columns used to store values and the index required for lookup

```
F128: Crosshair                        G128: Trace
F129: ------                           G129: -----
F130: @IF($MRIGHT+51=I131.$C$123.@NA)            G130: @IF($MRIGHT+51=I131,H131,@NA)

H128: ' Temp                           I129: "Count
H129: ' Values                         I130: \-
H130: ' -----                          I131: 1
H131: @CELLINDEX("Contents".$DATABLOCK.$SELECTION.I131)   I132: +I131+1
```

Appendix D   Named Blocks Referenced in Macro Routines

```
Q1: [W20] 'CENTER
R1: [W15] 'D125..F126
Q2: [W20] 'CROSSHAIRS
R2: [W15] 'F50
Q3: [W20] 'DATABLOCK
R3: [W15] 'A130..E231
Q4: [W20] 'F1DEFINED
R4: [W15] 'I122
Q5: [W20] 'F2DEFINED
R5: [W15] 'I123
Q6: [W20] 'F3DEFINED
R6: [W15] 'I124
Q7: [W20] 'F4DEFINED
R7: [W15] 'I125
Q8: [W20] 'FUN1
R8: [W15] 'C127
Q9: [W20] 'FUN2
R9: [W15] 'D127
Q10: [W20] 'FUN3
R10: [W15] 'E127
Q11: [W20] 'FUN4
R11: [W15] 'F127
Q12: [W20] 'GRAPHINSERT
R12: [W15] 'C105..G120
Q13: [W20] 'GRAPHMENU
R13: [W15] 'F15
Q14: [W20] 'MARKER
R14: [W15] 'A122..C123
Q15: [W20] 'MARKERON
R15: [W15] 'E99
Q16: [W20] 'MRIGHT
R16: [W15] 'I121
Q17: [W20] 'MUP
R17: [W15] 'I120
Q18: [W20] 'ONOFF
R18: [W15] 'F70
Q19: [W20] 'PRNTRNGE
R19: [W15] 'A105..H127
Q20: [W20] 'RANGE
R20: [W15] 'B115..B120
Q21: [W20] 'RANGEMENU
R21: [W15] 'F1
Q22: [W20] 'SELECTION
R22: [W15] 'F122
Q23: [W20] 'TMPXCEN
R23: [W15] 'I103
Q24: [W20] 'TMPYCEN
R24: [W15] 'I105
Q25: [W20] 'TRACE
R25: [W15] 'A125..C126
Q26: [W20] 'TRACEGRAPH
R26: [W15] 'D123..F123
Q27: [W20] 'TRACEMENU
R27: [W15] 'F60
Q28: [W20] 'TRACEON
R28: [W15] 'E98
Q29: [W20] 'TXMAX
R29: [W15] 'K103
Q30: [W20] 'TXMIN
R30: [W15] 'K102
Q31: [W20] 'TYMAX
R31: [W15] 'K105
Q32: [W20] 'TYMIN
R32: [W15] 'K104
Q33: [W20] 'XCENTER
R33: [W15] 'I102
Q34: [W20] 'XCENTER:
R34: [W15] 'I102
Q35: [W20] 'XFACT
R35: [W15] 'I108
Q36: [W20] 'XMAX
R36: [W15] 'B116
Q37: [W20] 'XMIN
R37: [W15] 'B115
Q38: [W20] 'YCENTER
R38: [W15] 'I104
Q39: [W20] 'YCENTER:
R39: [W15] 'I104
Q40: [W20] 'YFACT
R40: [W15] 'I109
Q41: [W20] 'YMAX
R41: [W15] 'B119
Q42: [W20] 'YMIN
R42: [W15] 'B118
Q43: [W20] 'ZIXMAX
R43: [W15] 'M102
Q44: [W20] 'ZIXMIN
R44: [W15] 'O102
Q45: [W20] 'ZIYMAX
R45: [W15] 'M104
Q46: [W20] 'ZIYMIN
R46: [W15] 'O104
Q47: [W20] 'ZOOMFACTORS
R47: [W15] 'G107
Q48: [W20] 'ZOOMMENU
R48: [W15] 'F25
Q49: [W20] 'ZOXMAX
R49: [W15] 'M103
Q50: [W20] 'ZOXMIN
R50: [W15] 'O103
Q51: [W20] 'ZOYMAX
R51: [W15] 'M105
Q52: [W20] 'ZOYMIN
R52: [W15] 'O105
Q53: [W20] '\M
R53: [W15] 'B9
Q54: [W20] '\P
R54: [W15] 'B97
Q55: [W20] '\R
R55: [W15] 'B1
```

Does Technology Improve Instruction? Case Studies in Networking and Multimedia

Mary Wall
Atlantic Community College
Mays Landing NJ
609-646-1079
Barry Centini
SRI, International
Arlington VA

ABSTRACT

In the current economic climate, colleges face diminished resources and increasing demands. While technology is often promoted as a means to better allocate existing resources, faculty are increasingly skeptical of the huge expenditures required. They ask:"Does the technology improve instruction?"

This paper presents two case studies, one in networking technology and one in multimedia, and gives a framework for program evaluation. Although the networking technology case study involved the entire college community and the multimedia case study involves only one class, the evaluation schema remains the same. Evaluation must address four questions:

1. What is the basis for the expected outcomes?
2. What resources are necessary to implement this technology?
3. What happened?
4. Did the expected outcomes occur?

BACKGROUND

Atlantic Community College is located in Mays Landing, New Jersey, 12 miles from Atlantic City. The college's enrollment has grown from 4000 students in 1990 to over 6000 students in 1993. In the same period the State and county governments cut funding. Despite the scarcity of economic resources, the college used monies from the general funds to increase technology on campus, believing that the introduction of networking and multimedia would improve instruction. While many faculty welcomed the new technology, others felt that the scarce resources could better be spent on other priorities such as increasing full time faculty. It was important, therefore, to evaluate the new technologies to see if their introduction did improve instruction.

CASE STUDY I: NETWORKING TECHNOLOGY

The Faculty Support System (FSS) is a commercial software product developed by The Robinson Group (TRG) of Tempe, Arizona. It is a network-based electronic gradebook that provides for teacher/student messages. A course builder allows the teacher to enter an activity set, while a grade book allows the teacher to enter grades; the software automatically keeps a total of points earned and calculates the final average. Thus the teacher is freed from clerical duties. In a study of electronic gradebooks (Zilbert, 1988), the automation of this task was found to save time for the teacher.

The grades are available for student viewing at three designated terminals throughout the college. After students view their grades they may send a message to the teacher. The teacher may send a message to the students at any time.

WHAT WAS THE BASIS FOR THE EXPECTED OUTCOMES?

The availability of grades for students and the ability to communicate electronically was seen by the college as a means of improving student success measured by grades and retention measured by the number of students who complete the course.

The work of Tinto (1987) and Astin (1987) point to the importance of integration of students into the college community as a factor in retention, with special attention to the out-to-class contacts of students with faculty. Students in other colleges who communicated electronically with faculty reported a reduced sense of isolation, a perceived increase in the effectiveness of communication and a pride in the practical knowledge of using electronic mail (Frost and Roberts, 1990.) Students' learning styles may determine a preference for electronic communication (Elsworth, 1991; Stein, 1991).

Since FSS permits students and faculty to communicate without being bound by time or place it was felt that the technology would aid in student integration and promote school success, measured by higher grades.

WHAT RESOURCES WERE NECESSARY TO IMPLEMENT?

To implement the technology the entire campus had to be networked. All teachers who participated in the project had a computer on their desk that was linked to the mainframe Student Information Systems database. Three terminals were dedicated to student FSS use. Teachers were given a small stipend ($200) for their participation.

WHAT HAPPENED?

A pilot group of 12 classes was randomly selected in Fall 1992. Faculty were trained and agreed to use the technology with one class. Teachers and students filled out questionnaires at the beginning and the end of the semester to assess the usefulness of the software. The teachers rated the software higher at the end of the semester than at the beginning, which indicates there may have been initial skepticism about the software. The advantages listed by the faculty were: receiving messages from students, calculating of final grades, and saving time.

Students did use the software: 66% looked at their grades and 60.2% sent the teacher a message. Initially teachers had voiced concern that students would use electronic communication exclusively, but this did not occur: 77% of the students noted that they "spoke with teacher outside of class."

During Spring 1993, 25 teachers are using the system. Teachers new to the project are using the software with one class, but the returning teachers have elected to put most of their classes on the system.

DID THE DESIRED OUTCOMES OCCUR?

There was no noticeable impact on retention in any of the twelve Fall 1992 pilot classes. However, in five of the six of the classes where the teacher taught one class using FSS and one class not using FSS, the grades were better in the FSS class.

Although no more students completed the FSS classes than did other classes taught by the same teacher, student and faculty response to FSS was positive. Student comments were

"I wish all my classes used FSS."

"I like FSS because I don't have to bug my teacher to check my grades."

Faculty liked receiving students' messages electronically and the automatic calculation of the final grades, although they did suggest enhancements to the system. Faculty want the ability to send one message to the entire class and to automatically transfer final grades from FSS to the mainframe database. TRG, the software vendor, will provide this facility in an August 1993 release.

CASE STUDY II: MULTIMEDIA

Although the FSS project required college-wide cooperation, the college funded several teachers to introduce technology into their own classroom. In the Principles of Marketing Class a hypercard stack was developed that included three college-produced Quicktime (QT) movies. This was used by students in Spring 1993.

One of the Principles of Marketing sections is delivered mainly by televised classes. The students meet in person only for final exams. Student evaluations from previous semesters suggested that although they liked the convenience of the format they did miss the classroom interaction.

This hypercard stack was developed to simulate classroom interaction. Students are given data on five cereals and asked to select one to market and give the reasons for their decision. They are then given feedback by three fictional members of their organization. After they receive the feedback, they are asked to re-evaluate their decision and to write a final report stating whether they changed their selection based on the feedback, and to defend their decision. In one version of the case study, the feedback is given on three hypercard cards. In the second version, the feedback is given in three QuickTime full motion videos.

WHAT WAS THE BASIS FOR THE EXPECTED OUTCOMES?

Is a picture worth more than words when it is a moving picture in color with sound? Images are how we first learn about the world and students do tend to process information in this way. Images "remain the most powerful means of communicating information" (Jones, 1993). This simulation was set up in the two separate modes to test whether faculty-produced Quicktime movies in hypercard had a greater effect on the student responses than text. If more students changed their initial cereal selection based on the movies, then the full motion video would be determined to have had an effect.

WHAT RESOURCES WERE REQUIRED?

The hypercard stack was written by the teacher of the course, who also wrote the script for the Quicktime movies. The movies were shot using a video recorder. Then a team of three – the teacher, the Director of Information Technology, and a media specialist – captured the video clips using a SuperMac VideoSpigot card, and the Screenplay application that comes with it. Adobe Premiere version 1 was used to trim the clips and save them as Quicktime movies. A Quicktime template developed at Cornell University was used to link the movies into Hypercard. It required three hours to covert the three 20-second segments into movies and bringing them into hypercard.

WHAT HAPPENED?

Students in the TV section used the program on their own in the college's computer lab with the assistance of a lab aide. Students in the traditional class also used the software; these students used this program as a class exercise with the teacher and the writer of the stack in the computer lab with them.

DID THE DESIRED RESULT OCCUR?

Students in the regular class gave high ratings to the program regardless of the version used. It was seen as a challenge and change from in-class instruction. The TV section will finish their work by the end of the semester when the evaluations will be collected.

SUMMARY

Technology must compete with other priorities for resources. Atlantic Community College has made an effort to evaluate the effects of technology on instruction. While a large project such as FSS requires administrative cooperation, a small project such as the Cereal Case Study requires only that the teacher be willing to set up the control conditions. Data from the evaluations can be used to improve instruction and guide administration in the wise allocation of resources.

REFERENCES

Astin, A., Korn, W. & Green, K. (1987). Retaining and satisfying students. Educational Record, 58(1), 36-42.

Ellsworth, J. H. (1991). Electronically mediated learning among adults. Paper presented at the National Conference on the Adult Learner, Columbia, SC, May 1991. (ERIC Document Reproduction Service No. ED 337 704).

Frost, T. and Roberts, D. (1990). Electronic mail evaluation — a case study. Paper presented at the meeting of ASPERA, Australia and South Pacific External Studies Association, November, 1990, John Hardie, ed. (ERIC Document Reproduction Service No. ED 338 226).

Jones, L. L. (1993). Campus forum. Campus Tech, Spring 1993, 27.

Sein, M. K. (1991). Learning style and the efficiency of computer training methods. Perceptual and Motor Skills, 72(10), 243-248.

Tinto, V. (1987). Leaving College: Rethinking the Causes and Cures of Attrition. Chicago: The University of Chicago Press.

Zilbert, A.B. (1988). Productivity tools for higher education, one approach: Grading Information System for university professors. Ann Arbor, MI: University Microfilms International Dissertation Information Service.

Working Smarter: TQM Principles Applied to an Overworked Computing Services Organization

William P. Wilson
Computing Services
Gettysburg College
Gettysburg PA 17325
wilson@gettysburg.edu
717-337-6933

With the establishment of the position of Associate Provost for Information Technology in the fall of 1990, Gettysburg College altered its computing structure from independent academic and administrative shops to a unified computing organization. At the same time, new organizations also came under Information Technology including Telecommunications, the printing office, and post office. Several new positions were created (trainer, publications, one additional administrative computing support person). A new campus network was planned and implemented over the next three years. Electronic mail was introduced and now flourishes; Internet was introduced and is catching on rapidly. Networking of the residence halls is to be completed this summer. Since the fall of 1990, then, our staff has not grown but our customer base, applications we need to support, and complexity of the operation have increased many times. Our staff rarely felt that a days work had come in and a days work had gone out; our plates almost always were fuller than when we started the day.

Realizing that staff expansion was not very likely in the current climate, our Associate Provost in the Fall of 1992 started his organization down the road to embrace Total Quality Management (TQM) principles. Our Associate Provost decided we needed to learn how to work smarter in order to increase the overall productivity of the organization as we struggled to keep up with our obligations. The general impression was that user satisfaction with our services was declining as it took longer for us to respond. The frequency of return visits was increasing as we spent less time to be sure our changes worked correctly. A classic vicious circle was developing; fire-fighting time was increasing dramatically as time to work on core projects fell. So, job satisfaction began to fall at the same time customers were complaining more about our service. We needed to try something!

After an introductory meeting where the concept was introduced, we started with some exercises to see how we spent our time as a group, the tasks we were performing, our level of satisfaction, and so on. We learned the following things:

(1) as a whole, we were spending as much as 40% of our time in firefighting mode;

(2) we didn't have data to back up any of our hunches

(3) communications within the organization was a big problem

(4) we had no shared vision of what the group should be striving toward as a computing services organization

(5) we didn't have time to learn/keep current as a group

(6) as a whole, we were under too much stress

TQM would appear to be a good match for our organization. It requires an organization to develop and maintain a clearly stated common vision among the team members. It asks the organization to collect data and work with that data rather than with hunches to determine where biggest successes can be achieved. Its emphasis, too, is to think in terms of process and to understand the processes of the organization. The approach, if followed, also addresses communications problems that were plaguing us. Finally, there was an emphasis on developing customer satisfaction by building or delivering excellent product, an area the group all agreed was a growing problem for us.

IMPLEMENTATION AND TRAINING

Our team underwent several stages of training in the TQM approach. First, our associate provost modeled some of the TQM tools in a series of team meetings. A 'team' was charged with developing our mission statement. This first month or two was used as familiarization of the concepts and to get consensus on the group using this approach to improve our productivity.

Next, the whole staff took a one-day retreat to participate in a TQM workshop put on by IBM. We basically went through, in an organized way, many of the exercises we had done in our group meetings back on campus. Brainstorming activities and collapsing of topics proved to be very instructional. These were done interactively on computers which made some elements of group dynamics more interesting and instructive.

Next, a consultant visited campus to teach all of us the tools TQM uses to work through a problem and develop a solution. These included techniques we had been exposed to such as brainstorming, to some new ones, such as Cause/Effect Charts, Imagineering, and a number of others. We were broken into groups, given a business scenario to work with, including data, and used the tools we were learning about to analyze the situation and develop solutions.

Our next stage was to break our group into three teams to work on small, self-contained problems that would allow the teams to use all the tools we had just learned about. Team leaders were selected, each team had a facilitator from the management team to help keep the team on task. Here, we learned to put the concepts into action, and did a bit of data collection, analysis, process development, implemented a change or two and finally went through the iterative process required to arrive at a good solution.

Currently, we have one full-blown team working on establishing a helpdesk system for our organization. This task was selected since it showed the most promise of giving us the most return for our investment in terms of release time for staff members.

RESULTS AND IMPLICATIONS

TQM has not solved all our problems, nor is it likely to in the immediate future. I think most of us feel there is some long-term benefit the approach will provide, but there are many intermediate frustrations.

(1) Communications are still bad

(2) Some promised input from staff on decisions does not seem to be happening

(3) Stopping for data collection slows down the response of staff to user requests

(4) Meetings have doubled (at least) and cut into productive time

(5) Demands by the management team for staff taking a course or in some other way to learn new material has resulted in additional pressures for some staff and have been misinterpreted by some

(6) Protecting turf has cropped up in response to the approach and needs to be dealt with

TQM has made us start to look at what we do and how we spend our time, and will begin to impact the way we establish priorities for new activities we take on. It has started to put a little order in what was chaos. It has slowed the pace a little. From the perspective of the staff, management still must be willing to deliver on some elements of the process, such as sharing more of the decision-making and including staff input before the decisions are made. Our customers are probably still not as aware as they should be about our new approach and that it is intentional that we are responding more deliberately than we have been in the past. Also, there seems to be a hope that by dealing with things in this way that some deeper personnel issues will not have to be dealt with. My guess is that TQM can not be used to escape from dealing with these painful discussions.

NOTICE

REPRODUCTION BASIS

- [x] 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").